Leandro Luciani Tavares

Utilização de mecanismos de roteamento para seleção de sistemas de *Question Answering*

Sorocaba, SP

30 de Maio de 2018

Leandro Luciani Tavares

Utilização de mecanismos de roteamento para seleção de sistemas de *Question Answering*

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação (PPGCC-So) da Universidade Federal de São Carlos como parte dos requisitos exigidos para a obtenção do título de Mestre em Ciência da Computação. Linha de pesquisa: Computação Científica e Inteligência Computacional.

Universidade Federal de São Carlos – UFSCar

Centro de Ciências em Gestão e Tecnologia – CCGT

Programa de Pós-Graduação em Ciência da Computação – PPGCC-So

Orientador: Prof. Dr. Tiago Agostinho de Almeida

Sorocaba, SP

30 de Maio de 2018

Luciani Tavares, Leandro

Utilização de mecanismos de roteamento para seleção de sistemas de Question Answering / Leandro Luciani Tavares. -- 2018.
99 f. : 30 cm.

Dissertação (mestrado)-Universidade Federal de São Carlos, campus Sorocaba, Sorocaba
Orientador: Prof. Dr. Tiago Agostinho de Almeida
Banca examinadora: Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho, Profa. Dra. Sahudy Montenegro González
Bibliografia

1. Classificação hierárquica de domínio. 2. Classificação de perguntas. 3. Gera- ção de perguntas.. I. Orientador. II. Universidade Federal de São Carlos. III. Título.

Ficha catalográfica elaborada pelo Programa de Geração Automática da Secretaria Geral de Informática (SIn).
DADOS FORNECIDOS PELO(A) AUTOR(A)
Bibliotecário(a) Responsável: Maria Aparecida de Lourdes Mariano – CRB/8 6979

UNIVERSIDADE FEDERAL DE SÃO CARLOS

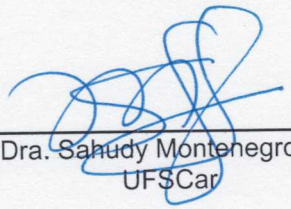
Centro de Ciências em Gestão e Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato Leandro Luciani Tavares, realizada em 30/05/2018:

Prof. Dr. Tiago Agostinho de Almeida
UFSCar

Prof. Dr. Andre Carlos Ponce de Leon Ferreira de Carvalho
USP



Profa. Dra. Sahudy Montenegro González
UFSCar

Certifico que a defesa realizou-se com a participação à distância do(s) membro(s) Andre Carlos Ponce de Leon Ferreira de Carvalho e, depois das arguições e deliberações realizadas, o(s) participante(s) à distância está(ao) de acordo com o conteúdo do parecer da banca examinadora redigido neste relatório de defesa.

Prof. Dr. Tiago Agostinho de Almeida

Dedico este trabalho aos meus amados avós, Maria do Carmo Benedetti Luciani (Vó Carmem) *e José Maria Luciani* (Vô Zé)*, que sempre torceram pelo meu sucesso e com toda sua simplicidade transmitiram a importância de me tornar uma pessoa de valor e caráter.*

Agradecimentos

Agradeço,

aos meus queridos pais, Laurene Lázara Luciani Tavares e Antonio Carlos Tavares, por me permitirem chegar até aqui e por todas as conversas e os sábios conselhos durante essa minha breve jornada;

ao meu talentoso orientador, Prof. Dr. Tiago Agostinho Almeida, pelos ensinamentos e pela paciência durante a execução deste trabalho;

ao prestativo Dr. Renato Moraes Silva, pela extensa colaboração para o sucesso desse trabalho;

ao polivalente amigo, Álvaro Nascimento Vieira, pela revisão deste e outros trabalhos;

aos mentores Enrique Pimentel Leite de Oliveira, Fábio Lopes Caversan e Rodrigo Cristiano Silva, pelos incentivos de sempre buscar conhecimento e formação e também, por permitirem me ausentar por diversas vezes das minhas atividades profissionais para o sucesso desse trabalho;

e a todos os professores que em algum momento participaram da minha formação, permitindo que eu chegasse até aqui.

"For millions of years, mankind lived just like the animals. Then something happened which unleashed the power of our imagination. We learned to talk and we learned to listen. Speech has allowed the communication of ideas, enabling human beings to work together to build the impossible. Mankind's greatest achievements have come about by talking, and its greatest failures by not talking. It doesn't have to be like this. Our greatest hopes could become reality in the future. With the technology at our disposal, the possibilities are unbounded. All we need to do is make sure we keep talking." (Stephen William Hawking)

Resumo

A evolução do modo de interação entre os humanos e os computadores vem acompanhando a evolução tecnológica dos próprios computadores. Esse processo culminou no surgimento de uma subárea da computação denominada *Question Answering (QA)*, que proporciona uma forma de interação natural entre a máquina e o homem — o modelo de interação pergunta-resposta. Esse modelo manifesta-se em ao menos duas formas de sistemas: sistemas de domínio restrito, os quais se resumem a temas específicos, mais complexos e limitados, e sistemas de domínio aberto, os quais abordam assuntos gerais, não se restringindo a um tópico particular, exibindo uma diversidade que impede a apresentação em maior detalhamento sobre qualquer tópico em especial. Idealmente, um sistema de QA deveria combinar de modo prático as características mais evidentes dos modelos existentes, a fim de unir a variedade de tópicos abordados pelos sistemas de domínio aberto e a minuciosidade dos sistemas de domínio restrito. Uma possível solução seria integrar diversas instâncias de sistemas de domínio restrito como um sistema de domínio aberto. Para isso, com base na pergunta do usuário, é necessário um mecanismo capaz de selecionar uma entre as instâncias disponíveis, a fim de que a instância selecionada produza a resposta sobre o domínio representado. Nesse contexto, este trabalho apresenta um mecanismo de seleção de instâncias de sistemas de QA, representado por um classificador de domínio hierárquico de perguntas. Os domínios são naturalmente organizados em uma taxonomia hierárquica. Ao classificar uma pergunta, esse classificador procura selecionar o sistema de QA mais adequado para produzir a resposta. Contudo, para treinar esse classificador, é mandatório obter dados rotulados de qualidade. Para contornar essa dependência e viabilizar a aplicação prática da proposta, neste trabalho foi aplicada uma estratégia automática de geração de perguntas baseadas em documentos, o que resultou em uma grande base de perguntas sintéticas. Ao avaliar o desempenho do classificador em uma base de perguntas reais, os resultados se mostraram bastante promissores, indicando que a estratégia de geração de perguntas automaticamente é viável para o treinamento do classificador. Assim sendo, o mecanismo de roteirização desenvolvido pode ser usado na composição de um sistema de QA híbrido mais robusto e ao mesmo tempo universal, de tal forma que agregue as principais qualidades de cada um dos tipos de sistemas de QAs.

Palavras-chaves: Classificação hierárquica de domínio. Classificação de perguntas. Geração de perguntas. Sistemas de QA.

Abstract

The evolution of the interaction between humans and computers has accompanied the technological evolution of computers themselves. This process culminated in the rise of a subfield of computing called Questioning Answering (QA), which provides a form of natural interaction between machines and humans — the Question-Answer interaction model. This model manifests itself in at least two forms of systems: restricted domain systems, which are specific, limited, more complex, and open domain systems, which address general subjects, not constrained to a particular topic, exhibiting a diversity that prevents the presentation in greater detail on any particular topic. Ideally, a QA system should combine in a practical way the main characteristics of the existing models in order to unite the variety of topics covered by the open domain systems and the thoroughness of the restricted domain systems. One possible solution is to combine several instances of restricted domain systems into a single open domain system. For this, a routing mechanism able to select one of the available instances must exist. Then, the selected instance should answer the question about the represented domain. In this work, a mechanism for selecting instances of QA systems is presented in shape of a hierarchical question domain classifier. Domains are naturally organized in a hierarchical taxonomy. When classifying the proposed questions into one of them, the classifier tries to select the most suitable QA system to answer the question. Although, for the purpose of training the classifier, quality data is mandatory. To tackle this dependency, an automatic question generation strategy based on documents was applied, resulting in a large synthetic question dataset. Results were promising when the classifier was evaluated against a real question dataset, suggesting that automatic question generation is feasible to train the classifier. In conclusion, the developed routing mechanism can be used to build a solid and universal hybrid QA system, ensembling the best qualities of each kind of system stand-alone.

Key-words: Hierarchical domain classification. Question classification. Question generation. QA systems.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AM	Aprendizado de Máquina
API	*Application Programming Interface (Interface de Programação de Aplicativos)*
BD	Banco de Dados
DAG	*Direct Acyclic Graphs* (Grafos Acíclicos Dirigidos)
GC	*Global Classifier* (Classificador Global)
IBM	*International Business Machines*
IG	*Information Gain* (Ganho de Informação)
IPTC	*International Press Telecommunications Council*
IQR	*Interquartile Range* (Intervalo Interquartil)
IR	*Information Retrieval* (Recuperação da Informação)
FAQ	*Frequently Asked Questions* (Perguntas Mais Frequentes)
LCN	*Local Classifier per Node* (Classificador Local por Nó)
LCPN	*Local Classifer per Parent Node* (Classificador Local por Nó Pai)
LCL	*Local Classifier per Level* (Classificador Local por Nível)
MLNP	*Mandatory Leaf Node Predicition* (Predição Obrigatória de Nós-folha)
NER	*Named Entity Recognition* (Reconhecimento de Entidades Mencionadas)
NB	*Naïve Bayes*
NB-M	*Naïve Bayes* Multinomial
NLIDBs	*Natural Language Interfaces for Databases* (Interfaces em Linguagem Natural para Banco de Dados)
NMLNP	*Non-Mandatory Leaf Node Prediction* (Predição Não-Obrigatória de Nós-folha)
NLTK	*Natural Language Toolkit*
NTCIR	*NII-NACSIS Test Collection for IR Systems*

POS	*Part-of-Speech Tagging* (Categorização Gramatical)
NLP	*Natural Language Processing* (Processamento de Linguagem Natural)
ODQA	*Open Domain Question Answering* (Sistemas de QA de Domínio Aberto)
QA	*Question Answering*
RDQA	*Restricted Domain Question Answering* (Sistemas de QA de Domínio Restrito)
SVM	*Support Vector Machines* (Máquina de Vetores de Suporte)
TREC	*Text Retrieval Conference*

Lista de símbolos

$\Uparrow$	Ascendência
$\lvert . \rvert$	Cardinalidade
G	Classe geral
c_i, c_j, c_k	Classes arbitrárias
C	Conjunto de classes
Te	Conjunto de exemplos de teste
Tr	Conjunto de exemplos de treinamento
$\Downarrow$	Descendência
$\cap$	Intersecção de conjuntos
$\Rightarrow$	Implicação material
$\mathcal{M}$	Mediana
$\not\prec$	Não precede
β	Parâmetro de ponderação da F-medida
$\in$	Pertence
$\prec$	Precede
$\forall$	Quantificação universal *(Para qualquer, para todos)*
R	Raiz da árvore
$\sum$	Somatória

Sumário

Introdução

O aumento do poder de processamento dos computadores, juntamente com o crescente volume de conteúdo produzido em diversos meios digitais, tem motivado uma evolução no modo de interação entre os humanos e as máquinas.

No decorrer dos últimos anos, uma diversidade de assistentes virtuais tem sido introduzidao, por exemplo, a *Siri (Apple)*, a *Google Now (Google)* e a *Cortana (Microsoft)*. De acordo com um relatório da Vertio Analytics[1], 42% dos usuários de *smartphones* nos Estados Unidos utilizam seus assistentes virtuais em média dez vezes ao mês, sendo que este número tende a aumentar a medida que os dispositivos são substituídos por versões mais novas. Considerando as taxas de utilização dos *smart speakers*, como o *Google Home* e o *Amazon Echo*, os números são ainda mais impressionantes. De acordo com uma pesquisa do IFTTT[2], em 2017, 60% dos usuários utilizaram seus rádios inteligentes ao menos 4 vezes ao dia.

Esses exemplos demonstram a evolução dos sistemas de *Question Answering* (QA) que datam por volta de 1960 (SIMMONS, 1965; GREEN et al., 1961). Os sistemas atuais buscam resolver tarefas do cotidiano dos usuários, como agendar compromissos, telefonar para alguém, responder sobre o clima e monitorar o preço de ações. Isso também demonstra a evolução no propósito dos sistemas de QA, originalmente propostos para somente responder as perguntas dos usuários utilizando linguagem natural.

As capacidades desses sistemas atuais os enquadram como sistemas de QA de domínio aberto[3], o que resumidamente denota sistemas ilimitados quanto ao número de domínios ou assuntos, mas capazes de produzir respostas simples e menos contextualizadas. Contudo, os desejos e expectativas dos usuários vêm aumentado e, consequentemente, eles esperam receber respostas mais sofisticadas e completas.

Sistemas de QA de domínio restrito, por sua vez, são capazes de produzir respostas mais completas e detalhadas. Porém, eles são limitados quanto ao domínio (assunto) de interesse, enquanto é esperado que os assistentes virtuais sejam os mais gerais possíveis, sendo capazes de produzir respostas sobre os mais variados assuntos.

Para lidar com essa limitação, Chung et al. (2004) apresentam uma estratégia que consiste em combinar uma série de sistemas de domínio restrito em um único sistema de domínio aberto. Assim, quando uma pergunta for apresentada, um mecanismo deve

---

1 *42 Percent of US Smartphone Owners Use AI Personal Assistant Monthly.* Disponível em <https://goo.gl/brxT3H>, acessado em 31/01/2018.

2 *2017 Voice assistant trends.* Disponível em <https://goo.gl/vjuR3m>, acessado em 31/01/2018.

3 Durante este trabalho, no que se refere a sistemas de QA, os termos **aberto** e **geral**, assim como os termos **específico** e **restrito** são empregados indistintamente.

ser capaz de selecionar o sistema de domínio restrito mais apropriado para responder a pergunta.

Considerando a afirmação de Kurzweil (2013), que domínios (assuntos) são naturalmente organizados de forma hierárquica, este trabalho propõe um mecanismo de roteamento implementado através de um **classificador hierárquico de domínio de perguntas**, cujo intuito é identificar o domínio de uma pergunta escrita em linguagem natural. Contudo, a criação e manutenção de um conjunto de dados rotulados para treinar esse classificador é um processo extremamente oneroso, impraticável de ser feito manualmente. Nesse contexto, este trabalho também propõe gerar automaticamente perguntas rotuladas com base nos documentos que poderão servir de fonte de informação para os sistemas de QA. Essa estratégia consiste em transformar sentenças declarativas dos documentos em sentenças interrogativas e propagar os rótulos dos documentos para as perguntas criadas.

Com a abordagem proposta, foi possível criar automaticamente um conjunto de dados com mais de três milhões de perguntas, organizadas em domínios e subdomínios utilizando uma taxonomia tradicionalmente empregada para tópicos de documentos. Esse conjunto de dados foi utilizado na tarefa de treinamento do classificador hierárquico de domínio de perguntas.

A principal motivação para desenvolver esse classificador é torná-lo uma alternativa para a seleção automática de sistemas de QA com base nos domínios das perguntas escritas pelos usuários. Além disso, será possível utilizá-lo em outras aplicações como categorização de perguntas em sistemas de Perguntas Mais Frequentes (FAQs) e portais de perguntas e respostas, por exemplo, o *Quora* e o *Yahoo! Answers*.

Dotado desse mecanismo, espera-se viabilizar a fusão de maneira prática e efetiva de sistemas de domínio aberto e domínio restrito, unindo as principais vantagens de cada categoria em um sistema robusto e unificado de QA. O diagrama da Figura 1 ilustra a arquitetura macro do sistema proposto. A pergunta proposta pelo utilizador é processada e enviada para o classificador de domínio que irá selecionar o sistema de QA de domínio restrito mais relevante para respondê-la (o sistema identificado por **1.4** na Figura 1). Em seguida, o sistema escolhido, com base nos documentos disponíveis como fonte de informação e da pergunta proposta, deverá produzir uma resposta para o utilizador.

Com base no exposto, este trabalho de pesquisa avalia a seguinte hipótese: *é possível criar automaticamente um conjunto rotulado de perguntas que pode ser empregado no treinamento de um classificador hierárquico de domínio*, e com isso permitir a criação de um sistema de *classificação de domínio de perguntas reais*?

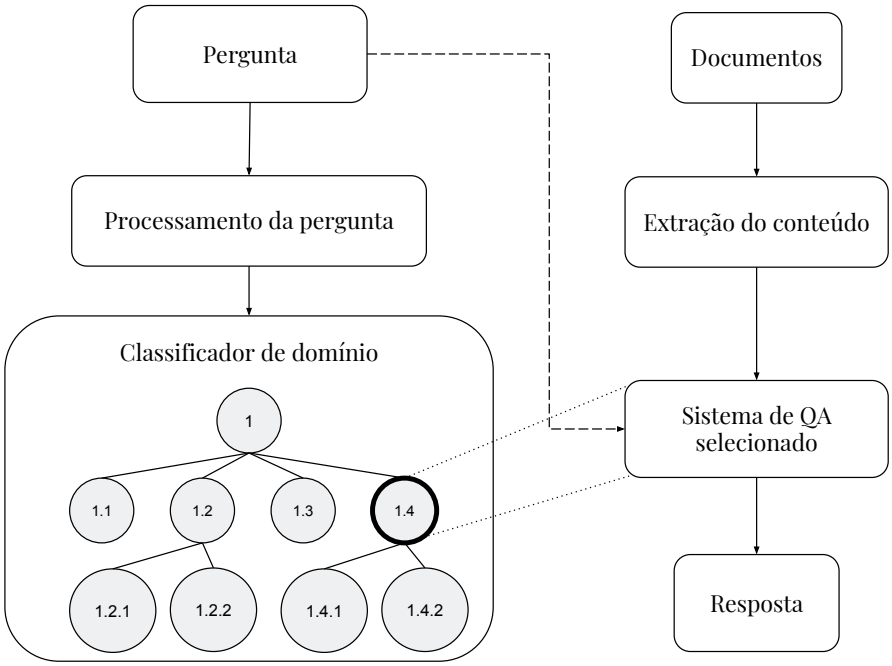


Figura 1 – Diagrama de um sistema de QA de domínio aberto composto por sistemas de QAs de domínio restrito.

Objetivos

Os principais objetivos desse trabalho são:

1. Propor um classificador hierárquico de domínio de perguntas que poderá ser utilizado na integração entre sistemas de QA de domínio aberto e restrito;
2. Propor uma estratégia automática para gerar bases de perguntas hierarquicamente rotuladas quanto ao domínio;
3. Compor e disponibilizar bases de perguntas rotuladas quanto ao domínio para treinamento e avaliação de sistemas de classificação hierárquica de perguntas;
4. Oferecer uma interface simples para o sistema de classificação proposto de tal forma que o usuário possa testá-lo.

Organização

Este trabalho está organizado na seguinte estrutura:

- o Capítulo 1 introduz os sistemas de QA e apresenta uma revisão dos principais trabalhos encontrados na literatura;

- o Capítulo 2 apresenta a proposta de um classificador hierárquico do domínio de perguntas para a integração entre os sistemas de QA de domínio aberto e restrito;
- o Capítulo 3 introduz os conceitos de classificação hierárquica, as diferentes abordagens e suas aplicações na literatura;
- o Capítulo 4 apresenta a geração automática de perguntas rotuladas com base em documentos, descrevendo as ferramentas e estratégias utilizadas;
- o Capítulo 5 descreve as etapas realizadas na execução dos experimentos e validação da hipótese, detalhando a metodologia experimental aplicada e discutindo os resultados obtidos;
- Por fim, o Capítulo 6 conclui a pesquisa realizada e oferece direcionamentos para trabalhos futuros.

1 Sistemas de QA

Conforme explicado por Lu, Cheng e Yang (2012), ao propor o Teste de Turing para julgar se uma máquina é dotada de inteligência, Alan Turing utilizou uma ideia de *chat* entre o humano e a máquina, e consequentemente propôs um protótipo de um sistema de QA (TURING, 1950).

Simmons (1965), notando a necessidade de formalizar esse novo tipo de sistema, propôs uma nova subárea da computação denominada QA *(Question Answering)* que, conforme explicado mais recentemente por Athira, Sreeja e Reghuraj (2013), se ocupa em responder a perguntas realizadas em linguagem natural extraindo respostas assertivas de conjuntos de documentos, sendo ambos, respostas e documentos apresentados em linguagem natural.

Segundo Lu, Cheng e Yang (2012), existem pelo menos quatro categorias de sistemas de QA, sendo elas:

Chatbots Tentam imitar o comportamento humano no decorrer de um diálogo, utilizando correspondência de padrões para produzir respostas apropriadas.

QAs que utilizam bases de conhecimento Baseiam-se em técnicas de inferência para encontrar respostas em grandes bases de conhecimento. Eles são restritos ao conteúdo presente nas bases de conhecimento disponíveis.

QAs baseados em sistemas de recuperação Utilizam as perguntas em linguagem natural proposta pelo usuário para disparar buscas sobre coleções de documentos e páginas na internet, retornando os documentos encontrados para o usuário.

QAs baseados em texto livre Recuperam a informação de documentos e da internet, porém contemplam um módulo adicional para extração da resposta, retornando para o usuário respostas em linguagem natural e não apenas documentos ou páginas.

As categorias apresentadas podem ser estratificadas em duas principais: *(i)* sistemas de QA de domínio restrito *(Restricted Domain Question Answering - RDQA)* e *(ii)* sistemas de QA de domínio aberto *(Open Domain Question Answering - ODQA)* (ALLAM; HAGGAG, 2012).

1.1 Sistemas de QA de domínio restrito (RDQAs)

RDQAs são capazes de responder perguntas apenas sobre determinado assunto (música, cinema, previsão do tempo, entre outros) e envolvem extenso uso de NLP (Proces-

samento de Linguagem Natural) e ontologias específicas para o domínio abordado, sempre visando uma maior contextualização e detalhamento das respostas, como apresentado por Allam e Haggag (2012) e Ramprasath e Hariharan (2012).

Minock (2005) postula que RDQAs têm como características serem delimitados, complexos e práticos, sendo inicialmente abordados pelas limitações computacionais existentes e aplicados como forma de se realizar consultas em linguagem natural em bases de dados *(Natural Language Interfaces for Databases - NLIDBs)*. Detalhando as três características postuladas por Minock (2005), entende-se que:

Delimitado um RDQA deve ser focado em determinado assunto de interesse, sendo que seu conteúdo deve ser composto por fatos e possuir certo grau de detalhamento.

Complexo um RDQA deve ainda conter diversas entidades inerentes ao domínio de interesse, sendo que essas entidades devem possuir diversos atributos e diversas relações entre si.

Prático um RDQA deve, por fim, responder perguntas que sejam relevantes para determinado grupo de usuários, e permitir que a aquisição e manutenção dos dados seja viável.

É evidenciado por Lu, Cheng e Yang (2012) e Mollá e Vicedo (2007) que a principal limitação de RDQAs é que eles não conseguem produzir respostas para perguntas cujo conteúdo não esteja representado nas bases de conhecimento.

1.2 Sistemas de QA de domínio aberto (ODQAs)

ODQAs são capazes de abranger perguntas e respostas sobre diversos assuntos, utilizando fontes universais de informação, por exemplo, enciclopédias. Dessa forma, demandam um esforço menor de preparação do conteúdo, conforme descrito por Lu, Cheng e Yang (2012).

Essa amplitude de conteúdo manifesta a característica suprema do domínio aberto — **a universalidade de tópicos**. Em contrapartida, produzem respostas menos contextualizadas e completas que RDQAs.

