TÉCNICO LISBOA

Extração de Informação Biológica de Artigos Científicos

Joana Alexandra Pimenta Gomes

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática e Computadores

Orientadores: Prof. Pável Pereira Calado
Prof. Tiago Morais Delgado Domingos

Júri

Presidente: Prof. Ernesto José Marques Morgado
Orientador: Prof. Pável Pereira Calado
Vogal: Prof. Bruno Emanuel da Graça Martins

Novembro 2016

Dedicado aos meus pais, ao meu irmão Carlos, à Cátia e ao Pedro

Agradecimentos

Primeiramente, gostaria de agradecer aos meus pais que sempre me apoiaram ao longo de todo o meu percurso académico e permitiram que eu pudesse estar hoje a mudar de rumo para um caminho mais promissor.

Agradeço ao meu irmão Carlos, que acreditou sempre em mim e me deu forças para continuar em frente e enfrentar todos os obstáculos, juntamente com a Cátia.

Agradeço ao Pedro, por ter estado sempre do meu lado e ter sido compreensivo nos momentos que foi necessário fazer sacrifícios em prol deste trabalho.

Agradeço ao coorientador Professor Tiago Domingos, ao Gonçalo Marques e ao Carlos Teixeira por me terem dado suporte em assuntos mais específicos da área da Biologia e por terem participado em discussões que me permitiram aprender e desenvolver este trabalho.

Por fim, agradeço ao meu orientador, Professor Pável Calado, que me auxiliou durante todo este ano, apresentando-me desafios e me guiando à procura de resoluções para convergir até este trabalho final.

Resumo

Ao longo dos anos, muitos trabalhos científicos têm sido publicados na área da Biologia a fim de compreender e antecipar os efeitos das mudanças globais que contribuem para a redução drástica da biodiversidade na Terra. Contudo, existe uma enorme dispersão do conhecimento e torna-se difícil o estudo aprofundado de cada espécie pois a informação é normalmente disseminada por muitos artigos diferentes. Com a evolução das tecnologias, técnicas de *Text Mining* têm sido desenvolvidas e utilizadas a fim de extrair automaticamente dados relevantes a partir de textos, imagens e gráficos.

Neste trabalho o objetivo principal é extrair informação sobre aves, presente em artigos científicos tentando responder à questão, "Será possível construir um sistema que possa extrair automaticamente dados de determinadas espécies de aves a partir de artigos científicos?". Para desenvolver a nossa solução criámos um sistema que procede à análise do texto através da combinação de técnicas de Processamento de Língua Natural, Expressões Regulares e algoritmos de Aprendizagem Automática. O sistema recebe, como entrada, o conjunto de artigos a analisar e, como resultado, apresenta os possíveis valores para as características da espécie que queremos observar (temperatura corporal, massa corporal, entre outros).

Como principal conclusão deste trabalho, demonstrámos que é possível construir um sistema para a extração de dados a partir de artigos científicos no domínio da Biologia. Contudo, ainda não é possível ter um sistema completamente automático tornando-se relevante um utilizador humano que possa resolver ambiguidades nos resultados.

Palavras-chave: Base de Conhecimento, *Slot Filling*, *Text Mining*, Extração de Informação, Aprendizagem Automática, Processamento de Língua Natural, Expressões Regulares

Abstract

During the years, many scientific documents have been submitted in the area of Biology in order to understand and anticipate the effects of global warming in drastically reducing the biodiversity of Earth. Besides this, there is a huge dispersion of knowledge and it becomes difficult to deeply study each species as the information is usually spread over different articles. With the evolution of technologies, text mining techniques have been developed and used in order to extract automatically relevant data from texts, images and charts.

In this work the main objective is to extract information on birds that are present in scientific articles trying to answer the question: "Is it possible to build a system that may extract automatically data regarding specific bird species from scientific articles?" To develop our solution, we created a system that analyses text through a combination of techniques of natural language processing, regular expressions and automatic learning algorithms. The system receives, as input, the set of documents to analyze and as a result it presents the possible values to the characteristics of the species that we want to analyze (body temperature, body mass, among others).

As main conclusion to this work, we demonstrated that it is possible to build a system that extracts data from scientific documents in the Biology domain. However, it is not yet possible to have a fully automatic process, being relevant to have a human user that may solve the result ambiguity.

Keywords: Knowledge Bases, Slot Filling, Text Mining, Information Extraction, Machine Learning, Natural Language Processing, Regular Expressions

Conteúdo

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

No Planeta Terra existem diversas formas de vida: seres humanos, animais, plantas, microrganismos e toda a multiplicidade de genes que os compõem e os diferenciam. Todos estes seres constituem a biodiversidade do Planeta e, ao longo dos séculos, a existência de algumas espécies tem vindo a ser ameaçada devido a vários fatores, nomeadamente, ambientais (poluição, incêndios) e caça. Com a extinção de variadas espécies tem aumentado a preocupação com a Natureza e com a sua diversidade biológica, tornando-se evidente a necessidade de desenvolver modelos para compreender e antecipar os efeitos das mudanças globais sobre a biodiversidade [52].

1.1 Motivação e Problema

A comunidade científica tem realizado muitos trabalhos de campo e têm sido publicados artigos sobre as diversas características dos seres biológicos do Planeta. Sendo a Biologia a ciência responsável por esta área de estudo, esta é de facto, uma ciência que se divide em vários ramos (por exemplo: zoologia, microbiologia, genética, fisiologia, biotecnologia, entre outras) o que contribui para uma enorme dispersão de informação relativa a uma determinada espécie [9].

Perante a existência de diversos artigos científicos nos vários ramos da Biologia, tornou-se difícil e longo o estudo aprofundado ao nível do grupo taxonómico e também todo o processo de agregação da informação em artigos científicos de revisão ou em bases de dados, pois seria necessário a leitura de um enorme número de documentos por especialistas. Contudo, durante o Século XX, iniciou-se a evolução da teoria da computação, a criação dos computadores e desenvolvimento de áreas computacionais como a Inteligência Artificial (IA) [39] e Processamento de Língua Natural (PLN) [22]. Por sua vez, estas áreas podem ter bastantes aplicações em Biologia, nomeadamente na migração do vasto conhecimento existente para um formato processável computacionalmente, para que possa ser encontrada a informação desejada automaticamente [47].

Para dar resposta às necessidades de informação da Biologia, têm sido criados sistemas para extração de informação e constituição de bases de dados. Na Internet podemos encontrar diversas bases de dados dedicadas aos vários ramos da Biologia. Por exemplo, no caso da biodiver-

sidade, encontramos a *Encyclopedia of Life* [1] que mantém dados sobre cada espécie e permite o acesso ao conhecimento sobre a vida na Terra; *Animal DiversityWeb* [2] mantém informação sobre a história da natureza, distribuição, classificação e conservação de espécies; e *ARKive* [3] que mantém informações, filmes e fotografias da biodiversidade na Terra. No entanto, tomando como exemplo o grupo taxonómico das aves, a maioria das bases de dados agrega informação sobre a sua fisionomia (valores de massa, comprimento), sendo raros os dados da sua fisiologia (temperatura média do corpo ou conteúdo hídrico).

Para obter os dados fisiológicos do grupo taxonómico das aves, propomos dar resposta ao problema que pode ser formulado na seguinte questão: "Será possível construir um sistema que possa extrair dados de determinadas espécies de aves a partir de artigos científicos?". Para dar resposta à questão formulada, definimos como principal objetivo construir um sistema para extrair informação sobre dados fisiológicos de aves provenientes de artigos científicos especializados.

1.2 Hipótese e Contributos

Neste trabalho deparamo-nos com algumas questões sobre a interpretação e extração de informação em textos não estruturados e, com a construção do nosso sistema, propomos encontrar soluções para que a extração de informação seja o mais correta possível utilizando técnicas de Processamento de Língua Natural, Expressões Regulares e algoritmos de Aprendizagem Automática.

As contribuições mais inovadoras deste trabalho são a obtenção de dados biológicos que até então não têm tido grande foco para a construção das bases de dados de Biologia existentes e a possibilidade de simplificar o trabalho dos investigadores a obter a informação necessária para as suas investigações. Apesar do foco deste sistema ser os dados das aves, este sistema pode ser generalizado a outros contextos mediante os dados e características a obter, o que poderá contribuir não só para a obtenção de informação para os investigadores biológicos, como também para investigadores de outros ramos.

1.3 Estrutura do Documento

O trabalho está dividido em quatro capítulos.

No Capítulo 2 apresentamos os conceitos relevantes à compreensão de todo o trabalho, a arquitetura de um sistema de Extração de Informação, os seus constituintes, o povoamento de Bases de Conhecimento e as métricas de avaliação utilizadas para avaliar os sistemas.

No Capítulo 3 expomos o trabalho relacionado relativamente aos principais sistemas dentro das várias abordagens de Extração de Informação e de povoamento de Bases de Conhecimento.

No Capítulo 4 mostramos como foi implementada a nossa solução apresentando a arquitetura do sistema e os vários módulos que o constituem.

[1] `https://www.eol.org/`
[2] `https://www.animaldiversity.org`
[3] `https://www.arkive.org`

No Capítulo 5 apresentamos os principais resultados obtidos ao longo do estudo e, por fim, no Capítulo 6 concluímos o trabalho refletindo sobre os principais pontos de todo o trabalho e apresentamos sugestões para melhorias futuras.

Capítulo 2

Conceitos

Neste capítulo são apresentados os conceitos que consideramos relevantes para o entendimento deste documento, nomeadamente, os conceitos de Bases de Conhecimento, bem como o conceito de *Text Mining* e conceitos relativos à Extração de Informação. Apresentamos a arquitetura de um sistema de Extração de Informação e explicamos as várias tarefas adjacentes. Expomos e analisamos as diferentes abordagens para a realização das várias tarefas de Extração de Informação que são parte integrante dos sistemas utilizados para criação e povoamento de Bases de Conhecimento e por fim apresentamos as métricas de avaliação utilizadas.

2.1 Bases de Conhecimento

O conhecimento é resultado do entendimento de diversas informações e dados subjacentes que estão interligados. Assim sendo, uma Base de Conhecimento refere-se a um conjunto de conhecimentos adquiridos sobre um determinado assunto ou domínio, sendo que os dados referentes a esses conhecimentos podem estar guardados numa base de dados relacional onde os dados se relacionam entre si [53].

As Bases de Conhecimento têm como objetivo facilitar a acessibilidade à informação inerente ao conhecimento e, sendo o conhecimento um conceito complexo que agrega variadas informações, os sistemas existentes para criar estas estruturas baseiam-se em processos que lidam com grandes quantidades de dados, de várias fontes (textos, tabelas, imagens ou diagramas), tamanhos e tipos [23].

De modo a obter os dados relevantes para as Bases de Conhecimento torna-se imperativo referir o conceito de *Text Mining* que é o processo para a obtenção de informação a partir de dados não estruturados, implicando técnicas de identificação, extração, gestão, integração e interpretação de dados de forma automática [2]. Um sistema de *Text Mining* começa por reunir todos os textos relevantes ao seu propósito (com a aplicação de técnicas de Recuperação de Informação [5]), identifica e extrai as informações desejadas (com a aplicação de técnicas e algoritmos de Extração de Informação [33]) e, por fim, os dados são interpretados e são encontradas associações entre estes para que possam ser utilizados para auxiliar em tomadas de decisão [2] (com a aplicação de técnicas de Data Mining [16]).