Conforme explicado por Mollá e Vicedo (2007) e Indurkhya e Damerau (2010), ODQAs eram principalmente focados em responder perguntas relacionadas a entidades simples e perguntas factuais, as quais podem ser expressadas em poucos termos, tais como: *"Where is the Taj Mahal?"* ou *"Who is Walt Disney?"*. Posteriormente, eles evoluiram para poder responder também a perguntas em formato de listas, como *"What movies did James Dean appear in?"*, e que envolvam relações entre as entidades, como *"What is the relation between Donald Trump and Barack Obama?"*.

ODQAs são capazes de produzir respostas mais sucintas e precisas, por outro lado, RDQAs produzem respostas mais longas, completas e contextualizadas, sendo portanto, mais expressivas para os usuários (MOLLÁ; VICEDO, 2007).

Segundo Minock (2005), usuários já estão confortáveis com a utilização de buscas baseadas em palavras-chave, sendo possível encontrar respostas semelhantes às produzidas por ODQAs. Portanto, a utilização de QA em domínios abertos pode não ser tão proveitosa quanto em domínios restritos.

Idealmente, um sistema de QA deveria possuir a *universalidade e precisão* do domínio aberto aliadas aos *detalhes e expressividade* do domínio restrito.

1.3 Arquitetura de sistemas de QA

A arquitetura geral de sistemas de QA é demonstrada na Figura 2, na qual se destacam os três módulos principais: análise de perguntas, recuperação de informação e extração de respostas (INDURKHYA; DAMERAU, 2010; LU; CHENG; YANG, 2012; ATHIRA; SREEJA; REGHURAJ, 2013).

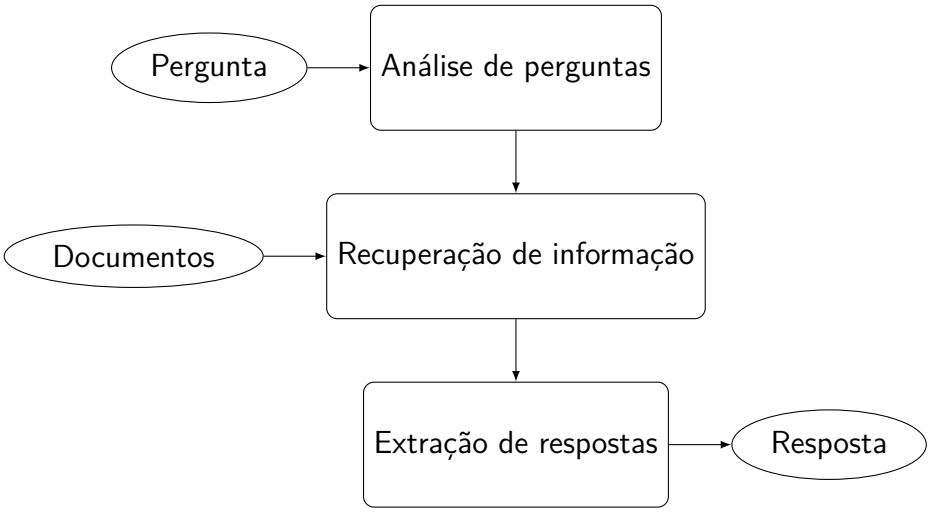


Figura 2 – Arquitetura padrão de sistemas de QA. Adaptado de Indurkhya e Damerau (2010).

Análise de perguntas: Módulo de interpretação das perguntas realizadas em linguagem natural, o qual aplica técnicas de NLP para tratamento das perguntas, preparando-as para o estágio subsequente.

Recuperação de informação: Módulo responsável por apresentar as informações coletadas de diversas fontes em formato adequado para associação com os resultados do módulo anterior, realizando a seleção dos fragmentos mais relevantes dos documentos, utilizando técnicas de NLP. Esse módulo também pode utilizar ontologias para complementar as fontes principais de informação.

Extração de respostas: Módulo encarregado de realizar a seleção da resposta mais apropriada entre as candidatas e a construção da resposta em linguagem natural como resultado da interação dos dois módulos anteriores.

1.4 Evolução dos sistemas de QA

As primeiras iniciativas de sistemas de QA apresentam-se como NLIDBs *(Natural Language Interfaces for Databases)*, como o *BASEBALL*, desenvolvido por Green et al. (1961), o qual respondia perguntas sobre jogos de beisebol ocorridos em uma temporada da liga americana, e o *LUNAR*, apresentado por Woods (1973), que respondia perguntas sobre análises de amostras rochosas das missões lunares Apollo.

Pode-se classificar o *LUNAR* e o *BASEBALL* como sistemas que realizavam a conversão das consultas do utilizador, realizadas em linguagem natural, em consultas para bases de dados previamente preparadas. Eles também podem ser classificados como RDQAs *(Restricted Domain Question Answering)* por abordarem apenas um tópico de interesse.

Brown e Burton (1975) propuseram a utilização de um método baseado em um conjunto predefinido de padrões de perguntas, a fim de extrair os fragmentos essenciais e, em seguida, utilizá-los para recuperar o conteúdo relevante para produção da resposta. Por sua vez, a primeira utilização de documentos textuais para extração das respostas foi apresentada por Lehnert (1977).

Um avanço ocorreu após intensa pesquisa entre as décadas de 1970 e 1980, a fim de utilizar os sistemas de QA para viabilizar testes das hipóteses propostas na área de NLP. Em decorrência deste avanço, Wilensky et al. (1988) propuseram um sistema de ajuda ao usuário, restrito ao domínio do sistema operacional UNIX. Ele apresentava uma representação formalizada das perguntas do utilizador, seguindo abordagem diferente dos primeiros sistemas de QA, que nada mais eram que interfaces em linguagem natural para bases de dados. Esse sistema também foi o primeiro QA focado em se adequar aos objetivos e experiência do usuário (MOLLÁ; VICEDO, 2007).

A tarefa de QA foi considerada extremamente custosa computacionalmente e sofreu um breve período de hiato durante a década de 1980. Sendo retomada na década de 1990 com o surgimento de conferências, como o TREC[1], que eram focadas em sistemas de QA, NLP e IR *(Information Retrieval)*. O avanço do poder computacional, as novas bases de conhecimento, perguntas e novos métodos de avaliação fomentaram a retomada das pesquisas, principalmente enfatizando as tarefas de QA de domínio aberto (VOORHEES, 2001).

[1] *Text Retrieval Conference.*

Kupiec (1993) propôs o *MURAX*, que é considerado um dos pioneiros na tarefa de QA de domínio aberto, pois utilizava a enciclopédia Grolier (1990) como fonte de informação. Kupiec (1993) combinou técnicas de processamento de linguagem com recuperação de informação para elaboração do sistema.

Um estudo realizado por Clarke e Terra (2003) demonstra um comparativo entre as estratégias de recuperação de fragmentos ou recuperação de documentos a fim de extrair as informações para responder perguntas factuais, sugerindo um melhor desempenho quando se utiliza a recuperação do documento completo. Porém, nesse caso, um pós-processamento para extração do fragmento que contém a resposta se faz muitas vezes necessário.

Lin et al. (2003) realizaram uma análise a fim de determinar quão detalhadas devem ser as repostas produzidas por um sistema de QA. Eles concluiram que os utilizadores preferem uma resposta com a dimensão de um parágrafo[2], contendo informações sobre o contexto da resposta em detrimento de uma resposta direta de apenas um termo ou uma mais longa que um parágrafo.

Com respeito aos RDQAs, Chung et al. (2004) abordaram o domínio climático, numa forma de interação semelhante a uma assistente virtual rudimentar. O sistema apresentado possuía uma ontologia elaborada manualmente para viabilizar a produção das respostas. Foi ainda apresentada uma proposta de utilização de um classificador de domínio a fim de se roteirizar as perguntas para instâncias mais específicas de sistemas de QA.

Ainda sobre domínios restritos, cita-se o domínio biográfico abordado por Tsur, Rijke e Sima'an (2004), que utilizou um conjunto limitado de tipos de perguntas, restringindo-se a questões do tipo *"Quem"*; o domínio de engenharia aeronáutica abordado por Diekerma, Yilmazel e Liddy (2004), os quais propuseram também as métricas de avaliação para sistemas de domínio restrito; e o domínio do turismo por Benamara (2004), que integra as capacidades de representação do conhecimento com procedimentos avançados de inferência.

Nyberg et al. (2005) apresentaram uma extensão para transformar sua proposta inicial do *JAVELIN* (2003), a qual era focada em domínio aberto, em um RDQA. As modificações incluem adição de recuperação semântica e extração de fatos e inferências, aliados às técnicas já existentes na proposta original: recuperação baseada em palavras-chave, extração de fragmentos e conciliação dos textos das respostas com o tipo de resposta esperada (NYBERG; FREDERKING, 2003).

Minock (2005) enunciou as características desejadas para RDQAs, postulando que os sistemas devem ser **delimitados**, **complexos** e **práticos**.

Conforme afirmaram Mollá e Vicedo (2007), o surgimento de outras conferências

[2] A resposta fornecida ao utilizador deve ser composta pela resposta exata para a pergunta proposta em conjunto com o parágrafo no qual a resposta estava inserida.

como CLEF[3] e a NTCIR[4] contribuiu para a evolução da pesquisa na área de QA. Eles declaram também que pesquisas focadas em RDQAs podem fomentar o surgimento de técnicas que não floresceriam com pesquisas voltadas apenas ao domínio aberto (VALLIN et al., 2005; KANDO, 2005).

Por sua vez, Frank et al. (2007) propuseram uma arquitetura híbrida e modular combinando fontes de informação estruturadas, semi-estruturadas e não-estruturadas, reutilizando técnicas aplicadas às NLIBDs, porém, agregando ontologias. Essa abordagem visa combinar as melhores características de RDQAs e ODQAs, tendo os resultados experimentais iniciais sugerido que, em caso de conflito na elaboração das respostas, deve-se optar pela resposta de caráter restrito. O *Watson*, um sistema apresentado pela IBM em 2011, pode ser classificado como um ODQA, porém, demanda um enorme poder computacional[5].

Miao, Su e Li (2010) abordaram o problema da pouca quantidade de informação das perguntas realizadas em linguagem natural. Como solução, foi proposta a expansão das perguntas utilizando os conceitos existentes. Os resultados mostraram que sistemas de QA podem se beneficiar dessa estratégia já utilizada em outras áreas de IR.

Mais recentemente, Kumar et al. (2016) propuseram um modelo utilizando uma rede neural com memória dinâmica, aplicando-o em diversas tarefas de NLP. Através de uma abordagem iterativa, ele ativa uma memória episódica, a qual apresenta a capacidade de inferência baseada em iterações anteriores. No mesmo momento, mais um RDQA foi apresentado por Chaudhri, Overholtzer e Spaulding (2015), no domínio da biologia, o *Inquire Biology*. Seguindo trabalhos anteriores, esse sistema utiliza uma ontologia elaborada manualmente por especialistas.

Por fim, as mais recentes manifestações de sistemas de QA são os diversos assistentes virtuais de *smartphones*, como a *Siri*, a *Google Now* e a *Cortana*, e os ainda mais atuais *smart speakers*, como o *Google Home* e o *Amazon Echo*.

1.5 Considerações finais

Este capítulo apresentou o propósito dos sistemas de QA bem como sua evolução, a qual demonstra o aperfeiçoamento desses sistemas às necessidades dos usuários, tanto em conteúdo como em usabilidade. Foram evidenciadas as principais características das categorias existentes, destacando as diferenças das duas principais: RDQAs e ODQAs. Também foi apresentada uma visão da arquitetura desses sistemas.

[3] *Cross Language Evaluation Forum.*
[4] *NII-NACSIS Test Collection for IR Systems.*
[5] *"Is Watson the smartest machine on earth?"*, disponível em: <http://www.theguardian.com/technology/2011/feb/17/ibm-computer-watson-wins-jeopardy>. Acessado em: 28/02/2018.

Como evidenciado ao longo do capítulo, não existe uma solução definitiva que compartilha as vantagens das duas categorias, sendo que ambas têm suas qualidades e deficiências. Porém, como comentado por Chung et al. (2004), seria benéfico um cenário híbrido, no qual as características de ambos os sistemas fossem unificadas. Para isso, é necessária uma estratégia que permita combinar essas características através de um mecanismo de seleção de sistemas de QA.

2 Classificação de domínio de perguntas

Sistemas de QA de domínio restrito (RDQAs) são especializados por determinado assunto de interesse, sendo capazes de produzir respostas contextualizadas e completas. Para isso, os RDQAs renunciam a capacidade de abordar diversos assuntos, sendo essa, a característica de outra categoria de sistemas de QA, os de domínio aberto (ODQAs). Contudo, eles oferecem respostas privadas de contextualização e completude (ALLAM; HAGGAG, 2012).

Essa distinção deve-se ao fato de que sistemas de domínio restrito podem recorrer a fontes de informação especializadas, ontologias e outros recursos. Além disso, as implementações desse tipo de sistema podem utilizar técnicas mais sofisticadas para a interpretação das perguntas, possibilitando produzir uma resposta de melhor qualidade (MOLLÁ; VICEDO, 2007).

Para exemplificar, a pergunta *"What caused the bank collapse last monday morning?"* ao ser respondida por um RDQA relacionado a finanças irá produzir uma resposta diferente de um RDQA relacionado à geografia. Respectivamente no domínio de finanças e geografia, as respostas poderiam ser: *(i) "The major cause of the bankruptcy of the Dealer Brother's Bank was the huge amount of withdrawals during ..."* e *(ii) "During monday's intense rain, the Elk River level increased incredibly fast resulting in a superflow for 21 hours, further, collapsing the banks close by the Hooper Dam, the Civil Defense engineers are already on site for damage ... "*. Neste exemplo, a expressão *"bank collapse"* foi interpretada de formas diferentes dependendo do contexto a que se refere. O sistema de QA de **domínio restrito** relacionado a finanças interpretou a expressão como sendo a falência de uma instituição financeira, enquanto o RDQA relacionado a geografia interpretou a expressão como sendo um deslizamento de terra.

Enquanto sistemas de domínio restrito produzem respostas como as exemplificadas no exemplo anterior, sistemas de QA de **domínio aberto** podem produzir respostas como *"Withdraws"* ou *"Heavy rain"*, igualmente corretas, porém, muito sucintas e objetivas. Respostas mais completas e contextualizadas, com dimensão de, em média, um parágrafo como as produzidas pelos RDQAs, são preferidas pelos utilizadores em detrimento de apenas se apresentar a resposta exata e não contextualizada (LIN et al., 2003).

Outra questão importante a ser considerada é a identificação do assunto abordado pela pergunta. Sistemas de QA de domínio aberto, pela sua natureza, podem produzir respostas relacionadas a assuntos diferentes do assunto da pergunta, resultando em uma resposta inadequada. Por exemplo, um ODQA poderia interpretar erroneamente a pergunta *"What caused the bank collapse last monday morning?"* como sendo uma pergunta de

geografia, enquanto a intenção do utilizador era realizar uma pergunta sobre finanças. Esse tipo de problema ocorre, pois esse tipo de sistema não possui recursos para contextualizar o domínio da pergunta. O mesmo problema não aconteceria com RDQAs, pois não faria sentido realizar uma pegunta sobre finanças para um sistema específico de geografia.

A proposta deste trabalho é oferecer uma solução para combinar as principais vantagens dos diferentes tipos de sistemas de QA (**domínio restrito** e **domínio aberto**). Com isso, é almejada a construção de um sistema unificado, capaz de abordar diversas áreas do conhecimento, vantagem apresentada pelos sistemas de QA de domínio aberto, e ainda produzir respostas mais elaboradas, nos quais sistemas de domínio restrito são mais eficientes.

A estratégia, conforme brevemente sugerido por Chung et al. (2004), consiste no fato de que diversos sistemas de domínio restrito podem ser agrupados, de forma a se apresentar como módulos de um sistema de QA integrado, desde que exista uma forma de selecionar qual desses módulos é o mais adequado para responder as perguntas propostas. Logo, surge a necessidade do desenvolvimento deste mecanismo de seleção.

Visto que cada sistema de domínio restrito aborda uma ou poucas áreas de interesse, é necessária uma forma de categorizar e sistematizar essas áreas. Conforme explicado por Kurzweil (2013), os humanos tendem naturalmente a organizar conceitos e objetos de forma hierárquica.

> *"O próprio mundo é intrinsecamente hierárquico: árvores contêm galhos; galhos contêm folhas; folhas contêm veios. Prédios contêm andares; andares contêm cômodos; cômodos contêm portas, janelas, paredes e pisos."* (KURZWEIL, 2013, p. 11).

Da mesma forma, assuntos também podem ser agrupados hierarquicamente como uma árvore de domínios. Com isso, a característica hierárquica da árvore permite especializar ou generalizar determinado assunto navegando pela hierarquia, respectivamente, ao se aproximar de seus nós-folha ou da raiz. A Figura 3 apresenta uma abstração dessa ideia, enquanto a Figura 4 exemplifica a estrutura da árvore utilizando assuntos triviais.

Na organização em árvore, cada um dos nós está relacionado a um sistema de QA de domínio restrito, o qual será potencialmente selecionado ao se submeter uma pergunta que se enquadre nos assuntos por eles abordados. Dessa forma, esse mecanismo direcionará a pergunta proposta para o sistema de QA que tenha maior probabilidade de produzir uma resposta relevante.

Ao selecionar um nó mais próximo da raiz, entende-se que o sistema mais adequado para produzir a resposta é um sistema menos especializado do que ao selecionar um nó próximo às folhas. Supondo os assuntos abordados na Figura 4, a pergunta *"Who is the*

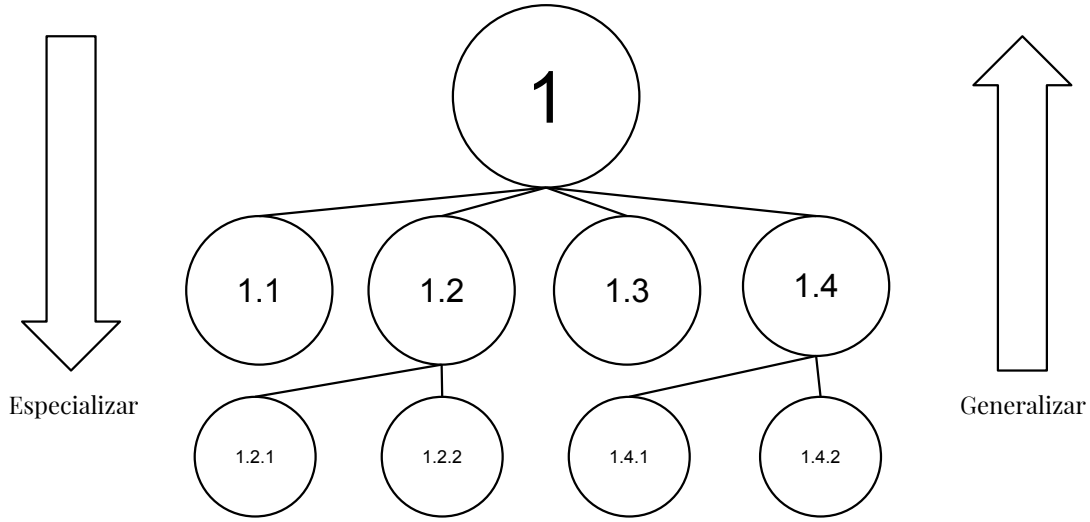


Figura 3 – Abstração dos conceitos de especialização e generalização sobre uma hierarquia de domínios. O nó raiz **1** representa o maior grau de generalização de determinado assunto, enquanto seus filhos (**1.1, 1.2, 1.3 e 1.4**) são versões mais especializadas do assunto abordado por seu pai, assim como os nós **1.2.1, 1.2.2, 1.4.1 e 1.4.2** são versões ainda mais especializadas de seus antecessores.

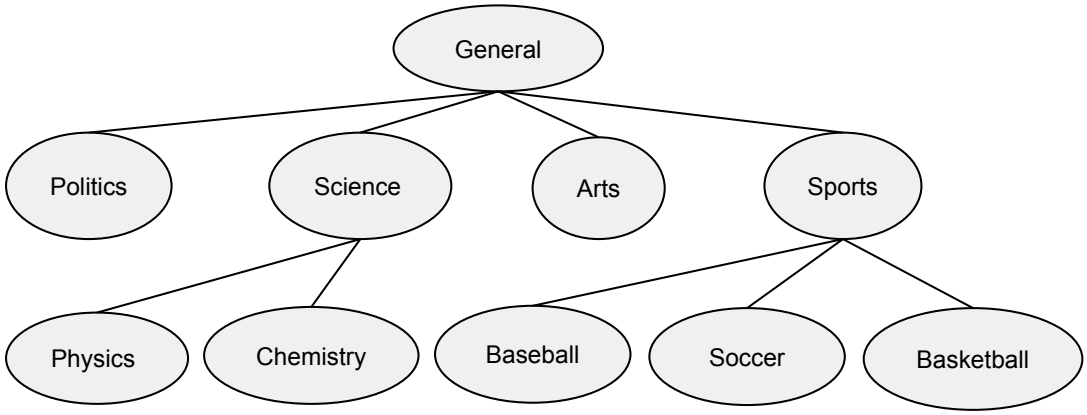


Figura 4 – Exemplificação dos conceitos de generalização e especialização utilizando assuntos triviais. O nó raiz **General** representa o maior grau de generalização do domínios abordados (sistema de QA de domínio aberto), enquanto seus filhos (**Politics, Science, Arts e Sports**) são assuntos mais especializados e, por sua vez, possuem sistemas de QA de domínio restrito associados. Da mesma forma, os nós **Physics e Chemistry** são versões mais especializadas de seu antecessor **Science**.

most profitable player in the PGA tournaments of the current decade?" combina melhor com o nó **"Sports"**, logo um sistema de QA relacionado a esse assunto é o mais adequado para produzir a resposta. A escolha do nó **"Sports"** deve-se ao fato de não existir nenhum nó-folha (sistema de QA de domínio restrito) mais adequado para essa pergunta.

Ainda utilizando a taxonomia da Figura 4, considere a pergunta: *"Which team*

scored more 3-points during the playoffs of 2017-18 season?" que é mais especializada e relacionada a um dos assuntos abordados nos nós-folha disponíveis. Nesse caso, o nó mais relacionado ao assunto é **"Basketball"**, logo o melhor sistema para responder essa pergunta seria o sistema de QA relacionado ao assunto de basquete.

De forma oposta, considere a pergunta: *"How the rise of Bitcoin influenced global economy, specially countries with high inflation rates like Venezuela?"*. Ela não se enquadra em nenhum dos nós mais especializados, podendo ser melhor associada com o nó raiz, no caso o nó **"General"**.

Para que esse sistema hierárquico de domínios funcione, um *roteirizador* de perguntas deve ser capaz de selecionar um entre os RDQAs que compõem o sistema integrado. Para isso, ele deve considerar os assuntos abordados por cada um desses RDQAs e também o assunto abordado pela pergunta submetida.

Na fase de operação, uma pergunta será inicialmente formulada pelo usuário. Ela, então, será classificada entre um dos domínios (assuntos) disponíveis. Em seguida, o RDQA relacionado a esse domínio receberá a mesma pergunta, consultando suas fontes de documentos e ontologias para respondê-la. Essa solução é representada pelo diagrama da Figura 1.

Duas características importantes devem ser consideradas: *(i)* o mecanismo de roteirização hierárquico deve usar fontes de documentos igualmente utilizadas pelos RDQAs de forma automática, dessa forma, alinhando e delimitando o conteúdo dos domínios abordados pelo classificador com os sistemas de QA e *(ii)* deve ser possível sua expansão e atualização à medida que novos domínios forem incorporados e que os domínios abordados sofram atualizações.

2.1 Considerações finais

Este capítulo exemplificou as duas principais categorias de sistemas de QA em um cenário de aplicação. Foi manifestada a ideia de produzir um único sistema híbrido, que combina as vantagens de ambos.