2.2 Extração de Informação

A área de Extração de Informação surge na comunidade de Processamento de Língua Natural, especificamente em duas competições de grande importância, *Message Understanding Conference* [1] e *Automatic Content Extraction* [2], com o objetivo de extrair automaticamente informação estruturada a partir de documentos não estruturados [40]. Inicialmente começaram por identificar apenas entidades, nomeadamente, nomes de pessoas e organizações e, de seguida, passaram a identificar também as relações entre as entidades [40]. Com a evolução das técnicas de extração e o aumento dos domínios que usufruem destas técnicas, os sistemas de extração têm sido utilizados para descobrir outro tipo de entidades, nomeadamente na área da Biologia: entidades biológicas (nome de espécies, de animais ou de células), suas características e relações [19].

2.2.1 Arquitetura de um Sistema de Extração de Informação

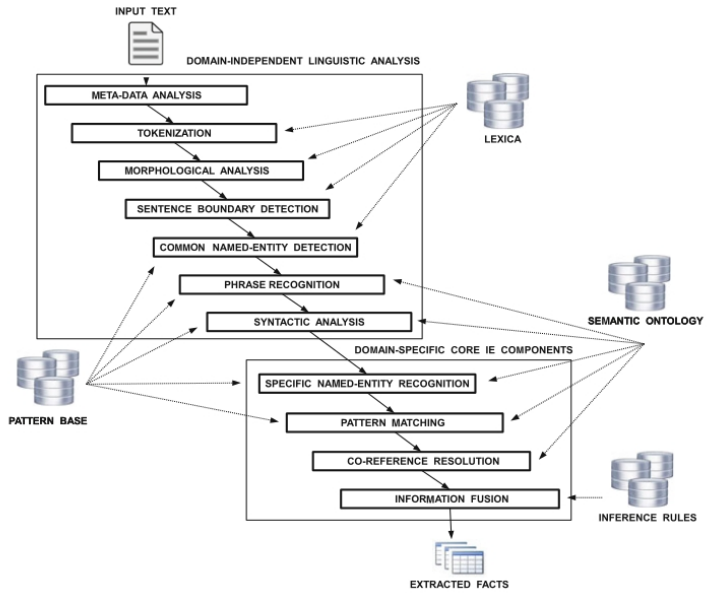

Figura 2.1: Arquitetura de referência de um sistema de Extração de Informação [35].

Num sistema de Extração de Informação, os documentos dos quais se deseja extrair a informação correspondem aos dados de entrada. Estes documentos têm que estar em formato digital para que a informação e os dados neles integrantes possam ser analisados. Para tal, é necessário utilizar tecnologias de Reconhecimento Ótico de Caracteres para proceder à conversão [14]. De forma a que o sistema saiba o que extraír, também é necessário fornecer um modelo com a definição das entidades e os campos a serem extraídos. Opcionalmente podem ser usadas Bases de Conhecimento, dicionários, glossários ou ontologias para identificar entidades. Na conclusão do processo de Extração de Informação, são obtidos dados num formato estruturado que podem ser posteriormente analisados [46].

[1] http://www-nlpir.nist.gov/related_projects/muc/index.html
[2] http://www.itl.nist.gov/iad/mig/tests/ace/

Cada sistema de Extração de Informação é construído para responder a questões de diferentes domínios. Contudo, apesar de poderem ter algumas diferenças significativas, existem determinados componentes comuns a todos os sistemas. Na figura 2.1 apresentamos a arquitetura de um sistema de Extração de Informação segundo Piskorski e Yangarber [35].

De acordo com a figura 2.1, podemos considerar no sistema de Extração de Informação dois grandes grupos de componentes linguísticos: os independentes do domínio e os dependentes do domínio. Relativamente às componentes independentes do domínio, estas são:

- **Análise de Metadados (*Meta-data Analysis*):** Análise da estrutura do documento identificando o título e o corpo do texto. Também é analisada a estrutura do texto relativa a parágrafos.
- **Tokenização (*Tokenization*):** Segmentação do texto em *token* (palavras) e classificação de cada *token* consoante a presença de maiúsculas, minúsculas, sinais de pontuação ou números.
- **Análise Morfológica (*Morphological Analysis*):** Análise e Extração de Informação morfológica de cada *token* através da análise gramatical (*Part of Speech*), ou seja, identificação de substantivo, adjetivo, artigo, pronome, numeral, verbo, advérbio, preposição, conjunção e interjeição.
- **Deteção de frases (*Sentence Boundary Detection*):** Segmentação do texto em frases, sendo cada uma representada por uma sequência de itens lexicais e a análise morfológica obtida na fase anterior.
- **Deteção de entidades nomeadas (*Common Named-entity Detection*):** Detetar as entidades independentemente do domínio, tais como, expressões temporais, numéricas ou referências geográficas.
- **Reconhecimento de frases (*Phrase Recognition*):** Identificação em pequena escala (frases) de estruturas locais, tais como, grupos nominais, grupos verbais, frases preposicionais, siglas e abreviaturas.
- **Análise sintática (*Syntactic Analysis*):** Análise da estrutura das frases identificando sujeito, predicado, complemento direto ou indireto. Esta análise pode ser profunda (análise de todas as interpretações possíveis e análise das relações gramaticais dentro de uma frase) ou superficial (análise restrita à identificação de estruturas e fenómenos linguísticos não recursivos que não são tratados, tais como problemas de ambiguidade).

Relativamente às componentes linguísticas dependentes do domínio, estas podem variar consoante os requisitos da aplicação. Neste grupo encontram-se as quatro tarefas principais da Extração de Informação, nomeadamente, Reconhecimento de Entidades Nomeadas (*Name Entity Recognition*), Resolução de Correferência (*Coreference Resolution*), Extração de Relações (*Relation Extraction*) e Extração de Eventos (*Event Extraction*) [35].

A tarefa de Reconhecimento de Entidades Nomeadas tem como objetivo reconhecer e classificar nomes encontrados num texto livre em tipos predefinidos, por exemplo, nomes de pessoas, de locais ou expressões numéricas [21]. As entidades a extrair dependem do domínio da Extração de Informação

sendo que no domínio da Biologia, as entidades são, normalmente, os nomes taxonómicos. Atualmente existem bases de dados que listam alguns desses nomes, nomeadamente, *Global Names Index* [3], *Catalogue of Life* [4], *Interim Register of Marine and Non-marine Genera* [5].

A tarefa de Resolução de Correferência diz respeito à identificação da presença da mesma entidade num texto com variações de nomes, ou seja, palavras diferentes que se referem à mesma entidade. A palavra pode estar presente como frase nominal, pronominal, nome ou implícito. Por fim, obtemos a identificação e classificação das relações entre entidades na tarefa de Extração de Relações [35].

Ao longo das várias tarefas é possível a aplicação de outros métodos para melhorar os resultados obtidos, tais como, padrões para identificar fragmentos de texto ou para descrever relações ou ainda para extrair atributos de modo a preencher os campos no modelo predefinido. Também podem ser aplicadas regras de inferência [27] para inferir conclusões válidas para preenchimento dos campos definidos [35].

2.2.2 Abordagens para Extração de Informação

Existem diversas abordagens para desenvolver um sistema de Extração de Informação. Segundo Krallinger, Erhardt e Valencia [26], para obtenção de dados no ramo da Biologia é comum seguir uma de quatro abordagens: baseada em dicionário, em regras, em algoritmos de Aprendizagem Automática ou então seguindo abordagens híbridas.

Abordagem baseada em dicionário

Utilizando uma lista de nomes relativos ao domínio em causa, realiza-se uma procura desses nomes no texto livre obtendo como resultado os nomes que se encontram em ambos. A principal desvantagem deste método é a necessidade de existir uma lista exaustiva dos nomes e termos a procurar, incluíndo erros ortográficos, variantes de nomes, abreviaturas, sinónimos e nomes ambíguos. No caso da área da Biologia é difícil manter listas exaustivas com toda a informação necessária devido, de entre outros factores, à constante atualização e descoberta de novos termos [46].

Abordagem baseada em regras

São definidas regras (por exemplo, Expressões Regulares [15]) constituídas por um padrão e uma ação a desenvolver. As regras são aplicadas ao texto livre e quando determinado padrão é identificado, a ação é realizada [23]. As regras podem ser elaboradas seguindo uma de duas metodologias: através da codificação manual ou da utilização de algoritmos de Aprendizagem Automática. Na primeira é necessário a existência de humanos peritos do domínio que definam as regras, ao contrário da segunda, em que a partir de exemplos estruturados já existentes, o sistema aprende as regras de extração [23].

[3] http://gni.globalnames.org/
[4] http://www.catalogueoflife.org/
[5] http://www.cmar.csiro.au/datacentre/irmng/

Neste método a principal desvantagem está presente nas regras criadas segundo o método de codificação manual, pois são tão específicas do domínio que raramente são aplicáveis a outros domínios. Por outro lado torna a construção do sistema bastante prolongado e pode excluir termos importantes que não correspondam exactamente aos padrões predefinidos [46].

Abordagem baseada em Aprendizagem Automática (*Machine Learning*)

A Aprendizagem Automática tem como objetivo o desenvolvimento de algoritmos para detetar automaticamente padrões em dados [41]. Podemos dividir os algoritmos de Aprendizagem Automática consoante o seu tipo de aprendizagem: aprendizagem supervisionada, semi-supervisionada ou não supervisionada [46].

A aprendizagem supervisionada é, normalmente, realizada através da indução de um modelo capaz de prever ocorrências futuras com base num grande conjunto de dados de treino [23]. Algoritmos que sigam uma aprendizagem supervisionada são difíceis de aplicar na área da Biologia devido à complexa tarefa de compilar um grande conjunto de dados de treino [46]. De forma a colmatar o elevado custo na preparação dos exemplos, surge a aprendizagem semi-supervisionada que se distingue da anterior no facto de necessitar de uma quantidade de exemplos bastante inferior [46].

Na aprendizagem não supervisionada não existe distinção entre dados de treino e dados de teste, sendo processados todos os dados de entrada com o objetivo de criar uma espécie de resumo ou aglomeração [41]. Normalmente, são algoritmos de agrupamento ou redução de dimensionalidade [46].

Abordagem híbrida

Cada domínio em que são aplicadas técnicas de Extração de Informação têm especificidades diferentes o que implica a necessidade de adaptar a solução. Deste modo, por vezes uma abordagem híbrida é utilizada para conjugar as vantagens das diferentes abordagens definidas anteriormente.

2.2.3 Aprendizagem Supervisionada para Classificação

A classificação tem como objetivo identificar a que conjunto de categorias uma nova observação pertence tendo em conta um conjunto de dados de treino que contém observações cuja a categoria é conhecida [45]. Para este fim, existem diversos classificadores. Seguidamente apresentamos alguns classificadores, nomeadamente: *Naïve Bayes*, *Suport Vector Machine*, K Vizinhos Mais Próximos, Regressão Logística, Árvore de Decisão e *Random Forest*.