Desenvolveu-se ainda a ideia, com base na afirmação de Kurzweil (2013), que o formato organizacional assumido pelas classes nos classificadores hierárquicos, os colocam como uma solução interessante nas tarefas de organizar e selecionar um domínio sistematicamente para as perguntas.

O capítulo define parte essencial da solução proposta: **um classificador hierárquico de domínio de perguntas para seleção de sistemas de QA.** Também é apresentado um mecanismo roteirizador, na forma de um classificador hierárquico de domínios que viabiliza a solução integrada de utilizar vários RDQAs para compor um

ODQA unificado.

O capítulo a seguir apresenta as principais características de cada abordagem de classificação hierárquica e, ainda, elege a abordagem que melhor corresponde com o problema de classificação hierárquica de perguntas.

3 Classificação hierárquica

Na área de Aprendizado de Máquina (AM), conforme apresentado por Kotsiantis, Zaharakis e Pintelas (2007), existe a tarefa de aprendizado supervisionado que consiste em construir um modelo capaz de atribuir rótulos *(labels)* dado um conjunto de características *(features)*. O modelo criado deve ser capaz de generalizar uma hipótese e, portanto, predizer os rótulos de amostras[1] futuras das quais se conhece apenas as características e se desconhece o rótulo.

Como subdivisão dos problemas de aprendizado supervisionado, existem duas categorias: **regressão**, quando a saída da predição é um valor real *(valores contínuos)*, e **classificação**, quando a saída pertence a uma categoria *(valores discretos)*. Exemplos dessas duas categorias de problemas são respectivamente, predizer o preço de venda de uma casa ou classificar um e-mail como sendo *spam* ou não.

Problemas de classificação hierárquicos são versões particulares de problemas de classificação nos quais as categorias estão organizadas hierarquicamente, diferente dos cenários *flat*[2] nos quais a organização hierárquica inexiste. Conforme apresentado por Wu, Zhang e Honavar (2005), essa organização hierárquica em forma de árvore pode ser visualizada como um conjunto parcialmente ordenado $(C, \prec)$, no qual C representa todas as categorias pertinentes ao domínio e $\prec$ representa a relação de precedência do tipo **"É UM"**.

Posteriormente, a definição foi reexaminada por Silla Jr. e Freitas (2011), indicando que para um problema ser de classificação hierárquica, a taxonomia das classes deve atender a quatro regras:

1. Um único elemento "principal" R é a raiz da árvore.
2. $\forall c_i, c_j \in C, \text{se } c_i \prec c_j \text{ então } c_j \not\prec c_i$
3. $\forall c_i, \in C,\ c_i \not\prec c_i$
4. $\forall c_i, c_j, c_k \in C,\ c_i \prec c_j \text{ e } c_j \prec c_k \ \Rightarrow\ c_i \prec c_k$

A regra 1 indica que existe apenas uma única categoria que precede todas as outras na definição hierárquica, sendo representada pela raiz R da árvore. Esta categoria é a que possui o maior grau de abstração entre as categorias, sendo esse rótulo passível de

[1] O termo **amostra** é utilizado de forma diferente do conceito estatístico, neste trabalho, uma amostra representa **uma única** entidade, **um único** exemplo, enquanto no sentido estatístico amostra denota um conjunto de exemplos selecionados de uma população.

[2] Neste trabalho, o termo **flat** é utilizado nos cenários de classificação não hierárquica.

atribuição a qualquer outra categoria subsequente na hierarquia, ou seja, qualquer entidade rotulada como uma categoria mais especializada pode também ser rotulada com o mesmo rótulo da raiz R (WU; ZHANG; HONAVAR, 2005; SILLA JR.; FREITAS, 2011).

A regra 2 indica que as relações são **assimétricas**, ou seja, se uma categoria c_i possui uma relação de precedência do tipo "É UM" com uma segunda categoria c_j, a segunda não pode possuir a mesma relação com a primeira. Para exemplificar, suponha a existência das categorias "Cachorro" e "Animal". Se existe a relação *"Cachorro* **é um** *Animal"*, logo, pela regra, não pode existir uma relação *"Animal* **é um** *Cachorro"*. A existência dessa última relação degeneraria a estrutura hierárquica, implicando incorretamente que todos "Animais" são "Cachorros" (SILLA JR.; FREITAS, 2011).

A regra 3 indica que uma categoria c_i não pode preceder hierarquicamente a própria categoria c_i, aplicando o princípio de **anti-reflexividade**. Essa regra, em conjunto com a regra 2 impede que a organização se torne um grafo cíclico, o que corromperia a estrutura hierárquica (WU; ZHANG; HONAVAR, 2005; SILLA JR.; FREITAS, 2011).

Por fim, a regra 4 indica o princípio de **transitividade**, ou seja, se existe uma relação de precedência, na qual c_i precede c_j e, uma segunda, na qual c_j precede c_k, implicitamente implica-se a existência de uma relação na qual c_i precede c_k. Para exemplificar, suponha a existência das classes "Cachorro", "Animal" e "Ser vivo", e também que existam as relações *"Animal* **é um** *Ser vivo"* e *"Cachorro* **é um** *Animal"*, logo a relação *"Cachorro* **é um** *Ser vivo"* existe implicitamente, podendo ser inferida navegando pela hierarquia (SILLA JR.; FREITAS, 2011).

Essencialmente, a organização hierárquica se torna mais especializada ao se percorrer a estrutura partindo da raiz para os nós-folha, sendo que cada nível que compõe essa estrutura representa classes em diferentes níveis de abstração (WU; ZHANG; HONAVAR, 2005).

Conforme descrito por Silla Jr. e Freitas (2011), além das configurações da taxonomia em árvore, existem configurações baseadas em Grafos Acíclicos Dirigidos (*Direct Acyclic Graphs* - DAG). Sendo que a principal diferença entre elas é que um DAG permite que as categorias possuam mais de um antecessor, enquanto nas configurações baseadas em árvore isso não é possível. A Figura 5 ilustra um exemplo baseado em DAG, enquanto a Figura 6 ilustra um exemplo baseado em árvore.

Figura 5 – Exemplo de configuração das categorias em forma de DAG. Os círculos representam as categorias. Nota-se que a configuração em DAG permite que uma categoria tenha mais de um ancestral (*e. g.,* E que possui B e C como ancestrais e G que possui D e E como ancestrais).

A escolha da configuração em DAG ou árvore é particular de cada cenário de

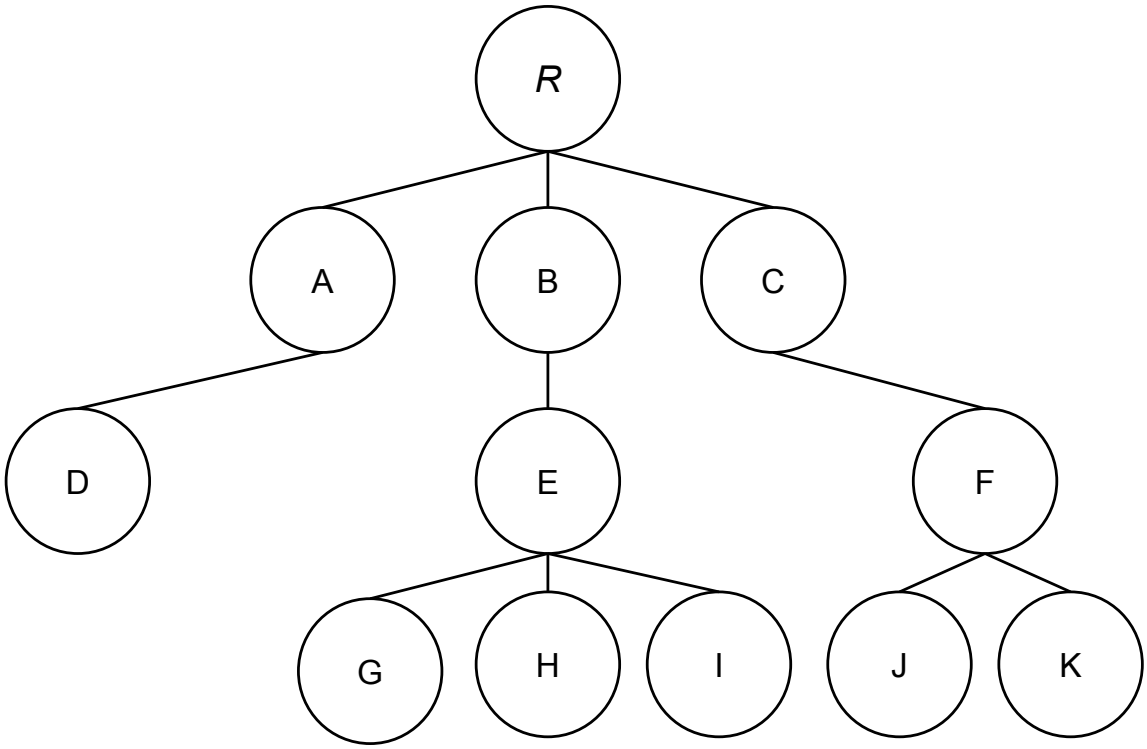


Figura 6 – Exemplo de configuração das categorias em forma de árvore. Os círculos representam as categorias. Nota-se que a configuração em árvore não permite que uma categoria tenha mais de um ancestral, como ocorre na configuração DAG.

utilização, sendo a escolha de árvore mais trivial, dada sua simplicidade. No cenário de classificação hierárquica de domínio de textos, em geral, uma subcategoria é uma especialização de uma única categoria (um único tópico de interesse), e não múltiplos. Dessa forma, conforme demonstrado por Silla Jr. e Freitas (2011), é mais conveniente utilizar uma configuração em árvore.

Uma outra particularidade dos classificadores hierárquicos, como introduzido por Sun e Lim (2001) e Freitas e Carvalho (2006), é quanto à obrigatoriedade de se predizer um nó-folha, o que é conhecido como *Mandatory Leaf Node Prediction* (MLNP) ou encerrar a predição em alguma classe intermediária, o que por sua vez é conhecido como *Non-Mandatory Leaf Node Prediction* (NMLNP), sendo essa decisão específica a cada cenário de aplicação.

Nos casos não-obrigatórios, o tratamento trivial é a utilização de limiares *(threshold)*. Ao se ultrapassar o valor do limiar, a predição segue para o próximo nível até encontrar um limiar que não possa ser ultrapassado ou se atinja um nó-folha.

Os classificadores hierárquicos foram organizados por Silla Jr. e Freitas (2011) em quatro abordagens, visto as diferenças na configuração em relação à taxonomia hierárquica:

LCN Classificadores locais por nó *(Local Classifier per Node)*.

LCPN Classificadores locais por nó pai *(Local Classifier per Parent Node)*.

LCL Classificadores locais por nível *(Local Classifier per Level)*.

CG Classificadores globais *(Global Classifier)*.

3.1 Classificadores locais por nó (LCN)

A abordagem LCN utiliza um classificador binário por nó, para cada um dos nós que compõem a taxonomia, exceto o nó raiz. O classificador é binário, pois sua tarefa consiste em indicar somente se a amostra pertence ou não à classe associada a ele.

Nessa abordagem, é necessário definir uma estratégia adicional para determinar quais amostras que pertencem às classes positivas e negativas dos nós não-folha (devendo considerar a existência das relações "É UM" e a Regra 4).

A Figura 7 indica um exemplo de configuração dos classificadores em relação à taxonomia utilizando essa abordagem. A abordagem LCN apresenta a vantagem de ser naturalmente *multilabel*[3], visto que cada um desses classificadores tem potencial de atribuir o rótulo positivo (a classe que o classificador representa) à amostra, em contrapartida, podem existir inconsistências.

As inconsistências podem acontecer em cenários que não sejam *multilabel*, pois não existe nenhuma restrição que obrigue os classificadores a respeitar a taxonomia hierárquica definida. Ou seja, diferentes classificadores associados a nós apresentados em subárvores distintas (logo, relacionados à classes desconexas) podem atribuir seu rótulo à amostra. Utilizando a Figura 7 como exemplo, uma amostra poderia ser associada com as categorias **1**, **2.1.1** e **2.1.2** e não estar associada com a categoria **2.1**.

Existe ainda, a desvantagem de existir um grande número de classificadores, dado que cada classe possuí um classificador, o que pode demandar alto esforço computacional para treinar todos os classificadores.

Na literatura, alguns trabalhos utilizam a abordagem LCN no contexto de categorização de textos. Por exemplo, D'Alessio et al. (2000) utilizaram um algoritmo guloso, sobre a base de notícias Reuters-21578[4], que explora a hierarquia um nível por vez, incluindo categorias intermediárias quando existir a possibilidade de ganho de desempenho, visando dessa forma atingir ao menos o desempenho de um classificador *flat*. Ainda na base Reuters-21578, Sun e Lim (2001) e Sun, Lim e Ng (2003) apresentaram uma abordagem

[3] Problemas *multilabel* permitem múltiplos rótulos associados a uma amostra.

[4] Reuters-21578 Text Categorization Collection Data Set. Disponível em <https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection>, acessado em: 20/04/2018.

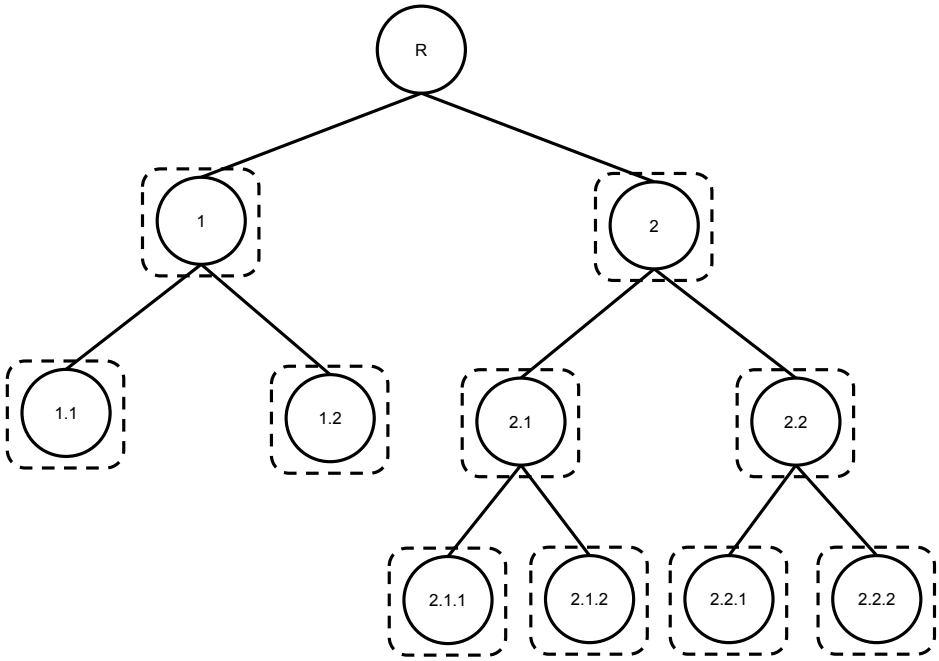


Figura 7 – Configuração dos classificadores binários em relação a taxonomia utilizando a abordagem LCN. Os círculos representam as classes e os quadrados tracejados representam os classificadores. Adaptado de Silla Jr. e Freitas (2011).

NMLNP, que permitia predizer categorias intermediárias da taxonomia e não somente categorias associadas a nós-folha. Ainda considerando a abordagem escolhida, eles propuseram métricas de avaliação baseadas em similaridade e distância.

Mladenić e Grobelnik (2003) apresentaram um problema baseado em documentos coletados da *internet*, utilizando sequências de palavras para representá-los, em vez de utilizar a abordagem tradicional que considera palavras isoladas. Em sequência, Sun et al. (2004) analisaram o problema do bloqueio[5] nos classificadores hierárquicos, propondo uma métrica para mensurar o fator de bloqueio e algumas estratégias para endereçar esse problema.

Cesa-Bianchi, Gentile e Zaniboni (2006a) analisaram o problema da propagação do erro de predição, aplicando uma estratégia de cálculo de erro para reavaliar o desempenho dos classificadores. Em seguida, Cesa-Bianchi, Gentile e Zaniboni (2006b) apresentam um novo algoritmo com características incrementais e desempenho próximo as Máquinas de Vetores de Suporte (*Support Vector Machines* - SVM) hierárquicas.

Esuli, Fagni e Sebastiani (2008) propuseram uma extensão do *AdaBoost.HM* proposto por Schapire e Singer (2000) para considerar as categorias hierárquicas. Jin et al. (2008) aplicaram a abordagem no contexto de classificação de literatura biomédica,

[5] Amostras incorretamente não atribuídas às categorias mais altas da hierarquia não seguem para os classificadores situados em níveis mais abaixo, *bloqueando* a classificação seguinte.

porém utilizaram uma variação da abordagem LCN em forma de grafos, através da representação de uma ontologia de genes de forma textual.

Punera e Ghosh (2008) apresentaram uma estratégia para aumentar o desempenho dos classificadores aplicando uma etapa de pós-processamento. Xue et al. (2008) exemplificaram um cenário de classificação de notícias com múltiplos níveis de profundidade. Mais recentemente, Bennett e Nguyen (2009) apresentaram uma estratégia que altera o balanceamento das classes na fase de treinamento para minimizar a propagação do erro de classificação.

3.2 Classificadores locais por nó pai (LCPN)

A abordagem LCPN utiliza um classificador para cada um dos nós não-folha presentes na taxonomia, inclusive o nó raiz. Dessa forma, cada classificador é *multiclasse*[6] e irá predizer um dos nós-filhos. A Figura 8 indica um exemplo de configuração dos classificadores em relação à taxonomia utilizando essa abordagem.

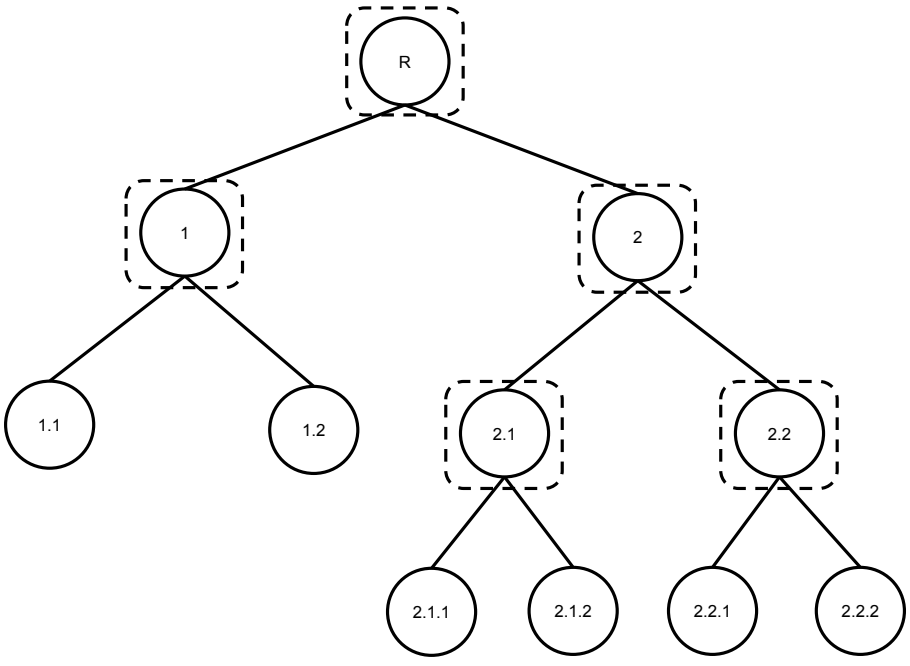


Figura 8 – Configuração dos classificadores *multiclasse* em relação a taxonomia utilizando a abordagem LCPN. Os círculos representam as classes e os quadrados tracejados representam os classificadores, sendo que estes irão predizer as respectivas classes-filha. Adaptado de Silla Jr. e Freitas (2011).

[6] Problemas *multiclasse* permitem apenas um rótulo por amostra, diferente de problemas *multilabel* que permitem múltiplos.

Assim como a abordagem LCN, a abordagem LCPN é *multilabel*, porém, é formada por menos classificadores do que a primeira, visto que os classificadores são *mutliclasse*. Mesmo assim, o problema das inconsistências em cenários que não sejam *multilabel* ainda persiste, visto que nenhum controle é aplicado para restringir a predição em subárvores desconexas. Utilizando a Figura 8 como exemplo, uma amostra poderia ser associada com as classes **1.1** pelo classificador do nó **1** e **2.1.1** pelo classificador do nó **2.1**.

No contexto de utilização de classificação hierárquica para textos, alguns trabalhos são apresentados a seguir.

Inicialmente, Koller e Sahami (1997) apresentaram uma abordagem que se baseia na seleção de atributos mais relevantes para os classificadores que compõem a solução. Em seguida, Chakrabarti et al. (1998) realizaram experimentos sobre bases de notícias, patentes e documentos da *internet* que demonstram que seleção de atributos e, também, informações relacionadas ao contexto são proveitosas em problemas de classificação hierárquica. Ao mesmo tempo, McCallum, Nigam et al. (1998) apresentaram uma estratégia para aumentar o desempenho dos classificadores através da organização dos documentos de forma hierárquica, em detrimento da abordagem *flat*, sendo as técnicas apresentadas particularmente úteis em cenários com muitas categorias.

Weigend, Wiener e Pedersen (1999) apresentaram uma abordagem utilizando a estratégia *"dividir e conquistar"* para fazer uso da organização hierárquica sobre a base Reuters-22173[7], a fim de aumentar o desempenho dos classificadores. Em sequência, Dumais e Chen (2000), utilizando SVM, também constataram que utilizar uma organização hierárquica contribui para o aumento do desempenho do classificadores. Ruiz e Srinivasan (2002) também aplicaram a estratégia *dividir e conquistar* para categorização de registros médicos, porém utilizando redes neurais. Tikk e Biró (2003) desenvolveram um método de aprendizagem incremental, baseado em tentativa e erro, relacionado ao domínio de patentes. Em seguida, Kriegel et al. (2004) desenvolveram um método capaz de combinar múltiplas representações para as mesmas amostras, e lidar com representações faltantes, para classificação hierárquica relacionada ao domínio biológico.

Tikk, Biró e Törcsvári (2007) voltaram a endereçar o problema de patentes, quando propuseram um novo método de seleção das categorias e atualização de pesos a fim de lidar com o grande número de categorias do problema. Em seguida, Gauch, Chandramouli e Ranganathan (2009), assim como em trabalhos anteriores, concluíram que organizar o problema de forma hierárquica, em vez de utilizar uma representação *flat* aumenta o desempenho do classificador.

Em sequência, Ghazi, Inkpen e Szpakowicz (2010) utilizaram uma abordagem hierárquica para categorização de emoções em textos, constatando o mesmo comporta-

[7] A base Reuters-22173 é uma versão anterior da Reuters-21578.

mento de melhora de desempenho da abordagem hierárquica sobre a abordagem *flat*. Notaram ainda, que o desbalanceamento das amostras é menos evidente ao se utilizar uma organização hierárquica, apesar de ainda afetar o desempenho do classificador.

Mais recentemente, Zhou, Xiao e Wu (2011) apresentaram uma variação do SVM, também sobre bases de notícias, a fim de lidar com o problema de diferenciação das classes nos níveis mais profundos dada a similaridade entre elas.

3.3 Classificadores locais por nível (LCL)

A abordagem LCL consiste em utilizar um classificador *multiclasse* para cada nível de hierarquia. A Figura 9 ilustra a configuração dos classificadores em relação à taxonomia utilizando essa abordagem. Uma desvantagem é que classificadores de nível mais profundo compartilham classes com diversos ancestrais distintos, sendo portanto necessário discriminar entre muitas classes.

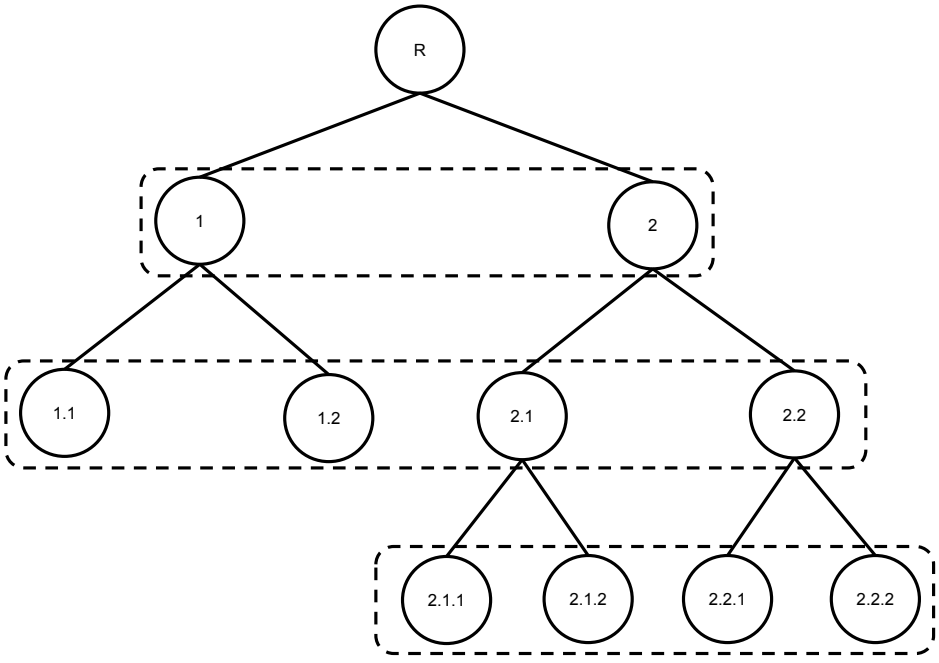


Figura 9 – Configuração dos classificadores *multiclasse* em relação à taxonomia utilizando a abordagem LCL. Os círculos representam as classes e os quadrados tracejados representam os classificadores. Adaptado de Silla Jr. e Freitas (2011).

Assim como descrito nas abordagens LCN e LCPN, algumas inconsistências preditivas podem ocorrer ao se utilizar essa abordagem, visto que não existe nenhum bloqueio que impeça uma predição entre classes provenientes de ancestrais distintos. Utilizando a Figura 9 como exemplo, o classificador de nível 1 (mais próximo a raiz R) pode ter como

saída a classe **1**, enquanto o classificador de nível 2 pode ter como saída a classe **2.1**. Em contrapartida, a abordagem apresenta um número reduzido de classificadores.

A abordagem LCL é a menos comum na literatura de classificação de textos, provavelmente devido à sua organização não intuitiva[8]. Um dos principais trabalhos que usa essa abordagem é apresentado por Clare e King (2003) no domínio de classificação genética.

3.4 Classificadores globais (GC)

A abordagem CG, conhecida também como *"Big Bang"*, consiste em classificadores que consideram a hierarquia de classes de uma única vez, produzindo um único classificador (SILLA JR.; FREITAS, 2011). A Figura 10 apresenta a configuração de um classificador global em relação a taxonomia.

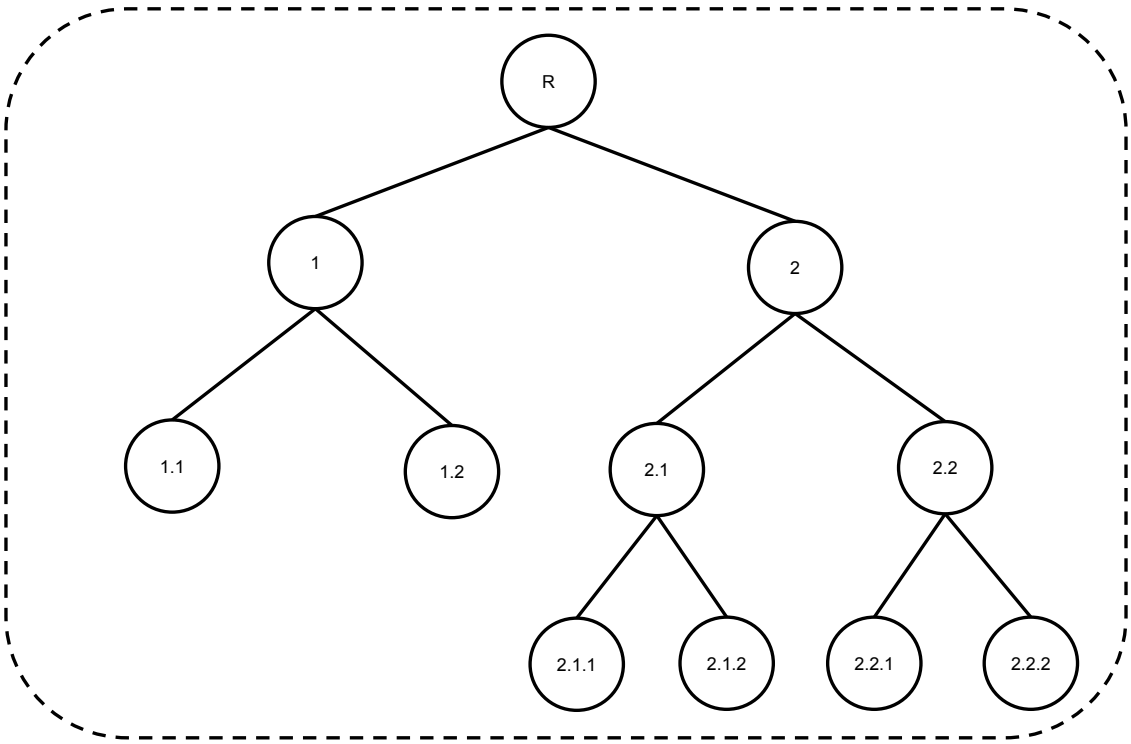


Figura 10 – Configuração do classificador global em relação a taxonomia. Os círculos representam as classes e os quadrados tracejados representam os classificadores. Adaptado de Silla Jr. e Freitas (2011).

A abordagem CG apresenta a vantagem de gerar um único classificador e naturalmente preserva a ancestralidade das classes. Todavia, o modelo produzido é mais

[8] Nessa abordagem um mesmo nível compartilha diversas categorias que não possuem o mesmo antecessor em comum.

complexo que qualquer umas das abordagens de classificadores locais e ainda renuncia a extensibilidade da taxonomia.

No contexto de utilização de classificação hierárquica de textos utilizando classificadores globais, os principais trabalhos são apresentados a seguir.

Labrou e Finin (1999) e Wang, Zhou e Liew (1999) apresentaram abordagens globais ao organizar uma coleção de documentos seguindo uma taxonomia que utiliza as categorias do *Yahoo!*.

O método proposto por Dekel, Keshet e Singer (2004) combinou ideias de análise *Bayesiana* e *large margin kernels* (VAPNIK, 1999), também relacionado a classificação de páginas da *internet*.

Cai e Hofmann (2004) apresentaram uma extensão do SVM, considerando informações do relacionamento entre as classes, porém, aplicado sobre o domínio de patentes. Por sua vez, Rousu et al. (2005) apresentaram uma variação do método *Maximum Margin Markov Networks*, aplicando uma estratégia de decomposição em subproblemas de uma única amostra e gradiente condicional para otimização dos subproblemas.

Kiritchenko, Matwin e Famili (2005) abordam o domínio de patentes e notícias, propondo ainda uma métrica de avaliação hierárquica baseada na ancestralidade de classes. Em seguida, Peng e Choi (2005) propuseram uma classificação baseada no significado das palavras e nas relações entre os grupos de significados, indicando que o classificador pode apresentar uma melhora de desempenho ao utilizar essa estratégia.

Mais recentemente, Qiu, Gao e Huang (2009) apresentaram um método para maximização global de margens, a fim de maximizar as margens não somente nas categorias folha, mas também nas categorias ancestrais.

O trabalho apresentado por Cerri e Carvalho (2010), no domínio de classificação genética *multilabel*, apresentou uma interessante comparação entre abordagens globais e locais: os resultados de abordagens locais, na sua maioria, foram superiores aos de abordagens globais, mesmo existindo o problema de propagação do erro na abordagem local.

Por fim, citam-se alguns trabalhos mais recentes na literatura que focam em problemas de classificação hierárquica de larga escala[9] como os trabalhos apresentados por Gopal et al. (2012), Ju, Moschitti e Johansson (2013), Oh e Jung (2017) e Peng et al. (2018).

[9] Por larga escala, pode-se entender um grande número de classes, múltiplos níveis hierárquicos ou uma grande quantidade de documentos.

3.5 Estratégias de avaliação de desempenho

Conforme postulado por Costa et al. (2007), as métricas de avaliação de classificadores hierárquicos podem ser categorizadas nos quatro tipos apresentados a seguir.

Baseado em distância Considera a distância entre a classe verdadeira e a classe predita para calcular o desempenho do classificador hierárquico, sendo uma extensão das métricas de classificadores *flat*. Esse tipo de métrica desconsidera a tendência de que a predição nos níveis mais profundos é mais difícil que nos níveis menos profundos.

Dependente de profundidade Tenta evitar as desvantagens dos métodos baseados em distância, atribuindo pesos maiores aos erros cometidos nos níveis menos profundos do que aos erros cometidos em níveis mais profundos. Isso é feito através de uma função que considera os pesos das arestas que conectam a classe verdadeira e a classe predita, assim como a profundidade que ambas se encontram. Dessa forma, apresentam a desvantagem de ser necessário determinar os pesos de cada aresta.

Baseado em semântica Utiliza uma noção intuitiva para calcular o desempenho do modelo preditivo baseado na similaridade entre as classes. Fundamenta-se na ideia que é menos danoso associar uma nova amostra a uma classe próxima da classe verdadeira do que associar a uma classe totalmente desconexa da classe verdadeira.

Baseado em hierarquia Essa estratégia considera as relações de descendência e ancestralidade para avaliar o desempenho dos classificadores. Essa abordagem visa considerar que a predição não é completamente incorreta ao predizer um classe que está localizada acima ou abaixo da subárvore da classe verdadeira.

3.5.1 Medidas de avaliação

Do ponto de vista de descendência, ou seja, em uma abordagem *top-down*, é demonstrado por Ipeirotis, Gravano e Sahami (2001) nas Equações 3.1, 3.2 e 3.3, as respectivas versões hierárquicas das medidas de precisão, revocação e F-medida:

$$\text{Precisão hierárquica: } HP \Downarrow = \frac{\sum_i |Desc(C_{predicted_i}) \cap Desc(C_{true_i})|}{\sum_i |Desc(C_{predicted_i})|} \tag{3.1}$$

$$\text{Revocação hierárquica: } HR \Downarrow = \frac{\sum_i |Desc(C_{predicted_i}) \cap Desc(C_{true_i})|}{\sum_i |Desc(C_{true_i})|} \tag{3.2}$$

$$\text{F-medida Hierárquica: } HF \Downarrow = 2 * \frac{HP \Downarrow * HR \Downarrow}{HP \Downarrow + HR \Downarrow} \tag{3.3}$$

Do ponto de vista da ascendência, ou seja, em uma abordagem *bottom-up*, as Equações 3.4, 3.5 e 3.6 são descritas por Kiritchenko, Matwin e Famili (2005) representando as respectivas versões hierárquicas das mesmas medidas:

$$\text{Precisão hierárquica: } HP \Uparrow = \frac{\sum_i |Ances(C_{predicted_i}) \cap Ances(C_{true_i})|}{\sum_i |Ances(C_{predicted_i})|} \tag{3.4}$$

$$\text{Revocação hierárquica: } HR \Uparrow = \frac{\sum_i |Ances(C_{predicted_i}) \cap Ances(C_{true_i})|}{\sum_i |Ances(C_{true_i})|} \tag{3.5}$$

$$\text{F-medida Hierárquica: } HF \Uparrow = 2 * \frac{HP \Uparrow * HR \Uparrow}{HP \Uparrow + HR \Uparrow} \tag{3.6}$$

Nas equações apresentadas acima, $Desc(C)$ representa o conjunto formado pela classe C e suas subclasses (ou seja todas as classes descendentes), enquanto que $Ances(C)$ representa o conjunto formado pela classe C e suas superclasses (ou seja, todas as classes antecedentes). Por fim, i é o índice da amostra atual do conjunto de teste sendo avaliada.

Nas propostas originais das F-medida hierárquicas (HF), um parâmetro β foi proposto para ponderar de forma distinta os valores da precisão hierárquica (HP) e da revocação hierárquica (HR). Nas versões aqui apresentadas (Equações 3.3 e 3.6), foi considerado $\beta = 1$, o que implica que HP e HR possuem pesos equivalentes, e assim, o parâmetro β pode ser suprimido, simplificando as Equações 3.3 e 3.6.

Considerando a Figura 11, é possível calcular as medidas $HP \Uparrow$, $HR \Uparrow$ e $HF \Uparrow$ propostas por Kiritchenko, Matwin e Famili (2005). Primeiramente, expande-se os conjuntos de classes ancestrais das classe correta G e da classe predita E, incluindo as próprias classes e descartando a classe R que é ancestral comum a todas classes, sendo portanto:

$$Ances(C_{true}) = Ances(G) = \{G, E, B\}$$

$$Ances(C_{predicted}) = Ances(E) = \{E, B\}$$

Em seguida, calcula-se a intersecção entre os ancestrais de G e E e a cardinalidade dos conjuntos de ancestrais.

$$|Ances(C_{predicted}) \cap Ances(C_{true})| = |Ances(G) \cap Ances(E)| = |\{E, B\}| = 2$$

$$|Ances(C_{true})| = |Ances(G)| = 3$$

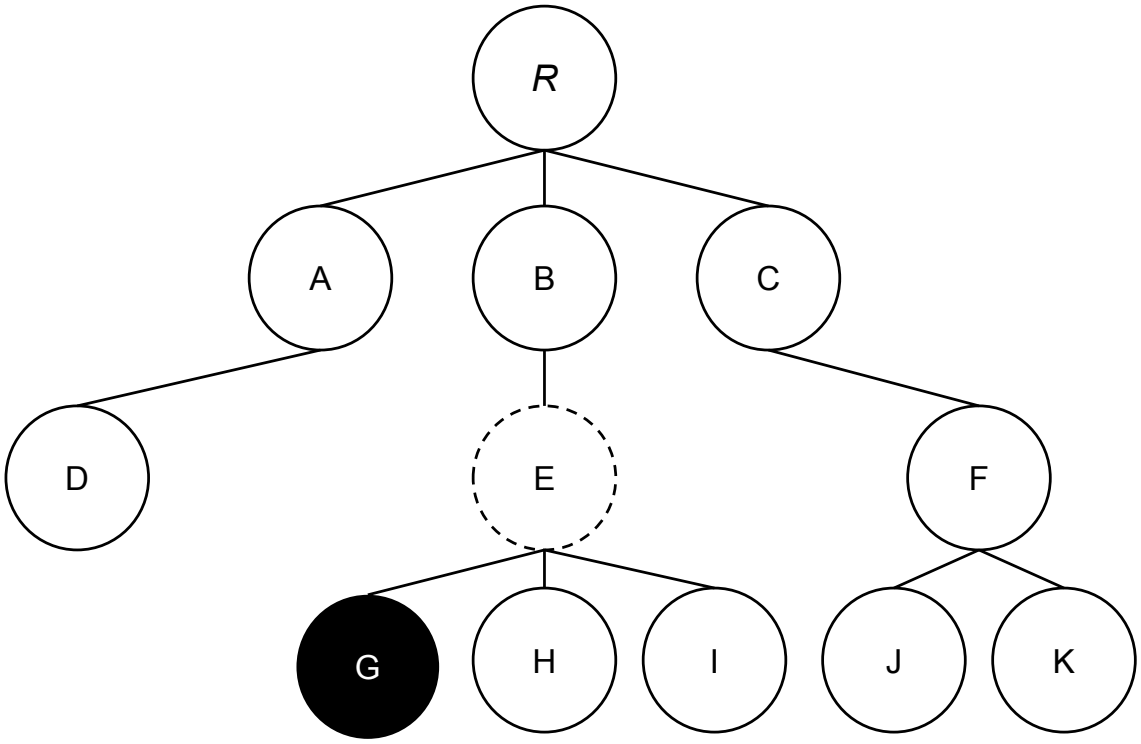


Figura 11 – Exemplo de hierarquia para cálculo das medidas de desempenho. Os círculos representam as classes que compõem a hierarquia, o círculo preenchido G representa a classe correta e o círculo tracejado E representa a classe predita. Adaptado de Kiritchenko, Matwin e Famili (2005).

$$|Ances(C_{predicted})| = |Ances(E)| = 2$$

Portanto, existe um conjunto de 2 rótulos atribuídos e 3 rótulos reais. Por fim, calculam-se, de fato, $HP\Uparrow$, $HR\Uparrow$ e $HF\Uparrow$:

$$HP\Uparrow = \frac{|Ances(C_{predicted}) \cap Ances(C_{true})|}{|Ances(C_{predicted})|} = \frac{|Ances(E) \cap Ances(G)|}{|Ances(E)|} = \frac{2}{2} = 1$$

$$HR\Uparrow = \frac{|Ances(C_{predicted}) \cap Ances(C_{true})|}{|Ances(C_{true})|} = \frac{|Ances(E) \cap Ances(G)|}{|Ances(G)|} = \frac{2}{3} = 0.666...$$

$$HF\Uparrow = 2 * \frac{HP\Uparrow * HR\Uparrow}{HP\Uparrow + HR\Uparrow} = 2 * \frac{1 * 2/3}{1 + 2/3} = 2 * \frac{2/3}{5/3} = 2 * \frac{2}{5} = 0.8$$

3.6 Adequações para o mecanismo proposto

O mecanismo de roteirização proposto consiste em uma variação da abordagem LCL. As modificações consistem em utilizar não somente um classificador por nível hierárquico, mas um classificador para cada conjunto de nós que partilham o mesmo nó pai. Uma segunda modificação consiste na criação de uma classe especial "**General**" para representar as amostras que não pertencem a nenhuma das outras classes. Essa configuração é ilustrada na Figura 12.

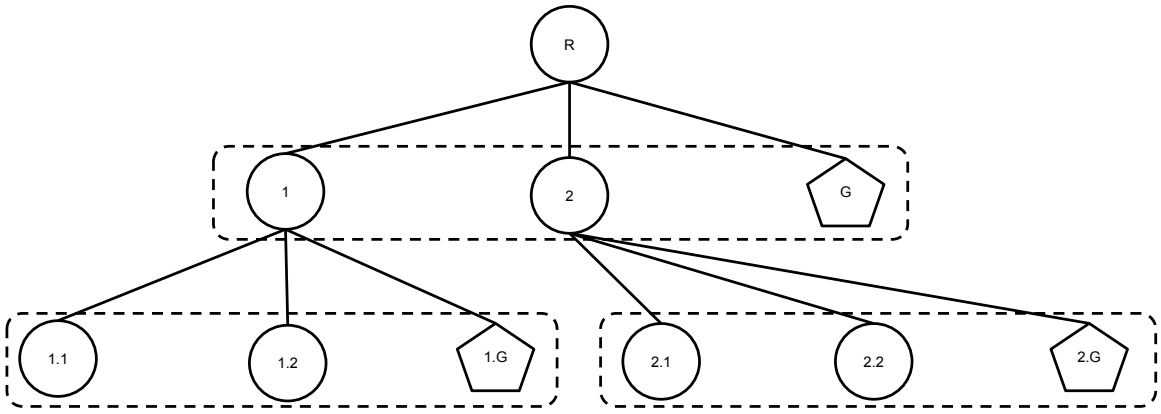


Figura 12 – Configuração dos classificadores no modelo proposto, no qual nós que partilham o mesmo nó-pai são agrupados em um classificador. Essa configuração dos classificadores apresenta ainda uma classe especial "**General**" para cada classificador. Os círculos representam as classes, os pentágonos representam a classe especial e os retângulos tracejados representam os classificadores.

Durante a fase de teste, a pergunta é classificada pelo nível mais raso. O resultado dessa predição é utilizado para selecionar o classificador do nível seguinte (se necessário). Caso a classe selecionada seja a classe especial "**General**" G do respectivo nível, a predição se encerra, caso contrário a predição segue até encontrar um nó-folha. Considerando o exposto, a abordagem adotada possui as características de ser *multiclasse*, pois seleciona uma entre as diversas classes disponíveis e MLNP, pois sempre obriga a seleção de um nó-folha.

Ao utilizar a abordagem ilustrada na Figura 12, alguns problemas dos classificadores locais, apresentados nas abordagens LCN, LCPN e LCL são endereçados. O primeiro deles são as inconsistências em tarefas *multiclasse*, visto que ao predizer uma classe nos níveis mais profundos, o resultado é consistente com a predição nos níveis superiores.

Essa consistência é possível, pois só existe uma subárvore abaixo de cada nó selecionado, tornando-se impossível associar classes desconexas à mesma amostra. Utilizando a Figura 12 como exemplo e supondo que o nó **1** foi selecionado, a predição segue para a subárvore que contém seus nós-filho (**1.1**, **1.2** ou **1.G**). A partir desse ponto, apenas nós

presentes na subárvore do nó **1** são permitidos, sendo que qualquer nó na subárvore do nó **2** não é passível de ser escolhido.

Com a criação da classe especial G, a definição de limiares (*threshold*) para indicar se a predição deve seguir para os níveis mais profundos ou encerrar no nível atual também foi solucionado. Novamente, tomando a Figura 12 como exemplo, a função do limiar é determinar se a predição encerra no nó **1** ou segue a subárvore com os nós-filho (**1.1**, **1.2** ou **1.G**). A existência da classe G, desprovida de subárvores, força o encerramento da predição nesse nó, não sendo necessária nenhuma estratégia para escolha de um limiar.

Este modelo apresenta ainda as vantagens de um número reduzido de classificadores em relação à abordagem LCN e, também, a modularidade dos classificadores que permite a extensão para novos domínios de forma mais simplificada que a abordagem CG.

Quanto à escolha da medida de avaliação para avaliar o mecanismo proposto, foram utilizadas as medidas baseadas na ancestralidade (abordagem *bottom-up*) propostas por Kiritchenko, Matwin e Famili (2005), principalmente devido ao crédito dado a predições parcialmente corretas. Por exemplo, supondo a Figura 11, o erro ao classificar uma amostra como E quando o correto seria G deve ser considerado menos grave que classificar a mesma amostra como sendo F, pois F está em uma subárvore distinta de E, e portanto, mais desconexa.

Considerando o contexto de sistemas de QA, por exemplo, se uma pergunta que pertence ao domínio *Football* for classificada como sendo de *Sports*, o erro pode ser considerado de pouca gravidade, visto que um RDQA menos especializado relacionado à *Sports* ainda deve ser capaz de produzir uma resposta sobre *Football*, mesmo que oferecendo menos detalhes do que o RDQA mais especializado. Portanto, dado o exemplo anterior, a "suavização do erro" torna a medida escolhida apropriada.

Uma outra característica interessante da medida selecionada é que erros nos níveis superiores são mais penalizados que em níveis inferiores (*e. g.,* supondo a Figura 11, é mais prejudicial classificar incorretamente uma amostra que pertence ao nó A como sendo do nó B do que uma amostra que pertence ao nó G como sendo do nó H). Por fim, a medida também apresenta a facilidade de ser computada e ser baseada somente na hierarquia, dispensando outros parâmetros adicionais (KIRITCHENKO; MATWIN; FAMILI, 2005).