Naïve Bayes

O classificador *Naïve Bayes* segue uma abordagem baseada no teorema de probabilidades de Bayes [41]. Para calcular a probabilidade de uma classe C_i dado um determinado conjunto de atributos $\vec{x}$ utiliza a Equação 2.1.

$$P(C_i|\vec{x}) = \frac{P(\vec{x}|C_i)P(C_i)}{P(\vec{x})} \tag{2.1}$$

Sendo que $P(C_i)$ é a probabilidade inicial da classe C_i e $P(\vec{x}|C_i)$ a probabilidade condicionada do $\vec{x}$ na classe C_i (Equação 2.2). A probabilidade de $\vec{x}$ é calculada segundo a Equação 2.3.

$$P(\vec{x}|C_i) = \prod P(x_j|C_i) \tag{2.2}$$

$$P(\vec{x}) = \sum_{i=1}^{M} P(\vec{x}|C_i)P(C_i) \tag{2.3}$$

Após calcular as probabilidades de todas as classes, a decisão é tomada verificando qual a classe que obtém maior probabilidade. Contudo, existem casos em que não se verifica um determinado atributo x_j numa classe C_i, deste modo a $P(x_j|C_i) = 0$ e, consequentemente, $P(C_i|\vec{x}) = 0$. Para que esta situação seja evitada utiliza-se o *Laplace smoothing* [51] somando um valor, normalmente o número um no numerador e no denominador como se pode observar na Equação 2.4 representado por k , em que N_{ij} é o número de vezes que x_j aparece na classe C_i, N_c é o número total de atributos na classe C_i e o n é o número de valores possíveis que o atributo pode apresentar [16].

$$P(x_j|C_i) = \frac{N_{ij} + k}{N_c + kn} \tag{2.4}$$

Este classificador tem a vantagem de ser facilmente calculável, rápido a implementar e a executar. Também não necessita de um grande volume de dados de treino pois converge facilmente para a resposta. Contudo torna-se uma abordagem ingénua porque assume a independência entre os atributos [42].

Suport Vector Machine **(SVM)**

O classificador utiliza modelos lineares para implementar limites não-lineares de classes em espaços com grande número de características [50]. Com o aumento da dimensionalidade do espaço de características, aumenta a complexidade da amostra e aumenta os desafios da complexidade computacional [41]. Deste modo, o classificador define zonas bem delimitadas traçando um hiperplano com o conhecimento adquirido nos dados de treino. Este plano permite maximizar os limites em relação às diversas classes existentes e permite classificar novos objetos com a maior precisão possível [41].

A utilização deste classificador tem a vantagem de ser eficaz em espaços de dimensões elevadas e em casos em que o número de dimensões é maior que o número de exemplos. Contudo, o classificador pode dar maus resultados perante a situação do número de características ser maior que o número de exemplos [8].

K Vizinhos Mais Próximos (*K Nearest Neighbors*)

O classificador K Vizinhos Mais Próximos utiliza um algoritmo simples que tem em consideração os dados de treino e prevê a classe de novos objetos tendo em conta as classes dos K vizinhos mais próximos do objeto em análise [41]. Ou seja, o objeto em análise é classificado consoante a maioria dos votos dos seus vizinhos sendo-lhe atribuido a mesma classe dos seus vizinhos [42].

Para definir a distância e perceber quais os vizinhos mais próximos, é utilizado a distância Euclidiana entre dois pontos. Sendo $X_1 = (x_{11}, x_{12}, ..., x_{1n})$ e $X_2 = (x_{21}, x_{22}, ..., x_{2n})$ a distância calcula-se segundo a equação 2.5 [16].

$$D(X1, X2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \tag{2.5}$$

A utilização deste classificador tem como vantagem ser bastante simples e funcionar bem em problemas básicos de reconhecimento. Contudo é bastante lento pois em cada predição calcula a distância a todos os vizinhos e a alteração do K também pode levar a uma alteração na classe resultante [16].

Regressão Logística

A Regressão Logística é um modelo linear para a classificação que prevê os valores tomados por uma classe utilizando a função logística representada na equação 2.6 onde $P(x) = P(Y = 1|x)$ ou $P(x) = P(Y = 0|x)$, tendo em atenção que a previsão é realizada tendo em conta problemas binários, assumindo valores de 0 ou 1 denominando o primeiro como probabilidade de fracasso e o segundo de probabilidade de sucesso [12].

$$log \frac{P(x)}{1 - P(x)} = \beta_0 + x\beta \tag{2.6}$$

O valor relativo a β deve ser estimado a partir dos dados de treino usando estimadores de Máxima Verossimilhança [6] que são métodos numéricos que realizam suposições sobre a distribuição dos dados [12]. Este classificador tem a vantagem de ser bastante rápido e funciona melhor quando tem de tomar decisões simples com pouca variância, sendo ideal na resolução de problemas binários [12].

Árvore de Decisão

A Árvore de Decisão divide os problemas numa sequência de decisões, que são modeladas como uma árvore. Num exemplo como o representado na figura 2.2, para se proceder à classificação de um objeto inicia-se o processo pelo nó de raiz que representa o atributo mais relevante da árvore. Posteriormente vai-se seguindo os nós que representam decisões que são tomadas consoante as características do objeto em classificação. Por fim chega-se a uma folha que representa a resposta do classificador. Cada árvore também contêm na sua constituição os arcos que unem nó a nó e nó a folha [41].

[6] `https://onlinecourses.science.psu.edu/stat414/node/191`

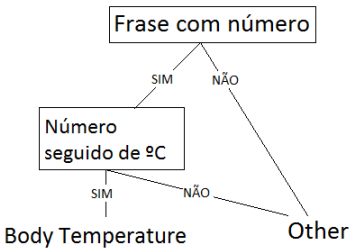

Figura 2.2: Exemplo de Árvore de Decisão.

As Árvores de Decisão têm a vantagem de serem fáceis de interpretar e de utilizar para proceder à classificação de um objeto, pois representam todas as possibilidades que esse objeto pode tomar, contudo, se houver uma alteração dos atributos é necessário reconstruir a árvore e facilmente chega a uma situação de *overfit* (um ajuste aos desvios causados por erros de medição, apresentando uma alta precisão, contudo o modelo não é uma boa representação da realidade) [41].

Random Forest

Para resolver a questão de *overfit* da Árvore de Decisão, surgiu algoritmos como o *Random Forest* que consiste na coleção de Árvores de Decisão, onde cada árvore é construída através de um algoritmo e de um conjunto de exemplos de treino obtidos dos dados de treino. A previsão é obtida pela maioria de votos sobre as previsões das árvores individuais [41]. O resultado obtido pelo *Random Forest* tende a ser o melhor do obtido pelos resultados das várias árvores utilizadas, como podemos verificar na figura 2.3 [7].

Random Forest tem a vantagem de ser eficiente em situações com muitos dados e, ao contrário da Árvore de Decisão, assegura que são utilizadas todas as variáveis [41].

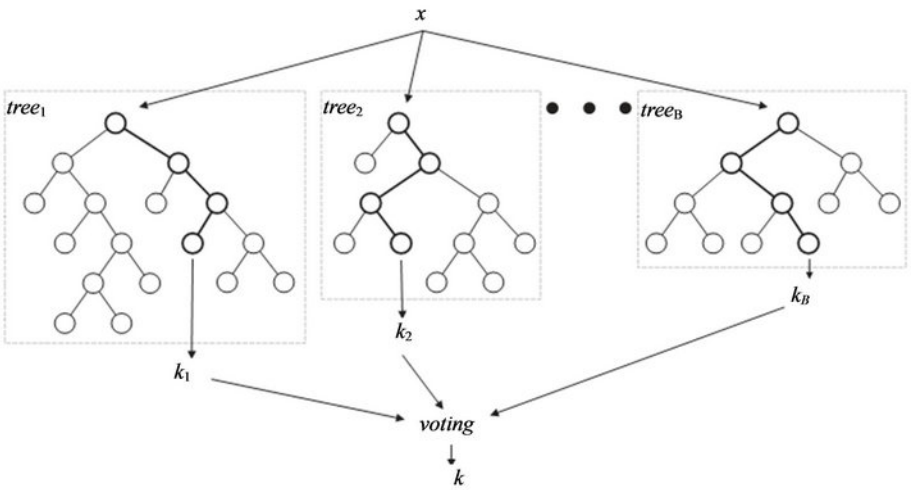

Figura 2.3: Exemplo Random Forest

[7] http://file.scirp.org/Html/6-9101686/f799e10c-50bd-48ec-9344-49d767083be5.jpg

2.3 Seleção de Características

Nos problemas de classificação, com a presença de um grande número de características utilizadas para caracterizar os dados que são processados pelos classificadores, pode resultar em situações de *overfit* tendo consequências para o seu desempenho. De forma a resolver este problema, têm sido criadas técnicas de redução de dimensionalidade [45].

A técnica de Seleção de Características tem como objetivo escolher um subconjunto dos recursos mais relevantes a partir do conjunto original de acordo com determinados critérios. Na figura 2.4 é apresentado uma estrutura para Seleção de Características segundo Tang [45], sendo que esta afeta principalmente a fase de treino do classificador. Primeiramente são selecionadas as características e, de seguida, são processados os dados com as características selecionadas pelo classificador [45].

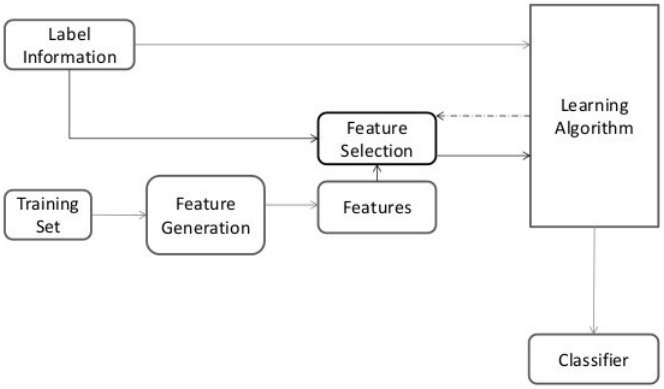

Figura 2.4: Seleção de Características

2.4 Povoamento de Bases de Conhecimento

A questão referente ao povoamento de Bases de Conhecimento (*Knowledge Base Population (KBP)*) [8] surgiu em 2009 com o intuíto de promover a investigação em sistemas automáticos que descobrissem informação sobre entidades a partir de um grande volume de documentos e, posteriormente, utilizassem esses dados para constituir Bases de Conhecimento [30]. Vários são os grupos que participam nesta competição que contempla várias subtarefas. Neste caso, interessa destacar a tarefa de *Slot Filling*.

Na tarefa de *Slot Filling* é necessário criar um sistema de Extração de Informação para obter determinados atributos de pessoas e organizações a partir de uma grande quantidade de documentos. Os participantes nesta tarefa têm acesso aos documentos de onde é necessário extrair a informação (normalmente dados provenientes de páginas de internet e de fóruns de discussão) e um modelo com uma lista de campos descrimidados (um exemplo pode ser visto na tabela 2.1) que têm que ser preenchidos. Assim sendo, o objetivo final é o povoamento dos vários campos da Base de Conhecimento utilizando a abordagem que melhor acharem se adequar ao problema [43].

[8] `http://www.nist.gov/tac/2015/KBP`

Tabela 2.1: Exemplos de campos sobre a entidade pessoa na tarefa *Slot Filling*

Name	Type	List?
per:alternative_names	Names	Yes
per:age	Value	No
per:date_of_birth	Value	No
per:religion	String	Yes

2.5 Métricas de avaliação

Para realizar a avaliação de desempenho dos sistemas de Extração de informação são utilizadas três métricas: *Accuracy*, *Precision*, *Recall* e *F-Measure* [11].

Tabela 2.2: Matriz de Confusão

	Verdade Positiva	Verdade Negativa
Previsão Positiva	VP	FP
Previsão Negativa	FN	VN

Tendo em consideração a matriz de confusão (Tabela 2.2) sendo VP o número de previsões positivas que estão corretas, FP o número de previsões positivas que estão incorretas, FN o número de previsões negativas que estão incorretas e VN o número de previsões negativas que estão corretas [11].