Para a eficiência da abordagem de classificação apresentada, é necessário que exista uma quantidade representativa de perguntas rotuladas quanto aos domínios de interesse. O esforço para realizar as tarefas de coletar, limpar e, principalmente, rotular e atualizar um grande número de perguntas manualmente é proibitivo e inescalável face a complexidade das tarefas e a quantidade de assuntos abordados. Isso, por sua vez incentiva a elaboração de uma solução mais robusta e instantânea, como um gerador automático de perguntas rotuladas.

3.7 Considerações finais

Este capítulo apresentou o conceito de classificação hierárquica, detalhando como esse modelo de classificação estende o modelo *flat*. Também, foi exemplificado o conjunto de diretrizes a serem respeitadas pela taxonomia de classes para caracterizar o problema como hierárquico.

Foram apresentadas, ainda, as configurações dos métodos de classificação que atendem a essas diretrizes, destacando-se vantagens e desvantagens de cada abordagem. Para cada uma, foram apresentados os principais trabalhos existentes na literatura no cenário de classificação de textos.

Por fim, apresentaram-se as principais estratégias para avaliação do desempenho, destacando-se as medidas hierárquicas e a configuração escolhida para o classificador hierárquico proposto neste trabalho, que endereça alguns dos problemas das abordagens apresentadas anteriormente.

4 Geração automática de perguntas rotuladas

Conforme apresentado no Capítulo 1, as fontes de informação de sistemas de QA são principalmente grandes *corpora* de documentos e ontologias, dos quais potencialmente são extraídas as respostas para as perguntas. Por outro lado, o mecanismo roteirizador, conforme proposto no Capítulo 2, é um classificador hierárquico de domínio de *perguntas* dos usuários.

Devido às diferentes características entre as perguntas e documentos textuais, não é ideal utilizar diretamente os próprios documentos do *corpus* para treinar um classificador de domínio de perguntas. Por exemplo, considere a seguinte pergunta: *"Who is a pioneer in cognitive sciences?"*. O fragmento de texto[1] apresentado a seguir contém a resposta destacada em negrito: *"Avram Noam Chomsky (born December 7, 1928) is an American linguist, philosopher, cognitive scientist, historian, social critic, and political activist. Sometimes described as "the father of modern linguistics",* ***Chomsky is also a major figure in analytic philosophy and one of the founders of the field of cognitive science..."***. As principais diferenças entre as características da pergunta e do fragmento de texto são destacadas a seguir.

Dimensão A pergunta é composta por apenas 7 palavras, enquanto o fragmento da resposta possui algumas dezenas e o texto original completo é composto por mais de 10 mil. Portanto, ao se adotar uma representação vetorial, a pergunta geralmente é representada por um vetor muito mais esparso que o texto original e ambos os vocabulários são diferentes em qualidade e principalmente em quantidade de atributos;

Sinônimos A pergunta apresenta o termo *"pioneer"* para definir o conceito de precursor, enquanto no texto original o sinônimo *"founder"* é utilizado para representar o mesmo conceito, sendo essa escolha uma entre as possíveis. Outros sinônimos, como: *"beginner"*, *"initiator"*, *"creator"* ou *"originator"* poderiam ter sido utilizados para representar o mesmo conceito sem prejuízo semântico.

Entidades Conforme apresentando por Ittycheriah, Franz e Roukos (2001), as informações sobre as entidades podem ser obtidas exclusivamente das perguntas e indicam o tipo de entidade esperada como resposta. No exemplo, o pronome interrogativo *"Who"* indica que a entidade esperada na resposta é uma pessoa, enquanto o texto original

[1] *Noam Chomsky*. Disponível em: <http://goo.gl/teoGsA>, acessado em: 03/06/2017.

não possui explicitado que se trata de uma pessoa e, ainda, durante o decorrer do texto outras entidades do mesmo tipo e também de outros são mencionadas.

A Tabela 1 sintetiza as principais diferenças entre os documentos e perguntas.

Tabela 1 – Principais diferenças entre documentos e perguntas.

Característica	**Documentos**	**Perguntas**
Dimensão	$10^3 < x < 10^5$ termos	$2 < x < 30$ termos
Sinônimos	muitas ocorrências	poucas ocorrências
Entidades	muitas ocorrências	poucas ocorrências

Com base nessas diferenças, é esperado que um classificador treinado com amostras que representam perguntas venha a obter desempenho superior na classificação de domínio de perguntas do que um modelo treinado com os documentos originais. Sem contar que a elevada quantidade de atributos presentes em cada documento poderia ser proibitiva para o treinamento do classificador, devido ao alto custo computacional envolvido.

Idealmente, o classificador de domínio deve se beneficiar dos documentos que serão utilizados pelos sistemas de QA para a produção das respostas. Dessa forma, permite-se que o mecanismo de seleção desfrute de conteúdo compatível com as capacidades dos sistemas de QA produzirem respostas e, ainda possa ser facilmente expandido para agregar novos domínios à medida que sistemas de domínio restrito sejam incorporados.

Diante disso, neste trabalho, é proposto um modelo para automatizar o processo de geração de perguntas e oferecer treinamento contínuo a um classificador de domínio. Esse modelo, ilustrado na Figura 13, consiste na criação de amostras sintéticas a partir dos documentos que poderão ser utilizados para compor o próprio *corpus* empregado para aquisição de conhecimento dos RDQAs.

Esses documentos devem ser representativos e heterogêneos, de forma que a informação presente em seu conteúdo seja pertinente aos sistemas de QA no momento deles atenderem as solicitações dos usuários, almejando cobrir ao máximo o conteúdo delimitado pelos domínios dos RDQAs em questão.

Conforme explicado no Capítulo 3, cada RDQA que compõe o sistema possui um domínio associado e cada documento coletado é associado a um determinado RDQA. Logo, os documentos estão associados a determinados domínios. Consequentemente, as perguntas que serão geradas tendo como base esses documentos estarão associadas ao mesmo domínio do documento de origem.

O processo de geração de perguntas pode acompanhar a atualização do *corpus*, sendo que novas amostras podem ser geradas quando documentos novos são inseridos no repositório de dados. Deste modo, o classificador de domínio pode ser automaticamente atualizado de maneira incremental, de forma a manter e até melhorar o seu poder preditivo

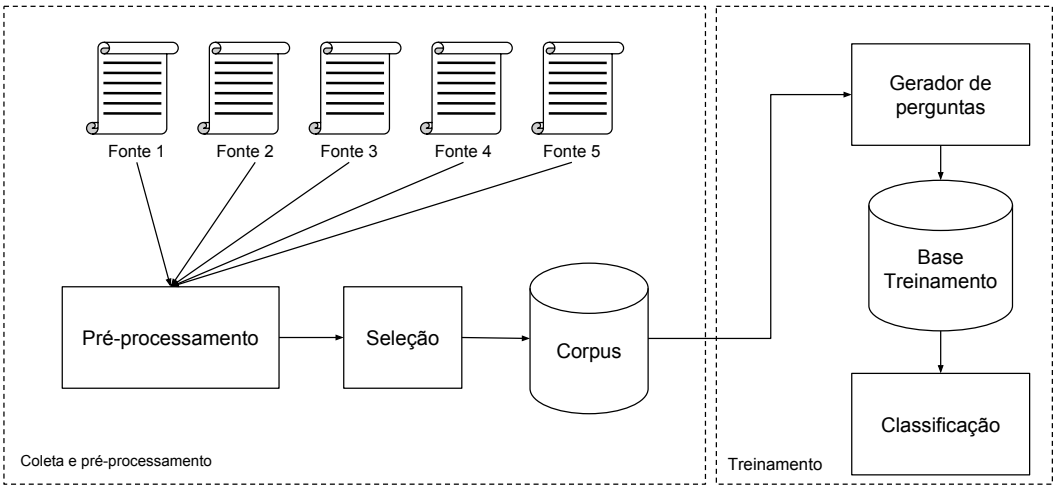


Figura 13 – Diagrama de treinamento do classificador: documentos são pré-processados e selecionados servindo para compor o *corpus* dos RDQAs e também para ser a entrada de um gerador de perguntas. O gerador de perguntas produz perguntas associadas ao mesmo domínio dos documentos de origem, sendo que esse domínio será a saída do classificador de domínio treinado com as perguntas produzidas na etapa anterior.

ao longo do tempo. Além disso, criar e manter manualmente uma base de dados rotulada de perguntas consumiria muito tempo e também poderia ser impeditivo para a geração desse classificador, justificando a elaboração de uma técnica automática.

Para a criação das perguntas de forma automática, uma versão modificada da ferramenta *Question Generation via Overgenerating Transformations and Ranking* (HEILMAN; SMITH, 2009) foi utilizada.

A ferramenta se baseia em processos de transformação seguindo um conjunto de regras bem definidas, que incluem: *"Wh-movement"*, *"subject-auxiliary inversion"* e *"verb decomposition"*. Essas transformações são aplicadas sobre os fragmentos de texto enviados para a ferramenta que, então, transforma sentenças declarativas em interrogativas (HEILMAN; SMITH, 2009).

A essência da técnica *Wh-movement* consiste em remover o foco da sentença declarativa e incluir um pronome interrogativo, em geral no início do sentença. Por exemplo, considerando a frase *"Grace Hopper invented the first compiler"*, após a transformação, o foco *"Grace Hopper"* é removido e o pronome *"Who"* é incluído no início da sentença. Ao final, a pergunta *"Who invented the first compiler?"* é gerada (HEILMAN; SMITH, 2009).

A técnica *subject-auxiliary inversion* consiste em realizar a inversão entre o sujeito da frase e um verbo auxiliar e, dessa forma, transformar a sentença declarativa em interrogativa. Supondo a frase *"Quentin Tarantino has been the most commented subject on the festival"*, ao inverter o sujeito *"Quentin Tarantino"* com o verbo auxiliar *"has"*, a sentença é transformada na pergunta *"Has Quentin Tarantino been the most commented*

subject on the festival?" (HEILMAN; SMITH, 2009).

Usando essas e outras técnicas, a ferramenta é capaz de produzir diversas perguntas, pois como afirmam Heilman e Smith (2009), múltiplas respostas existem dentro de uma sentença declarativa (*e.g.,* a sentença *"Francium was discovered by Marguerite Perey in France in 1939"*[2] pode gerar as seguintes perguntas: *"Where was Francium discovered by Marguerite Perey in 1939?"*, *"When was Francium discovered by Marguerite Perey in France?"*, *"Was francium discovered by Marguerite Perey in France in 1939?"*, *"By what was francium discovered in France in 1939?"*).

Segundo Heilman e Smith (2009), o processo generativo é baseado na premissa de que as perguntas podem ser visualizadas como transformações léxicas, sintáticas e semânticas de sentenças declarativas. Dessa forma, a ferramenta é dividida em três estágios, resumidos a seguir:

Transformações de NLP: As sentenças dos primeiros parágrafos do documento original são comprimidas e simplificadas, a fim de prevenir que os estágios seguintes gerem perguntas sem sentido e "artificiais".

Transdutor de perguntas: As sentenças declarativas produzidas pelo estágio anterior e outras sentenças declarativas que foram diretamente extraídas dos documentos são transformadas em perguntas sintéticas de diferentes tipos (*"Who"*, *"What"*, *"Where"*, *"When"* e *"How much"*).

Classificador de Perguntas: Um modelo baseado em regressão logística estima a probabilidade de aceitação de uma pergunta, penalizando perguntas que exibam deficiências[3], como a pergunta *"By what was francium discovered in France in 1939?"*.

Algumas configuração foram efetuadas para evitar perguntas de baixa qualidade, como aquelas em que o pronome não pode ser resolvido, (*e.g., "Who is she?"*), e perguntas com mais de 30 *tokens*, e para priorizar perguntas que iniciem por *"Wh"* (*"Who"*, *"What"*, *"Where"*, *"When"*). A Tabela 2 apresenta algumas perguntas geradas a partir do fragmento de uma notícia[4].

Além das configurações nativas da ferramenta apresentada acima, a mesma foi estendida para: (*i*) lidar com diferentes fontes de informação, (*ii*) atribuir os domínios e subdomínios normalizados dos documentos para as perguntas geradas a partir deles e (*iii*) descartar perguntas de baixa qualidade.

[2] *Francium*. Disponível em: <goo.gl/JfU75G>, acessado em 09/05/2018.

[3] As deficiências incluem perguntas vagas, sem sentido, com respostas óbvias, com o pronome interrogativo incorreto, entre outros casos.

[4] *Jay-Z and Marina Abramović rekindle art's relationship with pop*. Disponível em: <http://goo.gl/DdCicq>, acessado em 31/01/2018.

Tabela 2 – Perguntas geradas automaticamente a partir de um fragmento de texto.

Identificação	**Conteúdo**
Fragmento original	*The renowned performance artist Marina Abramović danced with Jay-Z at a New York art gallery this week as part of the rapper's latest video shoot. It is the latest in a series of crossovers between art and pop that at first glance seem unlikely. As if Ai Weiwei and Anish Kapoor performing Gangnam Style and Weiwei also recording a heavy metal album were not enough, here is Abramović, a legend of contemporary art, dancing to Jay-Z's Picasso Baby. Meanwhile Yoko Ono's Plastic Ono Band got rave reviews in London this summer. (...) Abramović is in her late 60s, Ai Weiwei well into his 50s. Yoko Ono, one half of art and pop's ultimate marriage, is 80.*
Pergunta 1	*Who is one half of art and pop's ultimate marriage?*
Pergunta 2	*Who is a legend of contemporary art?*
Pergunta 3	*What does Yoko Ono's Plastic Ono Band's not just another example of?*
Pergunta 4	*Who danced with Jay-Z at a New York art gallery this week as part of the rapper's latest video shoot?*
Pergunta 5	*Who is 80?*

Durante a apresentação dos documentos para o gerador, somente os **dois** parágrafos[5] iniciais dos documentos foram utilizados como entrada para a ferramenta de geração, sendo estes, segmentados em frases antes do envio. Os parágrafos próximos ao início do texto são mais informativos e tendem a produzir questões de maior qualidade (LIN; HOVY, 1997).

A ferramenta foi modificada para utilizar documentos previamente coletados de diversas fontes e armazenados em bancos de dados, neste ponto, já contemplando as informações de domínios e subdomínios normalizados. **Os domínios e subdomínios dos documentos são propagados como rótulos para as perguntas geradas**. Essa modificação, que aproveita os domínios dos documentos, é crucial para endereçar os problemas de rotulação manual e validar a hipótese proposta neste trabalho.

Um processo adicional de validação automática também foi incorporado, além do estágio nativo da ferramenta, para remover questões que não possuam informação suficiente para a classificação do domínio, como a Pergunta 5 apresentada na Tabela 2 (*"Who is 80?"*). Nesse processo, as *stopwords*[6] e caracteres não alfabéticos são temporariamente removidos e, se restarem menos de três *tokens*, a pergunta é descartada. Isso é realizado, pois essas perguntas são extremamente breves, e consequentemente, desprovidas de atributos para caracterização de seus domínios. Perguntas duplicadas também são descartadas.

Após a execução da etapa de geração de perguntas, ocorre a etapa de treinamento do classificador hierárquico que recebe como entrada amostras de perguntas sintéticas produzidas pela etapa anterior e produz como saída da predição, o domínio da pergunta.

O gerador de perguntas é capaz de criar automaticamente dezenas ou mesmo

[5] A tokenização de sentenças se dá através da função `sent_tokenize` da biblioteca NLTK. Disponível em: <goo.gl/QacrPg>, acessado em 09/05/2018.

[6] As *stopwords* removidas pertencem ao repositório da biblioteca NLTK. Disponível em: <http://www.nltk.org/book/ch02.html>, acessado em 31/01/2018.

centenas de amostras rotuladas a partir de um documento de texto. Assim, é possível obter facilmente uma base de treinamento de larga escala, composta por milhões de amostras rotuladas. Por fim, após o treinamento do classificador de domínio utilizando o conjunto de perguntas geradas, o mecanismo de roteirização estará completo e pronto para a fase de teste, como apresentado na Figura 13.

4.1 Considerações finais

Este capítulo apresentou as diferenças entre as características dos documentos e perguntas, indicando os desafios de se produzir o modelo proposto. Outra parte essencial da solução é apresentada: **um gerador automático de perguntas rotuladas que servirão como conjunto de treinamento**.

Esse módulo de geração de perguntas é responsável por produzir perguntas sintéticas que serão utilizadas para treinar o classificador hierárquico de domínio de perguntas empregado para selecionar o sistema de QA, conforme apresentado no Capítulo 2.

O gerador de perguntas sintéticas foi apresentado, expondo as técnicas utilizadas para a produção das perguntas e assegurar a qualidade das mesmas. O capítulo descreveu ainda, como o gerador se integra nas etapas de treinamento e operação do sistema unificado.

A seguir, é explicada a metodologia empregada na realização dos experimentos, detalhando as etapas e tarefas realizadas para desenvolver e avaliar o classificador de domínio hierárquico de perguntas.

5 Experimentos

A seguir, são detalhados os passos adotados para realização dos experimentos. São descritas as fases e tarefas planejadas e realizadas na validação da hipótese desta pesquisa e, também são fornecidas informações que subsidiam a preparação dos dados, experimentos e a avaliação dos resultados.

A metodologia experimental empregada está sumarizada na Figura 14, sendo que cada uma das etapas está detalhada nas seções subsequentes.

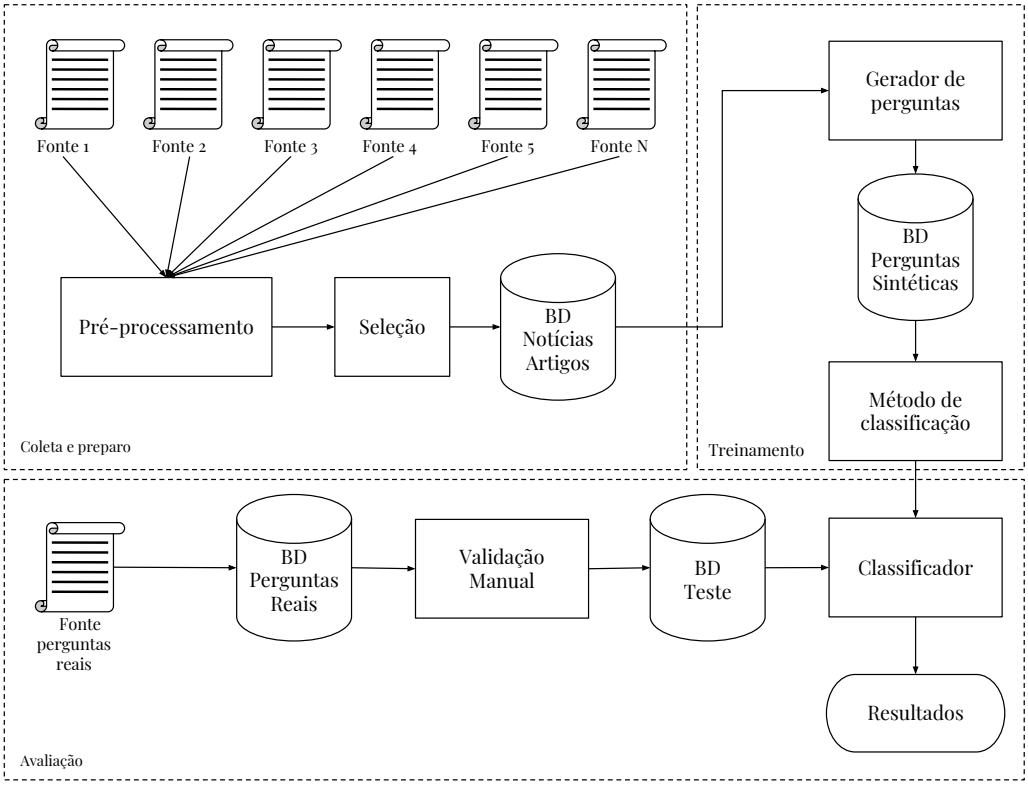


Figura 14 – Diagrama da metodologia experimental empregada. Primeiramente, foram coletados, pré-processados e selecionados os documentos que compõem o *corpus*. Em seguida, cada um desses documentos foi usado como entrada para a geração de perguntas sintéticas que, por sua vez, foram usadas para treinar o classificador de domínio. Para avaliar o desempenho desse classificador, foi utilizado um conjunto de teste composto por perguntas reais.

5.1 Coleta e preparo

A fase inicial dos experimentos consiste em preparar um *corpus* de documentos relacionados a um conjunto de categorias para que os mesmos possam servir como fonte de dados para produção das respostas e para a geração de perguntas sintéticas.

Inicialmente, foi coletado um *corpus* de documentos textuais composto por (*i*) artigos da Wikipedia anglófona[1] e (*ii*) notícias publicadas em língua inglesa entre 02/07/2004 e 02/10/2016 pelos seguintes portais e jornais *online*: BBC[2], Fox News[3], NBC News[4], Wired[5], The Guardian[6] e The Wall Street Journal[7].

O *corpus* é heterogêneo em sua composição e contém documentos de tamanhos variados e diferentes estilos de escrita. Essa abordagem visa adicionar variabilidade na composição das fontes, remetendo ao comportamento de um sistema de QA que busca fragmentos de texto para gerar as respostas em múltiplas fontes. O repositório de documentos coletados totalizou **6.660.814** documentos, dos quais 1.432.453 são notícias e 5.228.361 são artigos da *Wikipedia*.

Através de um *crawler*, as notícias foram capturadas e apenas o conteúdo textual das notícias, o título e a categoria foram extraídos, descartando-se a formatação do texto, assim como imagens, *links*, anúncios, propagandas, comentários, referências e outros elementos não textuais. A mesma estratégia de limpeza foi aplicada sobre os artigos da *Wikipedia*. Esse pré-processamento se fez necessário, pois o corpo dos documentos é a principal parte utilizada pelos sistemas de QA para extração de informação que irá produzir respostas. A utilização de outras partes do documento exigiria técnicas mais sofisticadas de pré-processamento para extração de algum conteúdo que pudesse ser aproveitado.

Em seguida, as categorias dos documentos foram normalizadas utilizando a taxonomia definida pelo IPTC (*International Press Telecommunications Council*)[8], sendo selecionados apenas os documentos pertencentes aos domínios e subdomínios ilustrados na Figura 15, resultando em 5 domínios e 21 subdomínios.

Conforme descrito no Capítulo 3, um subdomínio especial *General* foi incluído para representar todos os documentos que não se enquadrem nos outros subdomínios disponíveis.

Originalmente, cada documento é associado a um domínio e a um subdomínio, sendo esse rótulo atribuído pela fonte de informação, no caso os portais de notícias e a *Wikipedia*. Entretanto, cada fonte pode rotular o mesmo domínio de maneiras distintas, por exemplo, *"Sport"*, *"Sports"*, *"All Sports"*, sendo necessária a normalização das categorias seguindo o IPTC. Após a normalização, o *corpus* de documentos pertencentes aos domínios de interesse totalizou **531.641** documentos. A Tabela 3 apresenta a distribuição dos

[1] *Wikipedia Dumps*. Disponível em <http://goo.gl/peMnDM>, coletado em 20/11/2016.
[2] *BBC*. Disponível em <http://www.bbc.com>, acessado em 21/06/2017.
[3] *Fox News*. Disponível em <http://www.foxnews.com>, acessado em 21/06/2017.
[4] *NBC News*. Disponível em <http://www.nbcnews.com>, acessado em 21/06/2017.
[5] *Wired*. Disponível em <http://www.wired.com>, acessado em 21/06/2017.
[6] *The Guardian*. Disponível em <http://goo.gl/w7jcrP>, acessado em 21/06/2017.
[7] *The Wall Street Journal*. Disponível em <http://www.wsj.com>, acessado em 21/06/2017.
[8] *Select and Show IPTC NewsCode Taxonomies as Tree Diagram*. Disponível em: <http://show.newscodes.org/index.html?newscodes=subj>, acessado em 03/06/2017.