- **Accuracy** (equação 2.7): avalia a percentagem de todas as previsões corretas.

$$\textit{Accuracy} = \frac{VP + VN}{VP + FP + FN + VN} \tag{2.7}$$

- **Precision** (equação 2.8): Mede a proporção de campos extraídos que foram retornados corretamente comparando com todos os valores extraídos, quer sejam corretos, quer sejam incorretos [43].

$$\textit{Precision} = \frac{VP}{VP + FP} \tag{2.8}$$

- **Recall** (equação 2.9): Mede a proporção de campos corretos que foram retornados corretamente tendo em conta campos de referência esperados a serem preenchidos [43].

$$\textit{Recall} = \frac{VP}{VP + FN} \tag{2.9}$$

- **F-measure** (equação 2.10): Combina as duas métricas anteriores, *Precision* e *Recall*. Para que tenha uma *F-Measure* aceitável é necessário haver um equilíbrio entre as métricas envolvidas. O valor de β^2 é um valor não negativo para ajustar a sua ponderação relativa $(\beta^2 = 1.0)$ [35].

$$\textit{F-Measure} = \frac{(\beta^2 + 1) \times \textit{Precision} \times \textit{Recall}}{(\beta^2 \times \textit{Precision}) + \textit{Recall}} \tag{2.10}$$

2.6 Sumário

Neste capítulo apresentámos os principais conceitos relativos à criação e construção de Bases de Conhecimento. Apresentámos o conceito de *Text Mining* e especificámos a Extração de Informação, quer a nível de conceito, de arquitetura de Sistemas de Extração de Informação e as várias abordagens para desenvolver a sua função, relativamente à abordagem baseada em dicioánrio, regras, aprendizagem automática e abordagem híbrida. Expusemos alguns dos algoritmos de Aprendizagem Supervisionada para classificação, apresentámos o conceito de Seleção de Características e o desafio de povoamento de Bases de Conhecimento e, por fim, as métricas utilizadas para avaliar os sistemas desta natureza.

Capítulo 3

Trabalho Relacionado

Vários têm sido os sistemas implementados para Extração de Informação em texto. Neste trabalho desejamos obter informação a partir de artigos científicos, que embora sejam considerados documentos semi-estruturados, apresentam elementos de estrutura em comum: título, autores, resumo, palavras-chave e referências bibliográficas, a parte do corpo do texto tem uma estrutura bastante irregular e pode conter texto, imagens, tabelas e gráficos. Assim sendo, é necessário ter em conta a possibilidade de extrair informação de todos estes meios de organização de informação.

Relativamente aos artigos científicos no domínio da Biologia, podemos encontrar algumas características específicas que dificultam a Extração de Informação, nomeadamente:

- **Linguagem especializada:** A linguagem utilizada nos artigos de Biologia está em constante mudança devido às alterações do nosso entendimento sobre o ramo. São criados novos termos ou são removidos outros, por vezes, bastante ambíguos [19].

- **Diferença de Sintaxe:** Não existe uma sintaxe padrão nas descrições dos diferentes grupos taxonómicos, ou mesmo dentro do mesmo grupo taxonómico. Existem descrições escritas em inglês, noutras línguas e existem vários nomes e abreviaturas para a mesma entidade [46].

Segundo Thessen e Parr [47], para criar um sistema capaz de extrair associações de termos em informação relativa a grupos taxonómicos, o sistema tem que ter a capacidade de reconhecer os nomes taxonómicos no texto, de reconhecer vários termos usados para identificar a mesma espécie e ter a capacidade de determinar se a taxonomia mencionada refere-se a uma interação ecológica.

Portanto, foram implementados sistemas com o intuito de extração de nomes taxonómicos das espécies, extração de relações entre as entidades ou para o preenchimento de campos (*Slot Filling*) de forma a constituir uma Base de Conhecimento. Os sistemas podem ser implementados seguindo várias abordagens, nomeadamente, Dicionário, Regras ou Aprendizagem Automática. Contudo é de salientar que não são muito comuns sistemas a utilizar apenas uma das abordagens e por isso existem os sistemas que seguem uma abordagem híbrida onde conjugam mais que uma das abordagens referidas. De seguida apresentamos alguns sistemas das diferentes abordagens e sistemas com o objetivo de povoamento de Bases de Conhecimento.

3.1 Sistemas com recurso a dicionário

Os sistemas com recurso a dicionário estudados, revelaram ter como principal objetivo extrair e reconhecer nomes de entidades, que no caso da Biologia, são nomes taxonómicos.

Na área da Biologia, os nomes taxonómicos são frequentemente atualizados, o que converge para uma possibilidade de existir ambiguidade das palavras biológicas. Deste modo, o sistema *TaxonFinder* [1] identifica nomes científicos num texto através da comparação com várias listas de dados. As várias listas foram construídas manualmente por especialistas do domínio e, cada lista continha, respectivamente, nomes de espécies, nomes de géneros, nomes de famílias e palavras do léxico comum.

Primeiramente efetuaram a extração e análise das várias palavras através da aplicação de técnicas de Processamento de Língua Natural com o objetivo de obter as palavras começadas por maiúscula [46]. Depois de encontrado o primeiro candidato, o sistema verifica se este existe em alguma das várias listas. Se existir na lista de léxico comum é excluído e categorizado como não sendo um nome taxonómico, se por outro lado pertencer a uma das restantes listas então é um candidato a nome taxonómico.

Se o candidato existir na lista de nomes de géneros será necessário verificar se a palavra seguinte a esta no texto também se encontra escrita por letra maiúscula. Caso se verifique então vai procurando a próxima palavra até encontrar o polinómio final de todas as palavras seguidas começadas por maiúscula e categorizá-lo por nome de género. [46].

Como o sistema *TaxonFinder* se limita à utilização de dicionários, não irá identificar nomes recentes ou erros ortográficos, por outro lado, pode descobrir novas combinações de nomes desconhecidos devido à combinação dos vários polinómios diferentes que vai encontrando e adicionando às listas [46].

3.2 Sistemas com recurso a regras

As regras são normalmente utilizadas para melhorar a exatidão da Extração de Informação em sistemas com recurso a outras abordagens. Na literatura é possível encontrar dois sistemas que têm origem num sistema mais antigo, o *Large Scale Information Extraction (LASIE)* [24].

O sistema *Protein Active Site Template Acquisition(PASTA)* foi implementado com o objetivo de Extrair Informação sobre as funções dos aminoácidos em moléculas de proteínas, a partir de revistas científicas e artigos completos. O produto final seria a criação de uma Base de Conhecimento dos locais ativos das proteínas [21]. O sistema *Enzyme and Metabolic Pathways Information Extraction (EMPathIE)*, foi implementado para extrair detalhes de reações enzimáticas de artigos científicos provenientes de revistas científicas da área da Biologia e, ao contrário do *PASTA*, já continha uma base de dados construída manualmente com dados de enzimas [21].

Ambos os sistemas são constituídos pelas quatro etapas principais: 1) Pré-processamento de texto (análise da estrutura usando expressões regulares e divisão do texto em *tokens*); 2) Processamento lexical e terminologia (identificação de maiúsculas e minúsculas, listar os termos de componentes de diver-

[1] `http://taxonfinder.org/`

sas categorias, uso de conhecimento morfológico sobre sufixos padrões em bioquímica e a base de dados construída manualmente no caso do sistema *EMPathIE*); 3) Análise e interpretação semântica (são utilizadas regras para deteção de limitadores frásicos, análise gramatical (*part of speech*) e interpretação semântica); e 4) Interpretação do discurso (utilização de mecanismos de resolução de correferência para verificar duplicados e regras de inferência para preencher os campos pretendidos da Base de Conhecimento [21]).

3.3 Sistemas com recurso a Aprendizagem Automática

Os sistemas que utilizam apenas recursos de Aprendizagem Automática não são muitos, contudo o sistema *NetiNeti (Name Extraction from Textual Information-Name Extraction for Taxonomic Indexing)* é baseado em Aprendizagem Automática com o intuito de reconhecer e descobrir nomes científicos em texto tendo em consideração erros de Reconhecimento Ótico de Caracteres, erros ortográficos e variações dos nomes [1].

Inicialmente procede às tarefas de análise linguística independente do domínio, nomeadamente, análise morfológica e sintática com recurso a técnicas de Processamento de Língua Natural. De seguida obtiveram os nomes candidatos para utilizar na classificação. Tendo como objetivo verificar se os nomes candidatos se referiam a nomes científicos ou não, utilizaram o classificador de Aprendizagem Automática supervisionado, o *Naïve Bayes* e Entropia Máxima [16].

Para proceder à classificação foram utilizados como exemplos de treino frases com nomes científicos e frases com nomes não científicos anotados obtidos em resumos de artigos provenientes de *MEDLINE* [2] e da *Encyclopedia of Life*. Deste modo obtiveram uma classificação para os nomes candidatos segundo os exemplos aprendidos pelo classificador.

Por fim, para melhorar os resultados, recorreram a regras e métodos de filtragem. Comparando os resultados do sistema *NetiNeti* com um dicionário com nomes científicos resultante da anotação manual apartir de BHL [3], os autores perceberam que este sistema obteve melhores resultados e que é uma ferramenta bastante útil para saber a quantidade total de nomes científicos abordados num texto ou para extrair frases associadas a esses nomes científicos [1].

3.4 Sistemas com abordagens híbridas

A abordagem híbrida é a mais escolhida para implementação de sistemas de Extração de Informação devido à possibilidade de conjugar as vantagens de diversas abordagens.

TaxonGrab é um exemplo de um sistema criado para identificar nomes taxonómicos usando uma combinação de uma lista de termos em inglês não taxonómicos (o dicionário) e regras de nomenclatura binomial de Lineu [28]. Acreditando que a maior parte dos nomes taxonómicos não são utilizados na linguagem comum, a abordagem entende que se uma determinada palavra não existe no

[2] `https://www.nlm.nih.gov/bsd/pmresources.html`
[3] `http://www.biodiversitylibrary.org`

dicionário é porque pode ser um nome taxonómico [25]. Para confirmar se os termos extraídos são nomes taxonómicos, estes são comparados com regras criadas a partir da nomenclatura binomial de Lineu. Este sistema não é muito preciso, pois não contempla os problemas de erros ortográficos e as palavras noutra língua que não seja inglês, contudo, tem a vantagem de não necessitar de uma lista completa de nomes taxonómicos [25].

Considerando abordagens com recurso a dicionário para a extração de conceitos biológicos que são bastante simples e não têm em conta as variações dos termos (mudanças morfológicas, sintáticas ou semânticas), o sistema *MaxMatcher* apresenta uma abordagem aproximada para colmatar a problemática das variações dos termos. Deste modo, propõe não só a extração dos conceitos, mas também das palavras significativas que o constituem. Por exemplo, sendo o conceito *"proteína gyrB"*, o sistema faz a procura por *"proteína gyrB"* e também por *"gyrB"* [55]. Neste sistema utilizam como dicionário uma lista que contém milhões de conceitos sobre biomedicina e saúde, os seus sinónimos e as suas relações e, usam-no para calcular uma pontuação para cada palavra em relação aos conceitos biológicos através de uma matriz com palavras e conceitos. Para fazer a Extração de Informação usam algumas regras para identificar o conceito candidato. As regras podem ser, por exemplo, considerar que um conceito biológico poderá começar por substantivo, número ou adjetivo e não pode conter símbolos de pontuação (exceto hífen e aspas), verbos, conjunções e preposições (exceto *de*) [55]. Ao contrário do *TaxonGrab* este sistema tem a vantagem de identificar um conceito que sofra alterações na composição do seu nome [55].

O sistema *ProMiner* é outro sistema que conjuga o uso de dicionário e regras, tendo como objetivo identificar nomes de conceitos na área da biomedicina (proteínas e genes) [17]. O sistema consiste em três fases: geração de dicionário, deteção de ocorrências e filtragem.