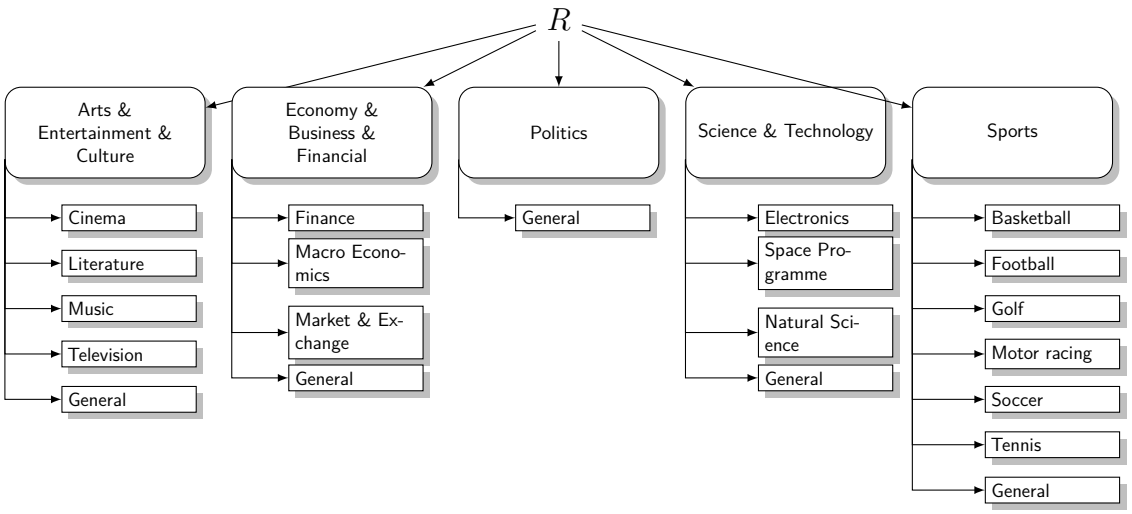


Figura 15 – Taxonomia selecionada para os experimentos. Retângulos com canto arredondado representam o primeiro nível hierárquico (domínio) e retângulos com canto angulado representam o segundo nível hierárquico (subdomínio). R representa a raiz da árvore.

documentos em cada categoria.

Por fim, apenas os documentos restritos às categorias de interesse, já com a identificação unificada para essas categorias, foram armazenados em um repositório (BD) para serem utilizados na fase seguinte.

5.2 Treinamento

A fase intermediária dos experimentos consiste em gerar perguntas sintéticas com base nos documentos preparados pela fase de coleta e preparo, e utilizar essas perguntas para treinamento de um classificador hierárquico de domínio.

A fase é iniciada pela geração das perguntas, sendo fundamental que essa tarefa gere um número de perguntas elevado, a fim de garantir a representatividade das perguntas no treinamento. Para isso, a versão modificada do gerador de perguntas, apresentada no Capítulo 4, foi aplicada sobre os documentos que compõem o *corpus*.

É importante salientar que: (*i*) os domínios e subdomínios foram propagados dos documentos para as perguntas geradas, portanto, não é necessário atribuir manualmente rótulos para cada pergunta; e (*ii*) o gerador é capaz de gerar dezenas (ou mesmo centenas) de perguntas para cada documento e, por isso, é possível criar, facilmente, grandes bases de dados com milhões de perguntas.

Após a execução do gerador, cada um dos documentos selecionados foi utilizado como entrada e, por fim, foram criadas automaticamente **3.517.858** perguntas sintéticas

Tabela 3 – Quantidade de documentos por domínio e subdomínio.

Domínio **Subdomínio**	$\|D\|$
Arts & Entertainment & Culture	**231.652**
Cinema	62.632
Literature	32.695
Music	61.889
Television	42.951
General	31.485
Economy & Business & Financial	**43.037**
Finance	19.915
Macroeconomics	6.874
Market & Exchange	9.315
General	6.933
Politics	**95.561**
General	95.561
Science & Technology	**51.699**
Electronics	3.442
Space Programme	4.535
Natural Sciences	20.274
General	23.448
Sports	**141.177**
Basketball	13.122
Football	65.872
Golf	3.296
Motor racing	2.742
Soccer	6.564
Tennis	5.516
General	44.065
Total	**531.641**

rotuladas (Tr), sendo o vocabulário composto por **567.528** *tokens*[9].

Utilizando os artifícios explicados no Capítulo 4, somente as perguntas geradas de boa qualidade e proveitosas para a tarefa de classificação, foram armazenadas em um repositório (BD), incluindo os rótulos dos documentos originais.

A Tabela 4 sumariza as estatísticas básicas do conjunto de treinamento Tr, sendo que a coluna $|D|$ representa a quantidade de perguntas. A coluna % apresenta a proporção

[9] A separação em *tokens* foi realizada usando a função `word_tokenize` da biblioteca NLTK. Disponível em <goo.gl/QacrPg>, acessado em 09/05/18.

de perguntas no conjunto. As colunas $\mathcal{M}$ e IQR são, respectivamente, a mediana e o intervalo interquartil do número de *tokens* por pergunta.

Um detalhe importante é que a mediana e o intervalo interquartil (IQR) dos *tokens* por pergunta é muito baixo, apesar do número elevado de *tokens* que compõem o vocabulário, o que indica que cada amostra é representada por um vetor altamente esparso.

Tabela 4 – Estatísticas básicas sobre o conjunto de treinamento (Tr).

Domínio	**Subdomínio**	$\|D\|$	%	$\mathcal{M}$	**IQR**
Arts & Entertainment & Culture		**1.305.860**	**37%**	**9**	**7**
	Cinema	368.752		9	7
	Literature	223.906		9	7
	Music	330.249		10	6
	Television	241.446		9	7
	General	141.507		9	7
Economy & Business & Financial		**265.196**	**8%**	**10**	**8**
	Finance	69.973		10	7
	Macro Economics	41.853		11	9
	Market & Exchange	25.729		10	7
	General	127.641		10	7
Politics		**262.284**	**7%**	**10**	**7**
	General	262.284		10	7
Science & Technology		**466.199**	**13%**	**10**	**7**
	Electronics	27.017		10	7
	Space Programme	103.284		9	6
	Natural Science	90.989		10	6
	General	244.909		10	7
Sports		**1.218.319**	**35%**	**10**	**8**
	Basketball	59.240		11	6
	Football	586.351		11	8
	Golf	27.144		10	8
	Motor racing	59.258		10	7
	Soccer	21.576		10	7
	Tennis	40.109		10	7
	General	424.641		10	7
Total		**3.517.858**	**100%**	–	–

Para apresentar mais detalhes sobre a composição da base de treinamento, a "relevância" de cada *token* foi calculada em relação a seu *information gain* (IG) (YANG; PEDERSEN, 1997) sobre o *corpus* completo e sobre cada um dos domínios envolvidos (exceto para o domínio *Politics*, visto que este possui apenas um subdomínio). A Tabela 5

apresenta os dez *tokens* que apresentaram o maior IG, podendo-se notar que os mesmos são pertinentes ao contexto dos domínios que representam. As frequências relativas dos *tokens* também foram calculadas e estão ilustradas pelas nuvens de palavras apresentadas na Figura 16.

Tabela 5 – Os dez *tokens* como maiores IG sobre o *corpus* completo e sobre cada um dos domínios.

***Corpus* completo**		**Arts & Entertainment & Culture**		**Economy & Business & Financial**	
Token	**IG**	**Token**	**IG**	**Token**	**IG**
team	0,013	film	0,050	bank	0,070
film	0,013	song	0,040	company	0,041
season	0,012	album	0,023	market	0,021
league	0,012	music	0,022	economic	0,017
games	0,011	book	0,018	banks	0,016
football	0,011	single	0,016	economy	0,014
airport	0,010	released	0,013	banking	0,010
play	0,010	novel	0,013	growth	0,008
species	0,009	game	0,012	economics	0,006
song	0,009	published	0,011	business	0,006

	Science & Technology		**Sports**	
	Token	**IG**	**Token**	**IG**
	airport	0,084	basketball	0,042
	aircraft	0,026	football	0.026
	history	0,014	season	0,017
	air	0,014	tournament	0,016
	airlines	0,012	men	0,016
	runway	0,011	held	0,013
	aviation	0,010	tennis	0,013
	airline	0,010	league	0,012
	operations	0,009	golf	0,012
	apple	0,008	soccer	0,012

No treinamento do classificador hierárquico de domínio, as perguntas sintéticas de Tr foram primeiramente convertidas para letras minúsculas e segmentadas em múltiplos *tokens* utilizando como delimitadores quaisquer caracteres não alfanuméricos. A representação utilizada foi o tradicional modelo espaço-vetorial (*bag of words*) e os valores usados para representar os termos foram a frequência de ocorrência em cada pergunta.

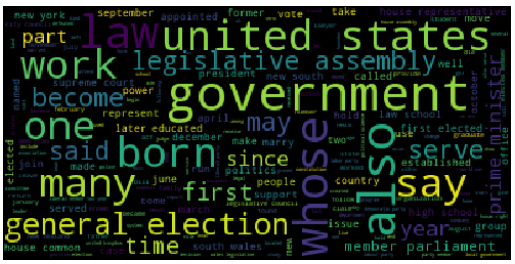
Devido ao volume elevado de amostras e à necessidade de aprendizado incremental e de velocidade na classificação, foi empregado o método de classificação *naïve Bayes* multinomial, que é amplamente utilizado para categorização de textos (KOLLER; SAHAMI, 1997; MCCALLUM; NIGAM et al., 1998; SILLA JR.; FREITAS, 2009; NGUYEN; COOPER; KAMEI, 2011), sendo portanto, um *baseline* natural para os experimentos. Ele foi implementado usando a linguagem de programação `Python`, através da função disponibilizada pela biblioteca `scikit-learn`[10].

Outros métodos de classificação tradicionais, como SVM e Redes Neurais Artificiais,

[10] *Scikit learn*. Disponível em <http://scikit-learn.org/stable/>, acessado em 23/06/2017.

(a) Arts & Entertainment & Culture

(b) Economy & Business & Financial



(c) Politics

(d) Science & Technology

(e) Sports

Figura 16 – Nuvem de palavras representando a frequência relativa das palavras para cada um dos domínios abordados.

não foram empregados devido ao elevado custo computacional e dificuldade em realizar treinamento incremental. Além disso, o propósito deste trabalho não é encontrar o melhor método de classificação existente para resolver o problema de classificação hierárquica, mas sim, certificar que um classificador treinado com as perguntas sintéticas pode ser empregado, com segurança, para classificar perguntas reais.

5.3 Avaliação

A hipótese de pesquisa a ser validada é se um classificador treinado com a base de perguntas sintéticas é capaz de inferir corretamente os domínios e subdomínios de perguntas reais, viabilizando a utilização do mesmo para a seleção de sistemas de QA.

Para avaliar se as perguntas sintéticas não possuem características distintas das

perguntas reais feitas por humanos, foi criado um conjunto de teste composto somente por perguntas reais, extraídas do portal *Quora*[11] através de um *crawler*. As perguntas coletadas foram distribuídas nos mesmos domínios e subdomínios de documentos do *corpus*, detalhados na Figura 15.

Como as perguntas do *Quora* são categorizadas automaticamente por um método de classificação proprietário, elas passaram por um processo manual de curadoria para assegurar que foram corretamente rotuladas. Após esse processo, restaram 2.100 perguntas reais na base de teste (Te).

A Tabela 6 apresenta as estatísticas básicas sobre o conjunto de teste Te, na qual a coluna $|D|$ exibe a quantidade de perguntas, % indica a distribuição das perguntas em cada domínio, $\mathcal{M}$ e IQR são a mediana e o intervalo interquartil do número de *tokens* por pergunta, respectivamente. Tanto as amostras do conjunto de treinamento (Tr) como de teste (Te) apresentam características semelhantes.

Tanto em Te como em Tr, as amostras são representadas por vetores altamente esparsos. Analisando as estatísticas $\mathcal{M}$ e IQR de ambos, pode-se constatar que as amostras que compõem Te possuem um número de termos ligeiramente maior que as que compõem Tr. Isso indica que as perguntas reais (Te) podem ser um pouco mais detalhadas que as perguntas geradas sinteticamente.

Analisando aleatoriamente as perguntas de Te, pode-se notar um comportamento dos usuários do portal *Quora*. Eles tendem a elaborar perguntas mais longas, o que nem sempre é acompanhado pelo gerador de perguntas. A dimensão das perguntas sintéticas de Tr depende diretamente da dimensão do fragmento do documento que as originou. Ainda assim, o fato das amostras de teste Te apresentarem ligeiramente mais termos que as amostras de Tr não deve ser prejudicial à capacidade preditiva do classificador, dada a característica probabilística do método de classificação empregado.

O cenário de aplicação considerado neste trabalho é composto por domínio e subdomínio, configurando um problema de classificação hierárquica. A abordagem utilizada pelos experimentos segue a estratégia apresentada no Capítulo 3, que pode ser resumida da seguinte forma: um classificador é treinado com todas as amostras de treinamento a fim de viabilizar a classificação do domínio da pergunta; em seguida, um classificador especializado é gerado para cada um dos subdomínios. Dessa forma se, por exemplo, o classificador geral indicar o domínio de uma dada pergunta como *Sports*, o classificador especializado em *Sports* é utilizado para classificar a pergunta em um dos seguintes subdomínios: *Basketball*, *Football*, *Golf*, *Motor racing*, *Soccer*, *Tennis* ou *General*.

A vantagem dessa abordagem *top-down* é que o problema de classificação hierárquica é dividido em vários problema menores e mais simples de classificação multiclasse. Além

[11] *Quora*. Disponível em <http://www.quora.com>, acessado em 21/06/2017.

Tabela 6 – Estatistícas sobre o conjunto de teste (Te).

Domínio / Subdomínio	$\|D\|$	%	$\mathcal{M}$	IQR
Arts & Entertainment & Culture	**500**	**24%**	**11**	**6**
Cinema	100		11	6
Literature	100		12,5	6
Music	100		13	6,2
Television	100		11	5
General	100		11	7
Economy & Business & Financial	**400**	**19%**	**12**	**7**
Finance	100		11	6
Macro Economics	100		11	7
Market & Exchange	100		11	6
General	100		12	10
Politics	**100**	**5%**	**12**	**7**
General	100		12	7
Science & Technology	**400**	**19%**	**13**	**7**
Electronics	100		12	6
Space Programme	100		13	6,2
Natural Science	100		13	8
General	100		13	7
Sports	**700**	**33%**	**12**	**7**
Basketball	100		12	7
Football	100		11,5	5,2
Golf	100		10	6
Motor racing	100		12,5	6,2
Soccer	100		12,5	7
Tennis	100		12	6
General	100		12	9
Total	**2.100**	**100%**	–	–

disso, a predição é consistente, pois segue sempre uma única subárvore. No entanto, uma desvantagem bem conhecida dessa abordagem é que um erro de classificação em uma das categorias poderá se propagar pelas subcategorias. Com o intuito de facilitar a reprodução dos experimentos, as bases de dados Tr e Te estão disponíveis publicamente para consulta[12]. Ainda, seguindo a metodologia apresentada na Figura 14, foram considerados três cenários experimentais distintos.

- **Cenário 1**: um classificador NB foi treinado utilizando a base de dados original

[12] *Labeled Documents and Questions Dataset*. Disponível em <bit.ly/qdcDataSet>, acessado em 25/06/2017.

(Tr);

- **Cenário 2**: um classificador NB foi treinado utilizando Tr após remoção de *stopwords*;
- **Cenário 3**: um classificador NB foi treinado utilizando Tr após a remoção de *stopwords* e balanceamento de classes por subamostragem aleatória (HE; GARCIA, 2009).

O classificador treinado em cada cenário foi utilizado para predizer os rótulos de perguntas reais presentes no conjunto de testes (Te). Logo, o desempenho obtido é uma medida da qualidade do conjunto de treinamento criado sinteticamente, o qual foi utilizado para treinamento do classificador.

Em todos os cenários, as perguntas de teste foram convertidas para letras minúsculas e segmentadas em múltiplos *tokens* utilizando como delimitadores quaisquer caracteres não alfanuméricos. Foi empregada a representação espaço-vetorial *(bag of words)* com a frequência de ocorrência dos termos na pergunta, da mesma forma que realizado para o treinamento do classificador hierárquico.

Seguindo as medidas hierárquicas apresentadas no Capítulo 3, a Tabela 7 apresenta o desempenho considerando a organização hierárquica das categorias, destacando em negrito os melhores desempenhos. A Tabela 8 apresenta os resultados obtidos no primeiro nível da hierarquia e a Tabela 9 apresenta os resultados obtidos no segundo nível, destacando em negrito as médias da F-medida e da acurácia.

Tabela 7 – Desempenho dos classificadores considerando a estrutura hierárquica das categorias.

	Cenário 1	**Cenário 2**	**Cenário 3**
HP⇑	0,799	0,805	**0,817**
HR⇑	0,708	0,733	**0,814**
HF⇑	0,751	0,767	**0,815**

Tabela 8 – Desempenho obtido no primeiro nível da hierarquia.

	Cenário 1	**Cenário 2**	**Cenário 3**
Domínio	**F-medida**		
Arts & Entertainment & Culture	0,854	0,853	0,896
Economy & Business & Financial	0,836	0,865	0,900
Politics	0,573	0,667	0,773
Science & Technology	0,836	0,870	0,860
Sports	0,926	0,930	0,924
Média	**0,805**	**0,840**	**0,871**
Acurácia	**0,861**	**0,877**	**0,893**

Os resultados apresentados na Tabela 7 indicam que o classificador obteve bom desempenho em todos os cenários (HF⇑ maior que $0,75$ em todos os experimentos). De

Tabela 9 – Resultados obtidos no segundo nível da hierarquia.

	Cenário 1	**Cenário 2**	**Cenário 3**
Domínio **Subdomínio**	**F-medida**		
Arts & Entertainment & Culture			
Cinema	0,453	0,465	0,520
Literature	0,331	0,406	0,784
Music	0,709	0,679	0,800
Television	0,526	0,541	0,559
General	0,185	0,222	0,539
Economy & Business & Financial			
Finance	0,229	0,208	0,297
Macro Economics	0,523	0,533	0,623
Market & Exchange	0,667	0,758	0,920
General	0,741	0,770	0,852
Politics			
General	0,728	0,774	0,637
Science & Technology			
Electronics	0,226	0,237	0,610
Space Programme	0,802	0,788	0,690
Natural Science	0,814	0,803	0,829
General	0,281	0,362	0,459
Sports			
Basketball	0,573	0,667	0,773
Football	0,363	0,412	0,431
Golf	0,039	0,095	0,750
Motor racing	0,553	0,601	0,677
Soccer	0,451	0,487	0,648
Tennis	0,817	0,813	0,880
General	0,810	0,857	0,966
Média	**0,515**	**0,547**	**0,678**
Acurácia	**0,538**	**0,571**	**0,680**

forma geral, os resultados indicam que as amostras criadas sinteticamente representam bem perguntas reais e, portanto, podem ser utilizadas para treinar e manter atualizados automaticamente mecanismos de seleção de domínios de QA.

Aprofundando a análise, pode-se verificar que o desempenho individual de alguns domínios foi bastante divergente, principalmente no Cenário 1. Por exemplo, a F-medida obtida pelo classificador no domínio *Sports* (Tabela 8) foi significativamente maior que em outros domínios, enquanto que a F-medida obtida no domínio *Politics* (Tabela 8) apresentou resultado bem inferior a média dos outros domínios.

Ao comparar os cenários avaliados, é possível notar que o Cenário 1, de forma geral, apresentou resultados inferiores. Dado que neste cenário as *stopwords* não foram removidas, o alto nível de ruído pode ter impactado negativamente o desempenho da representação computacional e do classificador. As Tabelas 7, 8 e 9 indicam que a remoção das *stopwords* (Cenário 2) claramente melhorou o desempenho da classificação.

Outra característica conhecida que pode afetar significativamente o desempenho da classificação é o fator de desbalanceamento entre as classes. Analisando a distribuição das

amostras (Tabela 4) percebe-se que o domínio *Politics*, que apresentou o pior desempenho em todos os cenários, é também o que tem menor número de amostras, representando 5% do total disponível, enquanto o cenário *Sports* que apresentou o melhor desempenho é representado por 35% do total de amostras.

Como os classificadores frequentemente beneficiam a classe majoritária, as classes desbalanceadas prejudicaram o resultado geral da classificação. Por esse motivo, conforme esperado, o balanceamento (Cenário 3) melhorou o desempenho, dado que a HF⇑ apresentada na Tabela 7 aumentou de 0,767 (Cenário 2) para 0,815 (Cenário 3).

Analisando a Tabela 9, pode-se constatar que o balançamento teve um efeito positivo em diversos subdomínios, especialmente em *Golf* no qual a F-medida saltou de 0,095 (Cenário 2) para 0,750 (Cenário 3); *Electronics*, que aumentou de 0,237 (Cenário 2) para 0,610 (Cenário 3); e *Literature*, que evoluiu de 0,406 (Cenário 2) para 0,784 (Cenário 3).

Ainda analisando a Tabela 9, nota-se que após o balançamento, alguns subdomínios apresentaram um desempenho notável, em relação a F-medida, destacando os subdomínios *Tennis* (0,880), *Market & Exchange* (0,920) e *Sports (general)* com F-medida de 0,966. O bom desempenho em vários cenários indica que as perguntas geradas automaticamente têm boa representatividade, sendo viáveis para o treinamento do classificador de domínios.

É importante recordar que todos os cenários apresentados foram treinandos com perguntas geradas sinteticamente e avaliados com perguntas reais. Logo, a origem das amostras é diferente. Essa metodologia difere dos cenários tradicionais de validação cruzada[13] nos quais um conjunto inicial de amostras de mesma origem é particionado em subconjuntos de treinamento e teste.

Relembrado isso, o desempenho em alguns domínios e subdomínios pode ser justificado pela divergência no conteúdo representado pelas perguntas dos conjuntos de treinamento e teste. Em resumo, entende-se que as perguntas reais não estão necessariamente delimitadas pelos mesmos conceitos que delimitam os domínios das perguntas sintéticas. Por exemplo, pode-se perceber esse comportamento no primeiro nível, analisando o desempenho das classes *Politics* e *Sports* (Tabela 8), os quais apresentam significativa diferença, e no segundo nível, com as classes *Finance* e *Tennis* (Tabela 9). Entende-se, portanto, que as perguntas reais dos conjuntos *Politics* e *Finance* podem estar menos correlacionadas aos assuntos presentes nos documentos que originaram as amostras sintéticas para o treinamento dos respectivos domínios.

Alguns subdomínios, como *Finance* e *Football* (Tabela 9), apresentaram pouca variação de desempenho entre os três cenários, enquanto outros, como *Literature* e *Golf* (Tabela 9) tiveram grande melhora após a remoção de *stopwords* e balanceamento das

[13] Holdout, k-fold ou leave-one-out.

amostras. Esse comportamento corrobora a hipótese de que o problema de desempenho é particular dos domínios em questão, e não da estratégia de geração automática de perguntas sintéticas.

Outra constatação é que a classificação no segundo nível hierárquico, de forma geral, é mais desafiadora que no primeiro nível, devido a complexidade inerente ao domínio que certamente tende a aumentar conforme os níveis vão se tornando mais profundos, conforme constatado também por Cesa-Bianchi, Gentile e Zaniboni (2006a). Por exemplo, é naturalmente mais simples classificar a pergunta *"What are the biggest money mistakes that small businesses commonly make?"* entre os domínios *Economy & Business & Financial* e *Sports*, do que classificar a mesma pergunta quanto ao subdomínios *Finance* e *Market & Exchange*.

Pelo menos seis subdomínios apresentaram F-medida igual ou superior a 80% (*e.g., Music*, *Market & Exchange*, *Economy & Business & Financial (general)*, *Natural Sciences*, *Tennis* e *Sports (general)*). Isso indica o potencial da proposta mesmo com o problema tornando-se mais complexo (devido ao segundo nível da hierarquia), e ainda incluindo potenciais erros de rotulação da fonte dos documentos que originaram as perguntas sintéticas.

Em resumo, apesar de alguns pormenores relacionados a delimitação do contexto dos domínios, os resultados indicam que a hipótese de pesquisa foi confirmada. Portanto, a estratégia de geração automática de perguntas sintéticas é eficiente e seguramente recomendável para o treinamento do classificador hierárquico de domínio de perguntas, que pode ser empregado na integração de múltiplos sistemas de QA de domínio restrito como um único sistema de QA de domínio aberto.

5.4 Ferramenta de classificação *online*

Uma ferramenta *online*[14] foi construída para permitir aos usuários experimentar o classificador desenvolvido e a capacidade do sistema. O modo de utilização é extremamente direto; sendo que o usuário precisa simplesmente digitar uma pergunta no idioma inglês e clicar no botão *"Classify"*, para que em seguida, o sistema classifique o domínio e subdomínio da pergunta apresentando o resultado em uma árvore de domínios.

A Figura 17 ilustra o resultado da classificação para a pergunta *"Who was Ayrton Senna?"*. Como esperado, o resultado da classificação foi *"Sports"*. O subdomínio também foi corretamente predito como *"Motor racing"* (Figura 18).

A ferramenta apresenta duas formas de visualizações da classificação: uma versão *flat* com somente um nível e a versão hierárquica com dois níveis. Na segunda, conforme

[14] *Question Domain Classifier*. Disponível em <http://bit.ly/qdcDemo>, acessado em 10/05/2018.

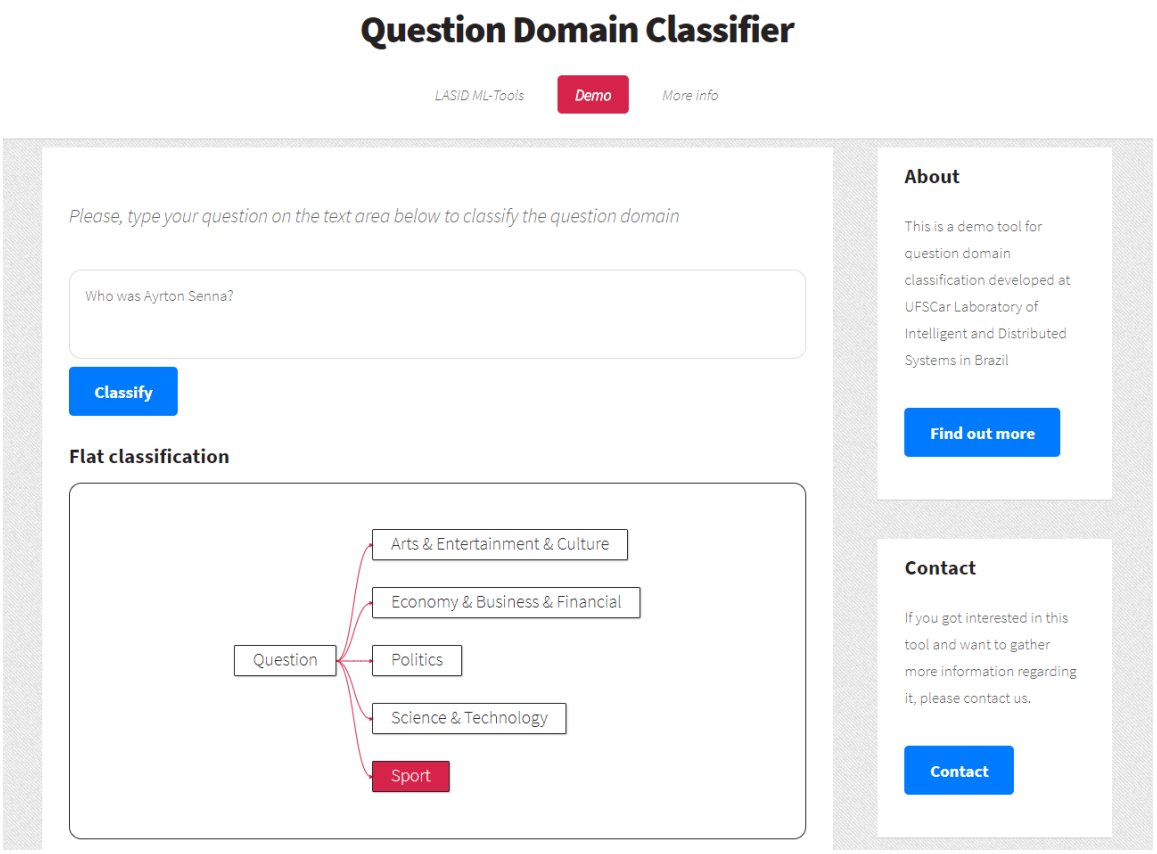


Figura 17 – Interface da ferramenta *online* de classificação de domínio de perguntas. Ao submeter a pergunta *"Who was Ayrton Senna?"* o domínio foi corretamente predito como *"Sports"*, destacado na árvore de domínios.

descrito no Capítulo 3, se não existir um domínio especializado suficiente, o domínio *General* é selecionado. Por exemplo, ao submeter a pergunta *"Is Curling an Olympic Sport?"*, no qual não existe nenhum subdomínio de *"Sports"* no nível dois que corresponda à pergunta, a predição se encerra no primeiro nível, com a seleção do domínio *"Sports (general)"*, conforme ilustrado na Figura 19.

De forma a implementar um processo de melhora contínua da capacidade preditiva do classificador, uma funcionalidade de *feedback* foi incorporada na ferramenta, como ilustrado na Figura 20. Assim, o usuário pode indicar se sua pergunta foi corretamente classificada ou informar o domínio e o subdomínio corretos entre os disponíveis.

A ferramenta desenvolvida permite que o usuário teste e comprove a eficiência do modelo proposto e a autenticidade dos resultados apresentados através de uma interface *online* de usabilidade simples. Através dela, é possibilitado ainda, que o usuário desenvolva uma noção intuitiva do funcionamento do mecanismo roteirizador que irá realizar a seleção de sistemas de QA. Ainda, caso o classificador falhe, é possível que o usuário dê *feedback* e realize correções de predições.

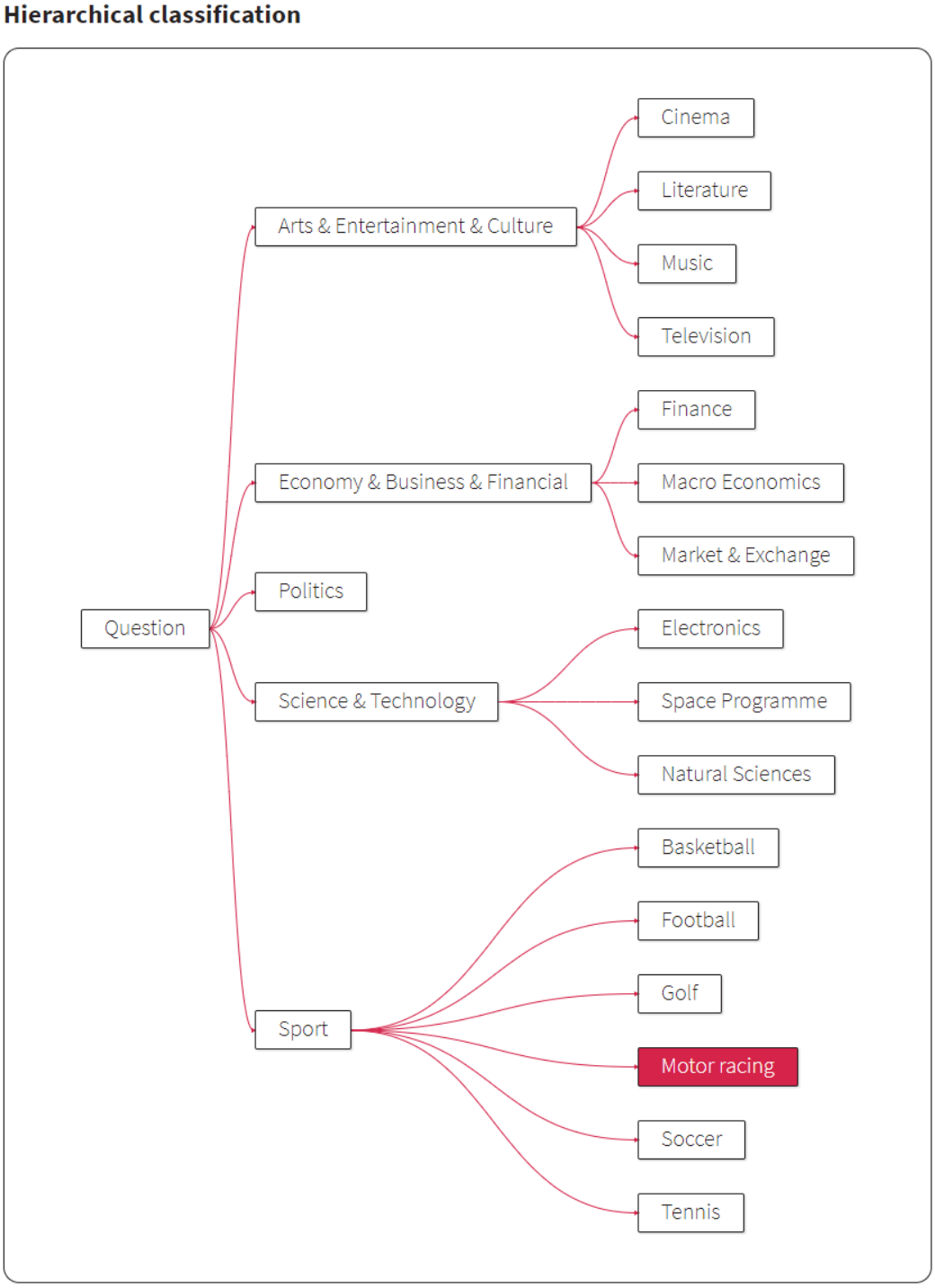


Figura 18 – Subdomínio predito pela ferramenta *online*. A pergunta *"Who was Ayrton Senna?"* teve seu subdomínio predito pelo classificador como *"Motor racing"*, destacado na árvore de domínios hierárquica.

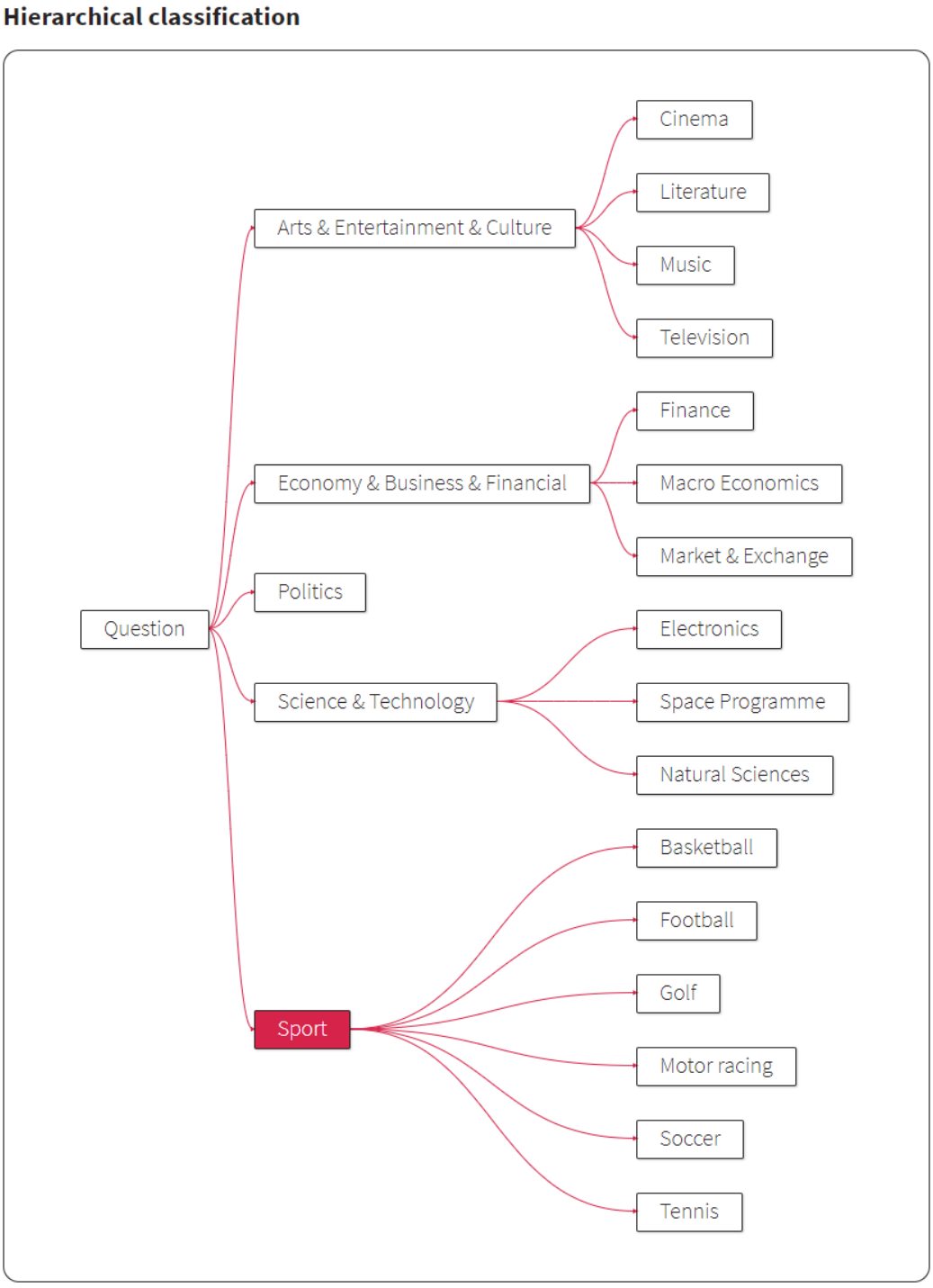


Figura 19 – Subdomínio *General* predito pela ferramenta *online*. A pergunta *"Is Curling an Olympic Sport?"* teve seu subdomínio predito pelo classificador como *"Sport (general)"*, destacado na árvore de domínios hierárquica. Nenhum subdomínio de *"Sports"* foi adequado o suficiente para a pergunta proposta, logo, o subdomínio *"General"* foi selecionado.

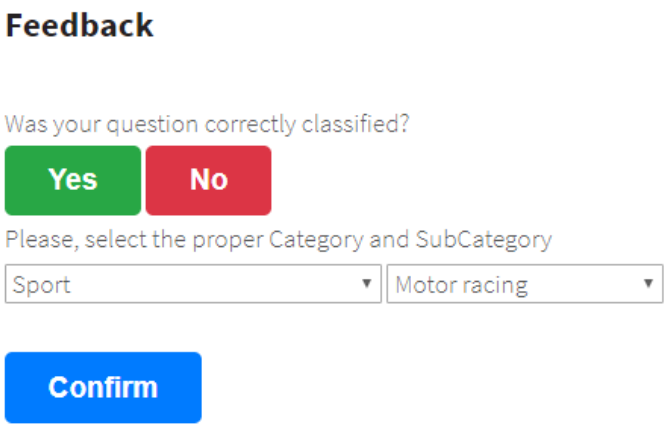


Figura 20 – Funcionalidade de *feedback* da ferramenta *online*. Através dela, o usuário pode indicar se a pergunta foi corretamente classificada e, em caso negativo, informar o domínio e subdomínio correto através de uma caixa de seleção.

5.5 Considerações finais

Este capítulo apresentou todos os passos executados, incluindo detalhes das técnicas e ferramentas utilizadas para realização dos experimentos. Os documentos coletados seguiram a pluralidade proposta nos capítulos anteriores, tanto quanto ao tema abordado, como no quesito temporal[15], contemplando diversos domínios e subdomínios.

Destacou-se a necessidade de variabilidade entre os documentos do *corpus* e a garantia de qualidade dos documentos originais, bem como das perguntas geradas sinteticamente, a fim de garantir a máxima qualidade da solução completa.

Detalhes sobre os conjuntos de dados envolvidos também foram revelados e discutidos. Os cenários experimentais foram apresentados e os resultados comparados e analisados. Uma ferramenta *online* que permite experimentar o classificador desenvolvido também foi apresentada.

Por fim, com base nos resultados experimentais, conclui-se que a hipótese — *criar automaticamente um conjunto rotulados de perguntas que possa ser empregado no treinamento de um classificador hierárquico de domínio* — é verdadeira e, portanto, as amostras sintéticas podem ser seguramente empregadas para treinar um classificador hierárquico de domínios de sistemas de QA.

[15] O intervalo entre as datas de publicação de alguns documentos supera 10 anos.

6 Conclusão

Sistemas de QA são amplamente utilizados e vêm ganhando cada vez mais notoriedade nos dias atuais, principalmente devido ao acelerado crescimento na utilização de assistente virtuais em *smartphones* e *smart speakers*.

A utilização desse tipo de sistema visa agregar valor para os usuários, simplificando e economizando tempo na realização de tarefas corriqueiras. Esses sistemas vêm evoluindo, de forma a atender as expectativas e desejos dos usuários que se tornam cada vez mais complexos, específicos e repletos de nuances.

Sistemas de QA de domínio restrito têm a capacidade de responder perguntas complexas, porém, se restringem a um único assunto de interesse. De forma antagônica, sistemas de QA de domínio aberto são capazes de responder sucintamente a perguntas sobre diversos assuntos. Idealmente, um sistema de QA deveria ter a amplitude de um sistema de domínio aberto e a profundidade de um de domínio restrito, para responder de forma precisa e com qualidade a perguntas sobre variados assuntos.

É possível integrar múltiplos QAs de domínio restrito para atuarem juntos, de forma transparente ao usuário, como um QA de domínio aberto. Nesse caso, é necessário empregar um sistema de classificação de domínio para selecionar o QA que deverá produzir a resposta. Portanto, a qualidade geral do sistema de QA depende diretamente do desempenho do classificador de domínio.

A etapa mais custosa e delicada dentre os processos de criação e manutenção de um sistema de classificação de domínio, como em diversas outras tarefas de aprendizado supervisionado, reside na coleta, preparo, seleção, rotulação manual e atualização de um conjunto representativo de amostras. Ainda sim, é desejável que a base de dados de treinamento seja ampla, atualizada e representativa em termos de amostras com diferentes contextos.

A fim de atender esses requisitos, este trabalho propôs uma forma de treinar um classificador a partir de uma base de dados criada de maneira automática usando um *corpus* de documentos. Para isso, foi utilizado um gerador de perguntas sintéticas capaz de produzir automaticamente milhões de amostras que foram utilizadas para treinar um classificador capaz de predizer o domínio de perguntas.

Esse gerador de perguntas sintéticas, além de criar um elevado número de amostras sintéticas, o faz incluindo os rótulos, representados por domínio e subdomínio. Essa estratégia reduz consideravelmente o esforço e tempo de preparação das bases de treinamento, visto que esse processo, que geralmente envolve diversas atividades a serem realizadas

manualmente, é realizado, neste trabalho, de forma completamente automática.

Para assegurar a qualidade da solução, o desempenho do modelo proposto foi avaliado usando um conjunto de perguntas reais, elaboradas por usuários de portais de perguntas e respostas, caracterizando o perfil de utilizadores de sistemas de QA.

Os resultados obtidos, com acurácia e F-medida hierárquica superiores a 81%, indicam que a proposta é apropriada e pode ser usada na prática como mecanismo de roteirização de perguntas para aplicação na seleção de sistemas de QA. De maneira geral, os resultados também evidenciaram que as amostras sintéticas possuem qualidade suficiente para obter bons resultados na predição de domínios em aplicações reais.

As vantagens apresentadas por esse modelo incluem: *(i)* a economia de tempo e esforço que seriam necessários para obter e manter atualizada uma significativa quantidade de perguntas reais rotuladas, necessárias para treinar o classificador de domínio, *(ii)* ele simplifica a extensão para novos domínios e subdomínios que venham a ser incorporados, agregando as qualidades de dinamismo e robustez, *(iii)* ele se baseia em documentos como fontes de informação para a geração das perguntas, sendo que estes, também podem servir como fonte de informação para os sistemas de QA e, por fim, *(iv)* o compartilhamento dos documentos entre o gerador de perguntas e os sistemas de QA permite que ambos atuem em harmonia quanto aos assuntos abordados, de forma que o classificador selecione sistemas de QA capazes de responder perguntas sobre os domínios em questão.

Por fim, destaca-se ainda que a classificação hierárquica de perguntas também pode ser utilizada em outras aplicações nas quais seja necessário categorizar amostras quanto a uma taxonomia. Citam-se como exemplos, sistemas de Perguntas Mais Frequentes (FAQ) e portais de perguntas e respostas, como os portais *Quora* e *Yahoo! Answers*, nos quais as perguntas enviadas pelos utilizadores são categorizadas quanto ao domínio.

6.1 Desafios e trabalhos futuros

Em trabalhos futuros, recomenda-se o estudo do problema de classificação com questões mais complexas, aplicando também técnicas de normalização léxica e indexação semântica, a fim de agregar mais atributos para a amostra (*e.g.*, sinônimos) e tratamento de gírias, símbolos e abreviações.

Recomenda-se ainda, estudar cenários nos quais as perguntas são propostas sequencialmente, em uma experiência semelhante a de um *chat*, no qual ocorre uma sequência de diálogos e considera informações sobre o contexto em que elas estão inseridas.

Por fim, também é sugerido avaliar o ganho de desempenho real de sistemas utilizando o mecanismo de classificação hierárquica proposto, através da integração de sistemas de QA existentes.

Apesar de existirem diversos sistemas de RDQAs e ODQAs, como apresentado no Capítulo 1, as implementações disponíveis em repositórios de código-fonte aberto são reduzidas. Para construir e avaliar um modelo completo da arquitetura aqui apresentada, que combina as características de ambos os tipos de sistema, alguns requisitos são fundamentais:

Publicação na Literatura Os QAs utilizados devem possuir resultados publicados na literatura, de forma a garantir resultados idôneos e relevantes para a avaliação do modelo integrado.

Código-fonte Idealmente, o código-fonte dos QAs deveria estar disponível publicamente em repositórios de código, de forma que seja possível reproduzir os resultados obtidos e integrar o classificador de domínio ao código existente.

Mínimo grau de integração Caso o código-fonte não esteja disponível, um mínimo grau de integração deve ser possível, seja via requisições Web, integração via arquivo ou BD, ou alguma forma de Interface de Programação de Aplicativos (API), de forma a permitir enviar a pergunta realizada pelo usuário e receber uma resposta.

Qualidade da resposta A resposta produzida pelo sistema deve ser relevante e relacionada com o foco da pergunta. Por exemplo, uma pergunta que inicie pelo pronome *"Who"* tem como resposta esperada uma entidade do tipo *"Pessoa"*, assim como uma pergunta que inicie pelo pronome *"When"* tem como resposta esperada uma entidade temporal (período ou data).

Ainda quanto a qualidade, espera-se que o sistema produza uma resposta relacionada ao contexto da pergunta e temporalmente válida. Por exemplo, ao se perguntar *"Who is the current president of Brazil?"* as respostas: *"Donald Trump"* e *"Juscelino Kubitschek"* estão erradas, a primeira, por estar descontextualizada e, a segunda, por ser temporalmente inválida.

Contextualização da resposta O sistema deve contextualizar o usuário quanto a resposta produzida. Conforme explicado por Lin et al. (2003), os usuários preferem sistemas que oferecem respostas com dimensão na ordem de um parágrafo em detrimento de uma resposta apenas de um termo.