Na primeira fase, o dicionário é gerado a partir de bases de dados de proteínas e genes contendo conceitos, significados, descrições físicas, nomes de famílias e outras anotações. De seguida, é realizada a deteção de ocorrências no texto percorrendo as várias palavras e comparando com as ocorrências no dicionário. Por fim são utilizadas expressões regulares para aperfeiçoar a procura, assumido que certas palavras que ocorrem com grande frequência não são candidatas a nomes de proteínas. Na fase final, são utilizados filtros para que os resultados sejam mais específicos para determinada pesquisa, nomeadamente filtros para resolver ambiguidades, isto porque mais que uma entidade pode compartilhar características ou siglas e, são aplicados filtros de organismos (baseados em *NCBI Taxonomy* [4]) que procura fazer correspondência com nomes de organismos de modo a aceitar ou rejeitar as ocorrências.

Este sistema tem a vantagem de ser facilmente adaptável a novas configurações e domínios, ao contrário dos sistemas baseados apenas em Aprendizagem Automática que necessitam de um conjunto abrangente de dados para constituir os exemplos de treino para obter bons resultados [17].

Para colmatar a necessidade da grande quantidade de exemplos de treino nas abordagens de Aprendizagem Automática, Craven e Kumlien [10] apresentam uma abordagem que tem como ponto de partida um conjunto de classes de interesse, as suas relações e um grupo de documentos a serem

[4] `http://www.ncbi.nlm.nih.gov/taxonomy`

processados. O objetivo do sistema é extrair informação dos documentos que preencha as classes e as relações pretendidas.

Um exemplo de relação a retirar é $localizaçãoSubcelular(Proteína, EstruturaSubcelular)$ onde são obtidas proteínas e as estruturas subcelulares onde se encontram. Assume-se uma relação binária do tipo $r(X, Y)$ sendo que para cada uma das variáveis existe um léxico semântico de palavras possíveis que podem ser usadas no caso de r. Por exemplo, o léxico de semântica de $EstruturaSubcelular$ contém as palavras núcleo e mitocôndrias.

A primeira fase da abordagem é identificar instâncias que possam expressar a relação r, para tal, verifica se a frase em análise contém as duas palavras (X e Y) e se é considerada um exemplo positivo. Caso não se verifique, essa frase será considerada um exemplo negativo. É utilizado o classificador *Naïve Bayes* considerando que a posição de cada palavra nas frases não interessa, ou seja, encontrar a palavra proteína no início da frase é a mesma coisa que no final e assume também que a ocorrência de uma palavra dum documento é independente de todas as outras palavras no mesmo documento [10].

Esta abordagem apresenta a limitação de obter os dados de treino de forma morosa, pois são obtidos manualmente através de um especialista que determina as palavras que correspondem às diversas classes. Para tal, os autores recorreram a bases de dados existentes para obter os dados de treino, nomeadamente, a partir de páginas de internet e resumos de artigos a partir do *PubMed* [5].

Perante a solução apresentada os autores identificaram uma nova limitação na fase de obtenção de candidatos onde, apenas as palavras eram consideradas ignorando a relação entre elas. Assim sendo, propuseram analisar a estrutura gramatical utilizando regras. Com esta alteração os resultados melhoraram bastante [10].

Outro sistema híbrido foi proposto por Abacha e Zweigenbaum [4] com o intuito de extração de relações entre doenças e tratamentos. Esta abordagem conjuga a utilização de regras e de Aprendizagem Automática. Primeiramente foram definidas regras por peritos definindo padrões lexicais obtidos manualmente a partir de artigos de *MEDLINE*, sendo referentes a sequências de palavras, marcação semântica e limitadores de frases. De seguida, foram definidas hierarquias para os vários padrões de modo a diferenciar padrões mais gerais dos mais específicos. Deste modo, permitia escolher padrões com menor nível de especificidade para obter mais candidatos.

Numa segunda fase, utilizaram um algoritmo de Aprendizagem Automática supervisionada, o classificador *Support Vector Machine* para classificar os candidatos. Dado um conjunto de exemplo de treino e as categorias predefinidas, o classificador decidiu para que categoria cada candidato tinha maior probabilidade de pertencer [4].

Um outro sistema foi implementado de modo a dar resposta à necessidade de ter uma base de conhecimento na área da biomedicina sobre a compreensão de processos celulares, doenças humanas e resposta à cura, esse sistema foi implementado por Mallory *et al.* com base na *framework Deepdive* [6] (apresentada na secção 3.5). O objetivo do sistema era constituir uma base de conhecimento com dados de interações entre proteínas e fatores de transição a partir de artigos científicos [29].

Primeiramente procederam à análise do texto através de Processamento de Língua Natural dos

[5] `http://www.ncbi.nlm.nih.gov/pubmed`
[6] `http://deepdive.stanford.edu`

documentos utilizando o *Stanford CoreNLP 1.3.4* [7] [29]. Assim, cada documento foi dividido em frases e cada frase dividida em palavras (*tokens*). Desta análise surgem *tokens* rotulados e grafos de dependências entre os *tokens*. Para extração de relações definiram um extrator gene-gene que perante uma frase devolvia as relações candidatas. Uma relação candidata contém duas componentes: ocorrência de nome de dois genes e as características que os interligam. Deste modo, para identificar o nome dos genes utilizaram um dicionário (constituído por dados do domínio) e para identificar as características que os interligam, definiram características padrão possíveis para as palavras que pudessem aparecer entre os dois nomes de genes.

Para rotular as relações candidatas foram utilizados como exemplos de treino dados obtidos por *Distant Supervision* [31], rotulando como verdadeiro, falso ou desconhecido a relação candidata se a interação fosse conhecida por uma fonte independente. Neste caso definiram a variável $is\_correct$ que podia ter os valores: verdadeiro, se a relação ocorresse em bases de dados com interações de proteínas; falso, se estivesse numa base de dados com casos de não interações de proteínas; ou desconhecido, se não estivesse em nenhuma das duas bases de dados. Estas marcações foram utilizadas para criar o grafo de factos e, através de regras de inferência a *framework* calculava automaticamente as probabilidades de serem relação entre genes ou não. Por fim o resultado final foi a constituição da base de conhecimento.

Este sistema foi o primeiro que aplicou a *framework DeepDive* no domínio biomédico e foi possível extrair cerca de 12.390 relações gene-gene em 100.000 documentos [29].

3.5 Sistemas de povoamento de Base de Conhecimento

Desde 2009 que vários grupos de investigadores participam na subtarefa de *Slot Filling* da competição de povoamento de Bases de Conhecimento. Como tal, decidimos destacar os sistemas que obtiveram melhores resultados em 2014. Segundo Surdeanu e Ji [44], os melhores sistemas tiveram 52% de intervenção humana e a melhor estratégia para Extração de Informação baseou-se em *Distant Supervision* para a realização do treino do sistema.

O sistema que apresentou os melhores resultados for proposto por Angeli *et al.* da Universidade de *Stanford* [3]. Este sistema teve como base uma *framework* desenvolvida pelos autores: *DeepDive*. Esta *framework* foi implementada para auxiliar na construção de sistemas com o objetivo de tornar mais simples a integração do conhecimento de um determinado domínio sem a preocupação da programação de todo o processo.

Com efeito, o sistema recebe como dados de entrada os dados não estruturados e, são executadas três grandes fases no processo de Extração de Informação: 1) extração de características; 2) engenharia probabilística; e 3) inferência e aprendizagem.

Na primeira fase os dados são processados recorrendo a técnicas de Processamento de Língua Natural usando dois analisadores: *Stanford CoreNLP* [8] e *Malt parser* [9], obtendo no final os candidatos

[7] http://www.stanfordnlp.github.io/CoreNLP/
[8] http://stanfordnlp.github.io/CoreNLP/
[9] http://www.maltparser.org/

das relações mencionadas e a frase de onde foi retirada a relação, constituíndo uma base de dados relacional (denominada *evidence schema*).

Seguidamente é construído um grafo que contém todos os factos extraídos, nomeadamente as relações obtidas. De modo a verificar a probabilidade das relações obtidas serem verdadeiras, são criados dicionários através de bases de dados externas do domónio, usando *Distant Supervision* [10], que constituem os exemplos positivos do modelo.

O modelo calcula a probabilidade das relações serem verdadeiras de forma automática e produz a base de dados de saída juntamente com todos os valores das diferentes probabilidades calculadas. Por fim, constitui a Base de Conhecimento com as relações verdadeiras.

O sistema classificado em segundo lugar foi proposto por Hong *et al.*, o sistema *RPI_Blender* [20]. O sistema continha três etapas distintas: 1) recuperar documentos pertinentes sobre as entidades; 2) detetar os documentos relevantes que estão na origem do preenchimento dos campos;e, 3) extrair a informação para preenchimento dos campos.

Na primeira fase construíram um motor de busca local de entidades que utilizando regras de correspondência procurava as entidades nos documentos. Era necessário que, se estivessem a procurar por uma frase com várias palavras, que no documento estes aparecessem em conjunto ou perto umas das outras (na mesma frase). Só assim o documento cumpriria as regras e então seria selecionado como relevante.

Na segunda fase, para verificarem a coexistência de palavras nos documentos relevantes, utilizaram expressões lógicas e palavras-chave. As expressões lógicas foram utilizadas para resolver a possibilidade de existirem várias alternativas de nomes para as entidades. Assim usaram dicionários de nomes provenientes da *Wikipedia* [11] e da *DBpedia* [12] e as expressões que continham os nomes originais e as possíveis variantes. Para definir as palavras-chave relevantes, usaram a medida de *Term Frequency–Inverse Document Frequency (TF-IDF)* [36] que contabiliza o número de ocorrência das palavras e escolheram as cinco palavras com maior valor de ocorrência [20].

Na última fase propuseram a utilização de *Temporality-Based Clustering Model (TBCM)* que percorre os documentos relevantes e usa-os para expandir o conjunto de candidatos. O processo é iterativo até que a similaridade média dos documentos pertinentes recém constituídos seja inferior a um limite predefinido [20].

O sistema seguinte foi apresentado por Bentor *et al.*, o sistema *BLP-TI* [6]. Este sistema reutiliza um outro sistema denominado por *RelationFactory* [38]. O *RelationFactory* é um sistema constituído por uma sequência com dois estados: 1) geração de candidatos, onde são escolhidos os documentos mais pertinentes e realiza a recolha de relações com base no tipo de entidade a extrair; 2) fase de validação do candidato onde utiliza o classificador *Support Vector Machine* para determinar se as frases candidatas expressam uma relação válida para a consulta realizada [38].

No sistema *BLP-TI* é possível definir cinco fases distintas: 1) Extração de relações; 2) Aprendizagem da estrutura das regras; 3) Aprendizagem do peso das regras; 4) inferência; e 5) pós-processamento.

[10] `https://www.nlm.nih.gov/bsd/pmresources.html`
[11] `https://en.wikipedia.org`
[12] `http://wiki.dbpedia.org`

Na primeira fase utilizaram o *RelationFactory* para obter as relações ocorridas no texto. Na fase seguinte criaram um grafo a partir das várias relações obtidas onde cada nó representava as instâncias das relações. Posteriormente o grafo foi processado pelo *Bayesian Logic Programs*, sistema constituído por regras Bayesianas e cláusulas de primeira ordem de *Horn* [13]. Na terceira fase, para atribuir pesos para as regras aprendidas utilizaram os estimadores de Máxima Verossimilhança. De seguida utilizaram métodos com recurso a redes bayesianas [6] para preencher os campos (slots).

Por fim, na última fase combinam instâncias das relações extraídas e inferidas para a submissão na competição dando prioridade à instância com maior probabilidade.