Por exemplo, para a pergunta *"Which president of Brazil moved the Federal capital to Brasilia?"*, a resposta esperada seria algo como *"President* ***Juscelino Kubitschek*** *ordered the construction of Brasília, fulfilling an article of the country's constitution dating back to 1891 stating that the capital should be moved from Rio de Janeiro to a place close to the center of the country."*. Ainda que responder apenas *"Juscelino Kubitschek"* não seja incorreto, a primeira resposta é mais apropriada, pois permite que o usuário tenha maior convicção quanto a sua corretude.

Infelizmente, no período de desenvolvimento desta pesquisa, os sistemas de QA encontrados não se adequaram em um ou mais dos requisitos elencados acima. Dessa forma, utilizar algum desses sistemas não agregaria resultado relevante para a avaliação do modelo de forma integrada. Portanto, a evolução dos próprios sistemas de QA disponíveis é necessária para que o sistema de integração proposto possa tornar-se passível de ser utilizado na prática.

Publicações

TAVARES, L. L.; SILVA, R. M.; ALMEIDA, T.A.. **Towards automatically creating large labeled datasets for training question domain classifiers.** *Proceedings of the 2018 IEEE International Joint Conference on Neural Networks (IJCNN' 18)*, Rio de Janeiro, RJ, Brasil, 2018. (Aceito para publicação)

TAVARES, L. L.; SILVA, R. M.; ALMEIDA, T.A.. **Rumo à integração de múltiplos QAs de domínio restrito: sistema de classificação de domínio através das perguntas.** *XIV Encontro Nacional de Inteligência Artificial e Computacional (ENIAC'17)*, Uberlândia, MG, Brasil, 2017. v. 1. p. 1-12.

Referências

ALLAM, A. M. N.; HAGGAG, M. H. The question answering systems: A survey. *International Journal of Research and Reviews in Information Sciences*, v. 2, n. 3, p. 211–220, 2012. Citado 3 vezes nas páginas 5, 6 e 13.

ATHIRA, P.; SREEJA, M.; REGHURAJ, P. Architecture of an Ontology-Based Domain-Specific Natural Language Question Answering System. *International Journal of Web & Semantic Technology*, v. 4, n. 4, p. 31–39, 2013. ISSN 09762280. Citado 2 vezes nas páginas 5 e 7.

BENAMARA, F. Cooperative question answering in restricted domains: the WEBCOOP experiment. In: *Proceedings of the 2004 Annual Meeting of the Association for Computational Linguistics (ACL '04) - Workshop on Question Answering in Restricted Domains*. Barcelona, Espanha: Association for Computational Linguistics, 2004. p. 31–38. Citado na página 9.

BENNETT, P. N.; NGUYEN, N. Refined experts: Improving classification in large taxonomies. In: *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)*. Nova Iorque, NY, EUA: ACM, 2009. p. 11–18. ISBN 978-1-60558-483-6. Citado na página 24.

BROWN, J. S.; BURTON, R. R. Multiple representations of knowledge for tutorial reasoning. *Representation and understanding*, Academic Press, New York, p. 311–349, 1975. Citado na página 8.

CAI, L.; HOFMANN, T. Hierarchical document categorization with support vector machines. In: ACM. *Proceedings of the 13th ACM International Conference on Information and knowledge management (CIKM '04)*. Washington, DC, EUA: ACM, 2004. p. 78–87. Citado na página 28.

CERRI, R.; CARVALHO, A. C. P. L. F. C. de. New top-down methods using svms for hierarchical multilabel classification problems. In: IEEE. *The 2010 International Joint Conference on Neural Networks (IJCNN '10)*. Barcelona, Espanha, 2010. p. 1–8. Citado na página 28.

CESA-BIANCHI, N.; GENTILE, C.; ZANIBONI, L. Hierarchical classification: combining bayes with svm. In: ACM. *Proceedings of the 23rd international conference on Machine learning (ICML '06)*. Nova Iorque, NY, EUA, 2006. p. 177–184. Citado 2 vezes nas páginas 23 e 53.

CESA-BIANCHI, N.; GENTILE, C.; ZANIBONI, L. Incremental algorithms for hierarchical classification. *Journal of Machine Learning Research*, v. 7, n. Jan, p. 31–54, 2006. Citado na página 23.

CHAKRABARTI, S. et al. Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies. *The VLDB Journal*, v. 7, n. 3, p. 163–178, 1998. ISSN 0949-877X. Citado na página 25.

CHAUDHRI, V. K.; OVERHOLTZER, A.; SPAULDING, A. An intelligent textbook that answers questions. In: LAMBRIX, P. et al. (Ed.). *Knowledge Engineering and Knowledge Management (EKAW '15)*. Cham: Springer International Publishing, 2015. p. 131–135. ISBN 978-3-319-17966-7. Citado na página 10.

CHUNG, H. et al. A Practical QA System in Restricted Domains. In: *Proceedings of the 2004 Annual Meeting of the Association for Computational Linguistics (ACL '04) - Workshop on Question Answering in Restricted Domains*. Barcelona, Espanha: ACL, 2004. p. 39–45. Citado 4 vezes nas páginas 1, 9, 11 e 14.

CLARE, A.; KING, R. D. Predicting gene function in saccharomyces cerevisiae. *Bioinformatics*, Oxford University Press, v. 19, n. 2, p. ii42–ii49, 2003. Citado na página 27.

CLARKE, C. L. A.; TERRA, E. L. Passage retrieval vs. document retrieval for factoid question answering. In: *Proceedings of the 2003 Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval (SIGIR '03)*. Nova Iorque, NY, EUA: ACM, 2003. p. 427–428. ISBN 1-58113-646-3. Citado na página 9.

COSTA, E. et al. A review of performance evaluation measures for hierarchical classifiers. In: *Evaluation Methods for machine Learning II: papers from the AAAI-2007 Workshop, part of the 22nd Conference on Artificial Intelligence (AAAI '07)*. Vancouver, Canadá: AAAI Press, 2007. p. 1–6. Citado na página 29.

D'ALESSIO, S. et al. The effect of using hierarchical classifiers in text categorization. In: *Content-Based Multimedia Information Access (RIAO '00)*. Paris, France, France: Le Centre de Hautes Etudes Internationales D'Informatique Documentaire, 2000. v. 1, p. 302–313. Citado na página 22.

DEKEL, O.; KESHET, J.; SINGER, Y. Large margin hierarchical classification. In: ACM. *Proceedings of the 21st international conference on Machine learning (ICML '04)*. Banff, Alberta, Canadá, 2004. p. 27. Citado na página 28.

DIEKERMA, A. R.; YILMAZEL, O.; LIDDY, E. D. Evaluation of restricted domain Question-Answering systems. In: *Proceedings of the 2004 Annual Meeting of the Association for Computational Linguistics (ACL '04) - Workshop on Question Answering in Restricted Domains*. Barcelona, Espanha: ACL, 2004. p. 2–7. Citado na página 9.

DUMAIS, S.; CHEN, H. Hierarchical classification of web content. In: *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '00)*. Nova Iorque, NY, EUA: ACM, 2000. p. 256–263. ISBN 1-58113-226-3. Citado na página 25.

ESULI, A.; FAGNI, T.; SEBASTIANI, F. Boosting multi-label hierarchical text categorization. *Information Retrieval*, Springer, v. 11, n. 4, p. 287–313, 2008. Citado na página 23.

FRANK, A. et al. Question answering from structured knowledge sources. *Journal of Applied Logic*, v. 5, n. 1, p. 20 – 48, 2007. ISSN 1570-8683. Citado na página 10.

FREITAS, A. A.; CARVALHO, A. C. P. de Leon F. de. Tutorial on hierarchical classification with applications in bioinformatics. In: TANIAR, D. (Ed.). *Research and*

trends in data mining technologies and applications. Hershey, PA, EUA: IGI Global, 2006. v. 1, p. 175–208. ISBN 9781599042718. Citado na página 21.

GAUCH, S.; CHANDRAMOULI, A.; RANGANATHAN, S. Training a hierarchical classifier using inter document relationships. *Journal of the Association for Information Science and Technology*, Wiley Online Library, v. 60, n. 1, p. 47–58, 2009. Citado na página 25.

GHAZI, D.; INKPEN, D.; SZPAKOWICZ, S. Hierarchical versus flat classification of emotions in text. In: *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 2010 workshop on computational approaches to analysis and generation of emotion in text (NAACL '10)*. Los Angeles, CA, EUA: Association for Computational Linguistics, 2010. p. 140–146. Citado na página 25.

GOPAL, S. et al. Bayesian models for large-scale hierarchical classification. In: *Advances in Neural Information Processing Systems (NIPS '12)*. Stateline, NV, EUA: Curran Associates, Inc., 2012. v. 25, p. 2411–2419. Citado na página 28.

GREEN, B. F. et al. Baseball: An automatic question answerer. In: ACM. *Proceeding of the 1961 Western Joint IRE-AIEE-ACM computer conference (IRE-AIEE-ACM '61)*. Los Angeles, CA, EUA, 1961. v. 19, p. 219–224. Citado 2 vezes nas páginas 1 e 8.

HE, H.; GARCIA, E. A. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 21, n. 9, p. 1263–1284, 2009. Citado na página 50.

HEILMAN, M.; SMITH, N. A. *Question generation via overgenerating transformations and ranking*. Pittsburgh, PA, EUA, 2009. Citado 2 vezes nas páginas 37 e 38.

INDURKHYA, N.; DAMERAU, F. J. *Handbook of Natural Language Processing*. Boca Raton, FL, EUA: Chapman & Hall / CRC, 2010. ISBN 9781420085921. Citado 3 vezes nas páginas 15, 6 e 7.

IPEIROTIS, P. G.; GRAVANO, L.; SAHAMI, M. Probe, count, and classify: categorizing hidden web databases. In: ACM. *Proceedings of the 2001 ACM SIGMOD international conference on Management of data (SIGMOD '01)*. Santa Bárbara, CA, EUA, 2001. v. 30, n. 2, p. 67–78. Citado na página 29.

ITTYCHERIAH, A.; FRANZ, M.; ROUKOS, S. IBM's statistical question answering system-TREC-10. In: *The 10th Text REtrieval Conference (TREC '01)*. Gaithersburg, Maryland: NIST, 2001. p. 258–264. Citado na página 35.

JIN, B. et al. Multi-label literature classification based on the gene ontology graph. *BMC bioinformatics*, BioMed Central, v. 9, n. 1, p. 525, 2008. Citado na página 23.

JU, Q.; MOSCHITTI, A.; JOHANSSON, R. Learning to rank from structures in hierarchical text classification. In: *European Conference on Information Retrieval (ECIR '13)*. Moscou, Russia: Springer, 2013. p. 183–194. Citado na página 28.

KANDO, N. Overview of the fifth ntcir workshop. In: *Proceedings of the 2005 NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access (NTCIR '05)*. Tóquio, Japão: National Institute of Informatics, 2005. Citado na página 10.

KIRITCHENKO, S.; MATWIN, S.; FAMILI, F. Functional annotation of genes using hierarchical text categorization. In: *Proceedings of the ACL Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics (ACL-ISMB '05)*. Detroit, MI, EUA: ACL, 2005. Citado 5 vezes nas páginas 16, 28, 30, 31 e 33.

KOLLER, D.; SAHAMI, M. Hierarchically classifying documents using very few words. In: *Proceedings of the 1997 International Conference on Machine Learning (ICML '97)*. São Francisco, CA, EUA: Morgan Kaufmann Publishers Inc., 1997. p. 170–178. ISBN 1-55860-486-3. Citado 2 vezes nas páginas 25 e 46.

KOTSIANTIS, S. B.; ZAHARAKIS, I.; PINTELAS, P. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, v. 160, p. 3–24, 2007. Citado na página 19.

KRIEGEL, H.-P. et al. Using support vector machines for classifying large sets of multi-represented objects. In: *Proceedings of the 2004 SIAM International Conference on Data Mining (SIAM '04)*. Lake Buena Vista, FL, EUA: Society for Industrial and Applied Mathematics, 2004. p. 102–113. Citado na página 25.

KUMAR, A. et al. Ask me anything: Dynamic memory networks for natural language processing. In: BALCAN, M. F.; WEINBERGER, K. Q. (Ed.). *Proceedings of The 33rd International Conference on Machine Learning (ICML '16)*. Nova Iorque, NY, EUA: PMLR, 2016. (Proceedings of Machine Learning Research, v. 48), p. 1378–1387. Citado na página 10.

KUPIEC, J. MURAX: A robust linguistic approach for question answering using an on-line encyclopedia. In: ACM. *Proceedings of the 1993 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93)*. Pittsburgh, PA, EUA, 1993. p. 181–190. Citado na página 9.

KURZWEIL, R. *How to create a mind: The secret of human thought revealed*. Londres, Inglaterra: Penguin, 2013. 352 p. Citado 3 vezes nas páginas 2, 14 e 16.

LABROU, Y.; FININ, T. Yahoo! as an ontology: Using yahoo! categories to describe documents. In: *Proceedings of the Eighth International Conference on Information and Knowledge Management (CIKM '99)*. Nova Iorque, NY, EUA: ACM, 1999. p. 180–187. ISBN 1-58113-146-1. Citado na página 28.

LEHNERT, W. Human and computational question answering. *Cognitive Science*, Lawrence Erlbaum Associates, Inc., v. 1, n. 1, p. 47–73, 1977. ISSN 1551-6709. Citado na página 8.

LIN, C.-Y.; HOVY, E. Identifying topics by position. In: *Proceedings of the 5th Conference on Applied Natural Language Processing (ANLC '97)*. Washington, DC, EUA: Association for Computational Linguistics, 1997. p. 283–290. Citado na página 39.

LIN, J. et al. The role of context in question answering systems. In: ACM. *Proceeding of 2003 Conference on Human Factors in Computing Systems (CHI '03)*. Fort Laurderdale, FL, EUA, 2003. p. 1006–1007. Citado 3 vezes nas páginas 9, 13 e 61.

LU, W.; CHENG, J.; YANG, Q. Question answering system based on web. In: *Proceedings of the 2012 International Conference on Intelligent Computation Technology and*

Automation (ICICTA '12). Zhangjiajie, China: IEEE Computer Society, 2012. p. 573–576. Citado 3 vezes nas páginas 5, 6 e 7.

MCCALLUM, A.; NIGAM, K. et al. A comparison of event models for naive bayes text classification. In: *The 15th workshop on learning for text categorization (AAAI '98)*. Madison, WI, EUA: AAAI Press, 1998. v. 752, p. 41–48. Citado 2 vezes nas páginas 25 e 46.

MIAO, Y.; SU, X.; LI, C. Improving question answering based on query expansion with Wikipedia. In: IEEE. *Proceedings of the 2010 IEEE International Conference on Tools with Artificial Intelligence (ICTAI '10)*. Arras, França, 2010. v. 2, p. 233–240. Citado na página 10.

MINOCK, M. Where are the 'killer applications' of restricted domain question answering. In: *Proceedings of the IJCAI Workshop on Knowledge Reasoning in Question Answering (IJCAI '05)*. Edimburgo, Escócia: AAAI Press, 2005. p. 4. Citado 3 vezes nas páginas 6, 7 e 9.

MLADENIĆ, D.; GROBELNIK, M. Feature selection on hierarchy of web documents. *Decision Support Systems*, Elsevier, v. 35, n. 1, p. 45–87, 2003. Citado na página 23.

MOLLÁ, D.; VICEDO, J. L. Question Answering in Restricted Domains: An Overview. *Computational Linguistics*, Cambridge, MA, EUA, v. 33, n. 1, p. 41–61, 2007. ISSN 0891-2017. Citado 5 vezes nas páginas 6, 7, 8, 9 e 13.

NGUYEN, H. M.; COOPER, E. W.; KAMEI, K. Online learning from imbalanced data streams. In: IEEE. *Proceedings of the 3rd International Conference of the Soft Computing and Pattern Recognition (SoCPaR '11)*. Dalian, China, 2011. p. 347–352. Citado na página 46.

NYBERG, E.; FREDERKING, R. JAVELIN: a flexible, planner-based architecture for question answering. In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL '03)*. Edmonton, Canadá: Association for Computational Linguistics, 2003. v. 4, p. 19–20. Citado na página 9.

NYBERG, E. et al. Extending the JAVELIN QA system with domain semantics. In: *Proceedings of the 2005 Conferece of the Association for Advancement of Artificial Intelligence (AAAI '05) - Question Answering in Restricted Domains Workshop*. Pittsburgh, PA, EUA: AAAI Press, 2005. Citado na página 9.

OH, H.-S.; JUNG, Y. Enhancing the narrow-down approach to large-scale hierarchical text classification with category path information. *Journal of Information Science Theory and Practice*, v. 5, p. 31–47, 09 2017. Citado na página 28.

PENG, H. et al. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In: *Proceedings of the 2018 World Wide Web Conference on World Wide Web (WWW '18)*. Lyon, França: International World Wide Web Conferences Steering Committee, 2018. p. 1063–1072. Citado na página 28.

PENG, X.; CHOI, B. Document Classifications Based on Word Semantic Hierarchies. *The IASTED International Conference on Artificial Intelligence and Applications (AIA' 05)*, p. 362–367, 2005. Citado na página 28.

PUNERA, K.; GHOSH, J. Enhanced hierarchical classification via isotonic smoothing. In: *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*. Nova Iorque, NY, EUA: ACM, 2008. p. 151–160. ISBN 978-1-60558-085-2. Citado na página 24.

QIU, X.; GAO, W.; HUANG, X. Hierarchical multi-class text categorization with global margin maximization. In: *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers (ACLShort '09)*. Stroudsburg, PA, EUA: Association for Computational Linguistics, 2009. p. 165–168. Citado na página 28.

RAMPRASATH, M.; HARIHARAN, S. A survey on question answering system. *International Journal of Research and Reviews in Information Sciences*, v. 2, n. 1, p. 171–179, 2012. Citado na página 6.

ROUSU, J. et al. Learning hierarchical multi-category text classification models. In: ACM. *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*. Bonn, Alemanha, 2005. p. 744–751. Citado na página 28.

RUIZ, M. E.; SRINIVASAN, P. Hierarchical text categorization using neural networks. *Information Retrieval*, Springer, v. 5, n. 1, p. 87–118, 2002. Citado na página 25.

SCHAPIRE, R. E.; SINGER, Y. Boostexter: A boosting-based system for text categorization. *Machine learning*, Springer, v. 39, n. 2-3, p. 135–168, 2000. Citado na página 23.

SILLA JR., C. N.; FREITAS, A. A. A global-model naive bayes approach to the hierarchical prediction of protein functions. In: IEEE. *The 2009 IEEE International Conference on Data Mining (ICDM '09)*. Miami, FL, EUA, 2009. p. 992–997. Citado na página 46.

SILLA JR., C. N.; FREITAS, A. A. A survey of hierarchical classification across different application domains. *Data Mining Knowledge Discovery*, Kluwer Academic Publishers, Hingham, MA, EUA, v. 22, n. 1-2, p. 31–72, jan. 2011. ISSN 1384-5810. Citado 9 vezes nas páginas 15, 16, 19, 20, 21, 23, 24, 26 e 27.

SIMMONS, R. F. Answering english questions by computer: A survey. *Communications of the ACM*, ACM, Nova Iorque, NY, EUA, v. 8, n. 1, p. 53–70, jan. 1965. ISSN 0001-0782. Citado 2 vezes nas páginas 1 e 5.

SUN, A.; LIM, E.-P. Hierarchical text classification and evaluation. In: *Proceedings 2001 IEEE International Conference on Data Mining (ICDM '01)*. Washington, DC, EUA: IEEE Computer Society, 2001. p. 521–528. ISBN 0769511198. Citado 2 vezes nas páginas 21 e 22.

SUN, A.; LIM, E.-P.; NG, W.-K. Performance measurement framework for hierarchical text classification. *Journal of the Association for Information Science and Technology*, Wiley Online Library, v. 54, n. 11, p. 1014–1028, 2003. Citado na página 22.

SUN, A. et al. Blocking reduction strategies in hierarchical text classification. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 16, n. 10, p. 1305–1308, 2004. Citado na página 23.

TIKK, D.; BIRÓ, G. Experiment with a hierarchical text categorization method on the wipo-alpha patent collection. In: IEEE. *Fourth International Symposium on Uncertainty Modeling and Analysis, 2003 (ISUMA '03)*. College Park, MD, EUA, 2003. p. 104–109. Citado na página 25.

TIKK, D.; BIRÓ, G.; TÖRCSVÁRI, A. A hierarchical online classifier for patent categorization. In: *Emerging technologies of text mining: Techniques and applications*. Hershey, PA, EUA: IGI Global, 2007. cap. 12, p. 244–267. Citado na página 25.

TSUR, O.; RIJKE, M. de; SIMA'AN, K. Biographer: Biography Questions as a Restricted Domain Question Answering Task. In: *Proceedings of the 2004 Annual Meeting of the Association for Computational Linguistics (ACL '04) - Workshop on Question Answering in Restricted Domains*. Barcelona, Espanha: ACL, 2004. p. 23–30. Citado na página 9.

TURING, A. M. Computing machinery and intelligence. *Mind*, v. 59, n. 236, p. 433–460, 1950. Citado na página 5.

VALLIN, A. et al. Overview of the CLEF 2005 multilingual question answering track. In: *Workshop of the Cross-Language Evaluation Forum for European Languages (CLEF '05)*. Viena, Áustria: Springer, 2005. p. 307–331. Citado na página 10.

VAPNIK, V. N. An overview of statistical learning theory. *IEEE transactions on neural networks*, IEEE, v. 10, n. 5, p. 988–999, 1999. Citado na página 28.

VOORHEES, E. M. The TREC question answering track. *Natural Language Engineering*, Cambridge Univ Press, v. 7, n. 04, p. 361–378, 2001. Citado na página 8.

WANG, K.; ZHOU, S.; LIEW, S. C. Building hierarchical classifiers using class proximity. In: *Proceedings of the 25th International Conference on Very Large Data Bases (VLDB '99)*. Edimburgo, Escócia: Morgan Kaufmann Publishers, 1999. v. 99, p. 363–374. Citado na página 28.

WEIGEND, A. S.; WIENER, E. D.; PEDERSEN, J. O. Exploiting hierarchy in text categorization. *Information Retrieval*, Springer, v. 1, n. 3, p. 193–216, 1999. Citado na página 25.

WILENSKY, R. et al. The berkeley UNIX consultant project. *Computational Linguistics*, MIT Press, Cambridge, MA, EUA, v. 14, n. 4, p. 35–84, dez. 1988. ISSN 0891-2017. Citado na página 8.

WOODS, W. A. Progress in natural language understanding: an application to lunar geology. In: ACM. *Proceedings of National Computer Conference and Exposition (AFIPS '73)*. Nova Iorque, NY, EUA, 1973. p. 441–450. Citado na página 8.

WU, F.; ZHANG, J.; HONAVAR, V. Learning classifiers using hierarchically structured class taxonomies. *Abstraction, Reformulation and Approximation*, Springer, p. 902–902, 2005. Citado 2 vezes nas páginas 19 e 20.

XUE, G.-R. et al. Deep classification in large-scale text hierarchies. In: ACM. *Proceedings of the 31st Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR '08)*. Singapura, Singapura, 2008. p. 619–626. Citado na página 24.

YANG, Y.; PEDERSEN, J. O. A comparative study on feature selection in text categorization. In: *Proceedings of the Fourteenth International Conference on Machine Learning (ICML '97)*. Nashville, TN, EUA: Morgan Kaufmann Publishers, 1997. v. 97, p. 412–420. Citado na página 45.

ZHOU, D.; XIAO, L.; WU, M. Hierarchical classification via orthogonal transfer. In: *Proceedings of the 28th International Conference on Machine Learning (ICML '11)*. Bellevue, WA, EUA: Omnipress, 2011. p. 801–808. Citado na página 26.