3.6 Sumário

Neste capítulo foram apresentados os sistemas criados para proceder à Extração de Informação separados pelas diversas abordagens, nomeadamente, utilizando dicionário, regras, algoritmos de Aprendizagem Automática e abordagens híbridas. Por fim apresentamos os três primeiros classificados na competição de povoamento de Bases de Conhecimento realizado em 2014 na subtarefa *Slot Filling*.

Capítulo 4

Sistema de Extração: Implementação

Neste capítulo é apresentado o Sistema de Extração realizado para a obtenção dos dados das aves. Começamos por apresentar a arquitetura do sistema no geral e, de seguida exploramos os diversos módulos que o constitui. Por fim explicamos o método utilizado para a seleção de características usadas nos classificadores.

4.1 Arquitetura

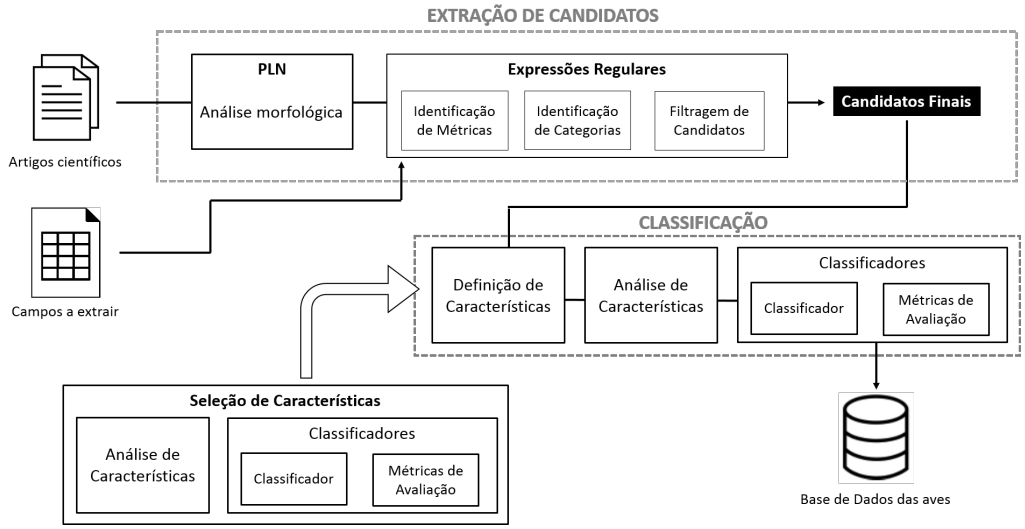

Figura 4.1: Arquitetura geral do Sistema de Extração

O nosso sistema segue uma abordagem híbrida complementando a abordagem segundo regras com a abordagem segundo Aprendizagem Automática. Assim sendo, para a análise dos textos são utilizadas técnicas de Processamento de Língua Natural, Expressões Regulares e vários classificadores.

Na figura 4.1 é possível verificar a arquitetura geral do nosso sistema, onde apresentamos os documentos de entrada no sistema, o sistema e o produto final.

Conhecimento prévio e documentos de entrada

Antes de construírmos o nosso sistema tivemos conhecimento de quais os dados necessários a obter no final para constituir a Base de Conhecimento. Assim sendo, através da análise de artigos científicos da área de Biologia era necessário extrair dados das aves de determinadas características que definimos como categorias neste sistema. Na Tabela 4.1 são apresentadas as categorias para quais o sistema está construído e as palavras que sabiamos à partida estarem associadas a cada categoria. Como sabiamos que as categorias pretendidas referiam-se a valores numéricos, decidimos envergar pela solução apresentada.

Tabela 4.1: Categorias e palavras relacionadas

Categoria	Palavras relacionadas
Body Mass	wet weight; dry weight; wet mass; dry mass; at birth; hatching; hatchling; at fledging; fledgling; adult; grams; kilograms
Body Temperature	chick; adult; body; temperature; Celsius
Egg Temperature	incubation; egg; temperature; Celsius
Fledging	fledging; leaves the nest; days
Incubation	incubation; hatching; days
Total Body Water	total; body; water; content; percentage

Relativamente ao sistema, os artigos científicos e as categorias para extrair a informação constituem os dados de entrada e, podemos destacar no nosso sistema três módulos: a extração de candidatos, classificação e seleção de características. Seguidamente iremos explicar cada um dos módulos relativamente ao seu objetivo, seu desenvolvimento e seus produtos.

4.1.1 Extração de Candidatos

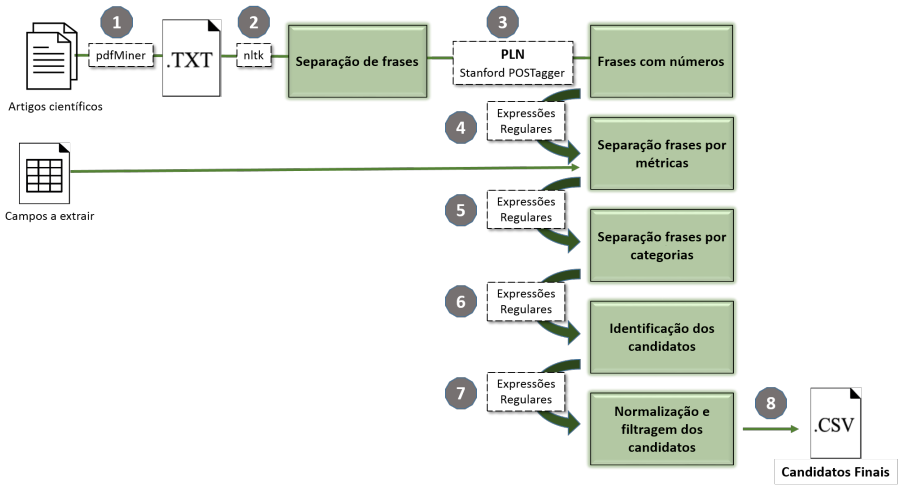

Figura 4.2: Módulo de Extração de candidatos

Os artigos científicos são documentos estruturados com campos em comum, sendo que a parte do corpo do documento, por vezes, apresenta o texto em várias colunas paralelas. Utilizámos a ferramenta *pdfMiner* [1] que nos permitiu obter o texto que guardámos em documentos em formato *.txt* para futura manipulação (1).

De seguida para obtermos a separação do texto em frases, utilizámos a biblioteca *Natural Language Toolkit (nltk)* [2] que nos permitiu separar as frases pelos elementos de pontuação de fim de frase. Contudo, devido à estrutura em colunas dos documentos, existiam algumas palavras que estavam separadas por um hífen. Para colmatar este problema analisámos todas as frases e eliminámos esse hífen para que a palavra se unisse (2).

Como era do nosso conhecimento que os dados a extrair eram apenas números, obtidas todas as frases dos textos, de seguida filtrámos as frases para considerarmos apenas as que continham números através da análise *Part-of-Speech* inserida na ferramenta *Stanford CoreNLP* [3] de modo a percorrer as frases e selecionar apenas as que continham números, quer no formato em dígitos quer por extenso (3).

Nas seguintes fases foram utilizadas Expressões Regulares para realizar a filtragem até obtermos os candidatos para cada uma das categorias. Deste modo, começamos por procurar as frases que continham na sua constituição determinadas grandezas que tinham sido definidas a posteriori (4) consoante as várias categorias. Como sabiamos que existiam determinadas palavras que poderiam estar presente nas frases que indicassem que seria de determinada categoria, de seguida, percorremos as várias frases obtidas na fase anterior verificando se continham as palavras referentes à categoria em análise (5). Se se confirmassem então seriam guardadas como possíveis candidatas a ter dados daquela categoria em concreto.

Tendo já as frases filtradas pelas diversas categorias, só faltava exrair os valores a analisar. Deste modo, com a utilização de expressões regulares, obtivemos os valores presentes nas frases (6). Estes valores poderiam ser apresentados como sendo um valor singular e a unidade de grandeza ou poderia ser um intervalo de valores em que a unidade de grandeza poderia estar no fim do intervalo ou estar associado a cada um dos valores. Obtidos os valores foi necessário transformar os números que se apresentavam por extenso em dígitos e realizar uma normalização como apresentamos na tabela 4.2. Valores com grandezas diferentes foram convertidos à unidade de grandeza e os valores apresentados em intervalos foram convertidos consoante o valor de cada número.

Tabela 4.2: Regras de Normalização

Caso	Normalização	Exemplo
Número maior, Número menor	Soma	20 e 3 = 23
Número menor, Número maior	Média	5 e 15 = 10

Como produto final deste módulo, apresentamos ficheiros em formato *.CSV* onde são guardados por cada categoria os candidatos na forma de: valor normalizado, o valor original e a frase de onde tem origem, como apresentamos na tabela 4.3.

[1] `https://pypi.python.org/pypi/pdfminer/`
[2] `http://www.nltk.org`
[3] `http://stanfordnlp.github.io/CoreNLP/`

Tabela 4.3: Exemplo de linha no ficheiro .CSV

Valor normalizado	Valor	Frase
358.78	356.7 + 2.08 g	”The Red Breasted Toucan was represented by two individuals, a male, whose mass was 356.7 + 2.08 g (n = 22)”

Referindo classe como sendo a categoria ao qual o candidato pertence e, como no módulo seguinte queriamos descobrir se os candidatos que identificámos faziam realmente parte da categoria identificada, foi necessário complementar cada uma da linha de entrada do documento gerado com a classe *“OTHER”*, como na tabela 4.4. No caso do nosso sistema a classe só podia ter dois valores, *”OTHER”* ou o nome de cada categoria.

Tabela 4.4: Exemplo de candidato

Valor normalizado	Valor	Frase	Classe
4.0	4 days	”Care should be taken to avoid removing the egg more than 4 days prior to hatching.”	OTHER

Os exemplos positivos que nos foram fornecidos foram classificados com a classe da categoria correspondente, como se pode verificar no exemplo presente na tabela 4.5. Contudo, devido à limitação na quantidade de exemplos positivos de cada classe, decidimos testar duas abordagens distintas: juntar todas as categorias num único sistema; criar vários sistemas sendo que cada um referente a cada categoria. No último os exemplos a passar para o classificador teriam que ter os candidatos positivos e uma quantidade de candidatos que seriam respetivamente uma proporção de um terço a mais dos exemplos positivos que detínhamos. Deste modo, para cada categoria obtivemos o total de ficheiros *.CSV* quantos era necessário de modo a repartir todos os exemplos a testar de acordo com a proporção estabelecida.

Tabela 4.5: Exemplo positivo da classe Incubation

Valor normalizado	Valor	Frase	Classe
42.2	42.2 days	”The incubation phrase ranges from 4 to 45 days for golden eagles, with an average of 42.4 days”	INCUBATION

4.1.2 Classificação

O módulo de classificação teve como objetivo a classificação dos candidatos de cada categoria sendo que referia a qual da categorias pertencia. Para tal, primeiramente definimos as características a analisar em cada candidato. As áreas das características são as seguintes:

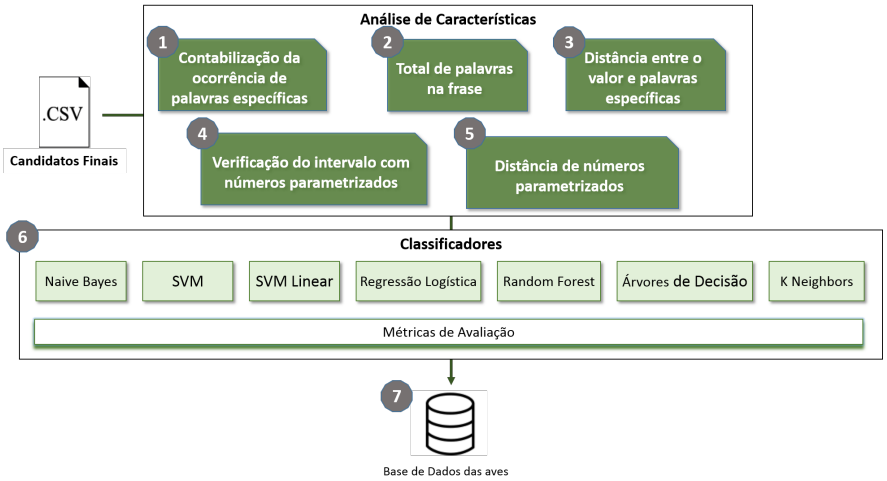

Figura 4.3: Módulo de Classificação

- **Contabilização da ocorrência de palavras nas frases:** Recorrendo à ferramenta *scikit-learn* [4], identificou todas as palavras nas frases dos exemplos todos, excluíndo as palavras que são apenas conectores frásicos e, contabilizou o número de ocorrência nas frases dos candidatos;
- **Total de palavras na frase:** Contabilizou o tamanho da frase relativamente ao número de palavras;
- **Distância entre o valor e palavras específicas:** Sabendo as palavras que podem existir nas frases dos candidatos, identificou quais as palavras presente e contabilizou o número de palavras que as distanciam do valor.
- **Verificação do intervalo com números parametrizados:** Sabendo os valores relativos à média e mediana das várias categorias, verificámos se o valor em análise estava perto desses valores;
- **Distância de números parametrizados:** Contabilizámos a distância numérica do valor em análise comparativamente aos valores de média e mediana

Depois de obtida a análise das características dos diversos candidatos tivemos que constituir o grupo de exemplos de treino e de teste. Como não tinhamos muitos exemplos positivos de cada categoria decidimos seguir a abordagem *Leave One Out* [5] na medida em que se realiza tantas iterações quantos os candidatos em teste. Ou seja, todos os candidatos são vistos como um exemplo de teste e os restantes estão no grupo dos exemplos de treino.

Utilizámos vários classificadores para verificarmos os que obtinham melhores resultados, nomeadamente o *Naïve Bayes*, *Support Vector Machine* (SVM), SVM Linear, Regressão Logística, *Random Forest*, Árvore de Decisão e K Vizinhos Mais Próximos. Para realizarmos a avaliação dos resultados utilizámos as métricas de avaliação utilizadas para analisar os sistemas de Extração de Informação, nomeadamente, *Accuracy, Precision, Recall* e *F1-Score*.

Por fim, obtivemos uma base de dados com as diversas categorias e os valores que o nosso sistema identificou como possíveis valores relativos às aves como no exemplo apresentado na tabela 4.6.

[4] http://scikit-learn.org
[5] http://scikit-learn.org/stable/modules/cross_validation.html

Tabela 4.6: Exemplo de Base de Conhecimento

Body Mass	Body Temperature	Egg Temperature	Fledging	Incubation	Total Body Water
1000 g	41.2 C	35.5 C	34 days	3 days	73%
305 g	49.09 C	34.15 C	37 days	8 days	68%
867 g	36.3 C	37 C	50 days	78 days	6%

4.2 Seleção de Características

Quando definimos as características, não sabíamos qual o seu impacto na classificação, deste modo, para otimizar a utilização das várias características e analisar quais as que melhor se adequam e permitem melhores resultados, realizámos o módulo de Seleção de Características seguindo o algoritmo de *Sequencial Feature Selection* [48]. Segundo este algoritmo (ver figura 4.7), o sistema vai percorrendo as diversas características e vai contabilizando uma métrica de avaliação definida e, consoante esses resultados, vai criando uma sequência das características que obtém um valor superior. No nosso caso, como não tínhamos muitos exemplos positivos das várias categorias, por vezes os resultados eram enviesados para uma das duas classes, o que resultava em valores de *accuracy* bastante altos e não estávamos a conseguir descobrir o pretendido. Assim sendo, definimos como métrica de avaliação *F1-Score*. Deste modo, o sistema percorreu todas as características e foi calculando o valor de *F1-Score* para a sequência e por fim obteve a sequência que permitia o melhor valor para a métrica utilizada.

Tabela 4.7: Pseudo-código do algoritmo *Sequencial Feature Selection*

```
listaCaracteristicas
F1-ScoreMaximo = 0
SequenciaFinal = []
se sequenciaFinal vazia:
  Por cada caracteristica em listaCaracteristicas:
    Analisa os candidatos com a caracteristica
    Executa o classificador com os candidatos
    Guarda lista com os vários F1-Score e com característica associada
    Verifica o F1-Score maior da lista de F1-Score
    F1-ScoreMaximo = F1-Score maior
    Adicionar Caracteristica correspondente ao F1-Score maior na SequenciaFinal
se não:
  Por cada caracteristica em listaCaracteristicas:
    Analisa os candidatos com a sequenciaFinal mais caracteristica
    Executa o Classificador com os candidatos
    Guarda lista com os vários F1-Score e com característica associada
    Verifica o maior F1-Score da lista de F1-Score
    Se F1-Score maior que F1-ScoreMaximo
      F1-ScoreMaximo = F1-Score
      Adicionar Caracteristica correspondente ao maior F1-Score na SequenciaFinal
```

4.3 Sumário

Neste capítulo apresentámos a estrutura do nosso sistema de Extração de Informação, apresentando os diversos módulos que a constituem, nomeadamente, Extração de Candidatos, Classificação e Seleção de Características.

Capítulo 5

Resultados

No presente capítulo apresentamos os resultados obtidos da execução do nosso sistema. Começamos por mostrar alguns exemplos que dificultaram a extração de informação, a extração de alguns valores corretos e, por fim, apresentamos os resultados das métricas de avaliação.

5.1 Processo de Extração

Os dados de entrada do nosso sistema foram 45 artigos científicos em formato *.pdf* que foram transformados de modo a obter o texto em formato que pudesse ser manipulado. Contudo, tendo em conta algumas limitações da ferramenta *pdfMiner* obtivemos algumas interpretações dos caracteres que não correspondiam ao que se encontrava no documento. Na figura 5.1 encontra-se um exemplo de texto presente nos *.pdf* e na tabela 5.1 o que resultou da sua transformação. Como é possível observar, o símbolo de graus (^oC) é representado de várias formas, existem valores que originalmente seriam valores décimais passaram a ser valores inteiros e, por fim, o número 0 passou a ser a letra *O*.

ENVIRONMENTAL CONDITIONS

Data from the navy meteorological station on Midway Island show that the macroclimate is remarkably equable. In January and February, the mean low is 15°C. and the mean high is 21°C. (extremes, 12.0° and 24.5°C.). In June and July, the mean low is 21.0°C. and the mean high is 28.5°C. (extremes, 18.5° and 31.0°C.). However, these temperatures were taken at a height of 21 meters above ground; the microclimates to which the nesting birds are exposed are more variable and are discussed in the accounts for each species.

Figura 5.1: Transformação do texto

Tabela 5.1: Transformação

ENVIRONMENTAL CONDITIONS Data from the navy meteorological station on Midway Island show that the macro climate is remarkably equable. In January and February, the mean low is 15'C. and the mean high is 21C. (extremes, 12.0' and 245C.). In June and July, the mean low is 21.O"C. and the mean high is 28.5'C. (extremes, 18.5' and 31.O'C.). However, these temperatures were taken at a height of 2 1 meters above ground; the microclimates to which the nesting birds are exposed are more variable and are discussed in the ac counts for each species.

De seguida, para obter os candidatos possíveis para cada categoria, realizámos várias tarefas de filtragem utilizando Expressões Regulares. Estas tarefas contemplaram a obtenção de frases com números, unidades de grandeza e palavras predefinidas. Durante este processo percebemos que, devido à forma como o texto se encontrava formatado, o sistema não conseguia distinguir algumas situações e que, desse modo, esses candidatos tiveram que ser excluídos e tidos em atenção. Na tabela 5.2 podemos verificar alguns dos casos em que essa exclusão foi necessária.

Tabela 5.2: Candidatos excluidos

Candidatos	Frase Origem	Motivo
19 kg 30 kg	”As a guide only, the important weights of growing Emus are as follows: ï‚§ Hatch ï‚§ Three months ï‚§ Six months ï‚§ 1 year ï‚§ 2 years 420 g 8 kg 19 kg 30 kg 50 kg (Kent 2008) âœEmus can achieve mature body weights of 49.5 kg, grow at an average growth rate of 68.4 g/d from birth to maturity, and reach maximum velocity of growth at 105 d of age when they weigh 9.8 kgâ (Dunk et al.”	Informação obtida através de uma tabela e não é possível identificar se se refere ao pretendido. Como os dois valores estão seguidos, a Expressão Regular que o identifica apenas percebe que podem estar relacionados.
20/100 g	”Total body water was 61% of body mass (ml H20/100 g BM) in both hydrated and dehy drated emus.”	Não consegue diferenciar que o 20 não seria um número a considerar, apenas os 100g.
3 02 g	”At ambient tem peratures below 17Ã¸C, thermal conductance equaled 0.069 + 0.0044 cm 3 02 g (cid:127) h (cid:127) Ã¸C (cid:127) (n = 5), which is 94% of the value expected from the average mass.”	Não diferencia o espaço que existe entre os números e identifica-os como intervalo de números.
100â“250 g	”The largest owl (Sceloglaux albifacies), at 600 g, wasstill too small to prey on young kiwi it fed mainly onlizards, and small birds weighing 100â“250 g (Worthy,2001).”	Existência de caracteres às vezes dificulta a extração e para que seja possivel apanhar todos os casos dá origem a alguns dados que não são relevantes.
2.02.3 kg	”ge in Days 5 15 25 30 40 60 Parentreared 130170 grams 500800 grams 1.01.2 kg 1.61.8 kg 2.02.3 kg 2.83.3 kg Handreared 90120 grams 300400 grams 700900 grams 1.01.4 kg 1.52.3 kg 2.83.3 kg The food regurgitated to newly hatched chicks is very watery.”	Números associados a outros números com vários pontos, o sistema não consegue diferenciar e saber qual o número.
4555 days	”Mean growth rates Â SD of 5 hoatzin sibling pairs with respect to hatching rank during the fledging period measured at 715 days and at 4555 days of age, respectively.”	Números que deveriam ser um intervalo e que devido à interpretação do leitor de *pdf* foram unidos e passaram a ser números conjuntos.

Ao analisármos os vários artigos obtivemos no total 1032 candidatos que se revelaram bons candidatos para serem analisados pelo classificador. De seguida apresentamos nas tabelas de candidatos aceites, alguns exemplos de candidatos que foram corretamente classificados. Na tabela 5.3 mostramos os exemplos de frases para a categoria *Body Mass*, na tabela 5.4 mostramos os exemplos para a categoria *Body Temperature*, na tabela 5.5 para a categoria *Egg Temperature*, na tabela 5.6 as frases da categoria *Fledging*, e nas últimas duas tabelas, tabela 5.7 e tabela 5.8 os exemplos correspondentes às categorias *Incubation* e *Total Body Water*.

Tabela 5.3: Candidatos Aceites (*Body Mass*)

Body Mass	
Valor Extraído	Frase de Origem
26.8 g; 25.2 g; 18.5 g	"The following weights were recorded: Unpipped egg: 26.8 g Pipped egg: 25.2 g Hatchling owls: 18.5 g; 18.4 g"
358.78 g	"Ramphastos dicolorus.The RedBreasted Toucan was represented by two individuals, a male, whose mass was 356.7 + 2.08 g (n = 22)"
635.8g ;528.6g	"Of two Toco Toucans, the male had a mean mass equal to 634.8 + 1.00 g (n = 31) and the female had a mass equal to 527.6 + 1.00 g (n = 30)"
107.7g; 52.5g	"We found similar (butlower) values in the rhea: In a 3dayold bird theyolk content in this species was still 29.4% (107.7g fresh mass 52.5 g dry mass) of the juvenilemass"

Tabela 5.4: Candidatos Aceites (*Body Temperature*)

Body Temperature	
Valor Extraído	Frase de Origem
38.8 C; 40.0 C	"On that date the abdominal skin temperature was 38.8 C and the cloaca1 temperature was 40.0 C."
22 C	"The responses of the body temperatures of the owl chicks at different ages to ambient temperatures of about 22 C are shown in Figs 1."
7 C	"As they grew older the chicks showed only gradual improvement in body temperature regulation, and even at 10 days of age there was a decline of about 7 C during one hour of exposure to moderate air temperature."
28.5 C	"The 3 day old chick began strong total body shivering at a body temperature of 28.5 C, but its temperature continued to decline."

Tabela 5.5: Candidatos Aceites (*Egg Temperature*)

Egg Temperature	
Valor Extraído	Frase de Origem
35.5 C	“On 6 April, at an air temperature of 23 C, the prepared egg temperature remained constant at 35.5 C for 20 minutes.”
39.8 C; 39.3 C	“Only the females developed an incubation patch, and its temperature was 39.8 C (Speotyto) and 39.3 C (Tyto).”
32 C	“The effect of temperature induced drift on the tuners was eliminated by placing the tuners and demodulator in an incubator held at 32 C”
31 C	”In both species, the daytime equilibrium temperatures at embryo depth hover around 31C”

Tabela 5.6: Candidatos Aceites (*Fledging*)

Fledging	
Valor Extraído	Frase de Origem
10 d	“The older of the two chicks grew much more rapidly than the other the latter disappeared from the nest on 22 June (age about 10 days) and was presumably eaten by the female parent.”
6.25 d (0.5 to 12 days)	“Barth (1949) recorded body temperatures of nestling Snowy Owls (Nyctea from 0.5 to 12 days of age exposed to air temperatures of 5.5 to scandiaca) 10.5 C for intervals of I5 to 29 minutes after the departure of the brooding parent.”
50 d	”Weight and frequency of feedings decline during the last 50 days in the nest, but the fasts chicks undergo at this time do not exceed those experienced earlier in the fledgling period”
2 d	”A single bird at the nest more than two days before egglaying is usually the male if numerous observations indicate that the same bird is nearly always present.”

Tabela 5.7: Candidatos Aceites (*Incubation*)

Incubation	
Valor Extraído	Frase de Origem
3 d	“YBM incubated for only 3 days before deserting, and 8 days later she was keeping company and copulating with PBG at the nest he had selected.”
85 d	“Two eggs that took 85 days to hatch were excluded from the calculation of this mean in one the egg twice rolled in the dove prion is prolonged out of a badly built nest and remained unincubated from I to 48 hours until replaced by the observer.”
78.5 d (75 to 82 days)	”The lengths of the shifts by both sexes vary throughout the 75 to 82 days of incubation”
26 d; 38 d	”Moreover, in exulans (Figure 25) during the first half of incubation ranges of male shifts are always higher than those of females, and the longest a female was known to incubate was 26 days compared to 38 days by a male.’.”

Tabela 5.8: Candidatos Aceites (*Total Body Water*)

Total Body Water	
Valor Extraído	Frase de Origem
71.2% (69.7 + 1.5%)	“Larger trout (21 g)fed to 35dayold chicks had a similar watercontent of 69.7 + 1.5% (n = 2).”
2% (1 to 3%)	“The estimated loss would have been about 2 to 5% of the body weight over a 24 hr period (Bartholomew & Dawson, 1953) or a possible error of 1 to 3% in our estimates of total body water.”
40%; 50%	”Birds with salt glands, then, have about 40% of their body water in the extracellular space, while the rooster had about 50% of its water outside of cells ”
68%	”The % total body water of wild emus was 68%, notably higher than in the current experiments”

5.2 Avaliação da Classificação

Depois de proceder à obtenção dos candidatos, estes foram utilizados como dados de teste no módulo de Classificação. Para proceder à classificação, utilizámos diversos classificadores de modo a compreender qual o mais adequado para o nosso problema. Como o número de exemplos positivos eram muito poucos (ver tabela 5.9), decidimos não só ter foco na métrica de avaliação *Accuracy* mas também dar relevo aos valores de *F1-Score* para a categoria que estávamos a estudar.

Tabela 5.9: Número de exemplos positivos por categoria

Categoria	Número de exemplos
Body Mass	6
Body Temperature	11
Egg Temperature	15
Fledging	10
Incubation	12
Total Body Water	6

Para verificar qual o classificador se adaptava melhor ao nosso problema, testámos as duas abordagens distintas.Primeiramente juntámos os candidatos de todas as categorias e procedemos à classificação tendo em conta todas as categorias e, de seguida, de seguida realizámos os testes para cada categoria em separado tendo os seus candidatos e os seus exemplos positivos. Seguidamente apresentamos os resultados para todas as categorias no mesmo sistema e para cada categoria em sistemas diferentes.

5.2.1 Resultados para todas as categorias

Como podemos observar na tabela 5.10, os classificadores que obtiveram maior valor de *Accuracy* foram Árvore de Decisão com 0.973534 e *Random Forest* com 0.95463. Os restantes, apesar de terem obtido *Accuracy* superior a 0.60, os valores de *F1-Score* não foram todos positivos, existindo em cada caso pelo menos uma das categorias com *F1-Score* indefinida, o que indica que não foram identificados casos para essa categoria e os resultados foram enviasados para as restantes categorias. O classificador *Naïve Bayes* obteve uma *Accuracy* de 0.17580 e para as várias categorias valores de *F1-Score* inferiores a 0.45.

No classificador Árvore de Decisão a categoria *Body Mass* obteve 1.00 de *F1-Score*, a categoria *Body Temperature* obteve 0.96, *Egg Temperature* obteve 0.92, *Fledging* obteve 0.95, *Incubation* obteve 0.94 e *Total Body Water* obteve 0.92 para *F1-Score*. No classificador *Random Forest* a categoria *Body Mass* obteve 0.96 de *F1-Score*, a categoria *Body Temperature* obteve 0.94, *Egg Temperature* obteve 0.87, *Fledging* obteve 0.87, *Incubation* obteve 0.88 e *Total Body Water* obteve 0.96 para *F1-Score*.

Tabela 5.10: Resultados para todas as categorias

			Body Mass	Body Temp.	Egg Temp.	Fledging	Incubation	Total Body Water	Other
Naïve Bayes	Accuracy 0.17580	Precision	0.16	0.30	0.10	0.14	0.08	0.25	0.76
		Recall	0.50	0.91	0.33	0.20	0.75	0.83	0.06
		F1-Score	0.24	0.45	0.15	0.17	0.15	0.38	0.11
SVM	Accuracy 0.84877	Precision	0.89	0.91	0.56	0.00	1.00	0.91	0.86
		Recall	0.67	0.91	0.33	0.00	0.17	0.83	0.97
		F1-Score	0.76	0.91	0.42	∞	0.29	0.87	0.91
SVM Linear	Accuracy 0.61625	Precision	0.10	0.00	0.00	0.13	0.09	0.00	0.79
		Recall	0.25	0.00	0.00	0.10	0.08	0.00	0.78
		F1-Score	0.14	∞	∞	0.11	0.09	∞	0.79
Regressão Logística	Accuracy 0.81096	Precision	0.57	0.75	0.64	0.50	0.00	0.71	0.83
		Recall	0.33	0.55	0.30	0.10	0.00	0.42	0.97
		F1-Score	0.42	0.63	0.41	0.17	∞	0.53	0.89
Random Forest	Accuracy 0.95463	Precision	0.92	0.88	0.82	0.89	0.85	1.00	0.98
		Recall	1.00	1.00	0.93	0.85	0.92	0.92	0.96
		F1-Score	0.96	0.94	0.87	0.87	0.88	0.96	0.97
Árvore de Decisão	Accuracy 0.973534	Precision	1.00	0.92	0.86	0.91	0.89	0.86	1.00
		Recall	1.00	1.00	1.00	1.00	1.00	1.00	0.97
		F1-Score	1.00	0.96	0.92	0.95	0.94	0.92	0.98
K Vizinhos Mais Próximos	Accuracy 0.73724	Precision	0.00	0.70	0.45	0.11	0.22	0.44	0.84
		Recall	0.00	0.64	0.47	0.10	0.25	0.33	0.86
		F1-Score	∞	0.67	0.46	0.11	0.24	0.38	0.85

5.2.2 Resultados para *Body Mass*

Como podemos observar na tabela 5.11, os classificadores obtiveram valores muito próximos. Os classificadores *Random Forest*, Árvore de Decisão e K Vizinhos Mais Próximos obtiveram uma *Accuracy* de 0.95652 com *F1-Score* de 0.92 para a categoria de *Body Mass*. Os classificadores SVM Linear e Regressão Logística obtiveram uma *Accuracy* de 0.91304 com um *F1-Score* de 0.83 e o classificador *Naïve Bayes* conseguiu obter uma *Accuracy* de 0.9 com um *F1-Score* de 0.92. Por fim o classificador SVM apresentou *Accuracy* de 0.75, contudo o valor de *F1-Score* foi indefinido devido aos valores terem sido enviasados para a classe *Other* e, deste modo, não identificarem nenhum caso como sendo da categoria *Body Mass*.

Tabela 5.11: Resultados para *Body Mass*

			Body Mass	Other
Naïve Bayes	Accuracy 0.9	Precision	0.86	1.00
		Recall	1.00	0.75
		F1-Score	0.92	0.86
SVM	Accuracy 0.6	Precision	0.00	0.60
		Recall	0.00	1.00
		F1-Score	∞	0.75
SVM Linear	Accuracy 0.91301	Precision	0.83	0.94
		Recall	0.83	0.94
		F1-Score	0.83	0.94
Regressão Logística	Accuracy 0.91304	Precision	0.83	0.94
		Recall	0.83	0.94
		F1-Score	0.83	0.94
Random Forest	Accuracy 0.95652	Precision	0.86	1.00
		Recall	1.00	0.94
		F1-Score	0.92	0.97
Árvore de Decisão	Accuracy 0.95652	Precision	0.86	1.00
		Recall	1.00	0.94
		F1-Score	0.92	0.97
K Vizinhos Mais Próximos	Accuracy 0.95652	Precision	0.86	1.00
		Recall	1.00	0.94
		F1-Score	0.92	0.97

5.2.3 Resultados para *Body Temperature*

Como podemos observar na tabela 5.12, os classificadores *Random Forest* e Árvore de Decisão obtiveram valores de *Accuracy* bastante próximos, respetivamente, 0.97674 e 0.95348, contudo os valores de *F1-Score* foram diferenciadores, sendo 0.96 e 0.90 respetivamente. O classificador Regressão Logística obteve uma *Accuracy* de 0.93023 com um *F1-Score* de 0.88 para a categoria *Body Temperature* e o K Vizinhos Mais Próximos obteve *Accuracy* de 0.95348 e *F1-Score* de 0.92. De seguida, o classificador SVM obteve 0.83720 de *Accuracy* e 0.59 de *F1-Score* e o classificador SVM Linear com *Accuracy* de 0.69767 e *F1-Score* de 0.48. Por fim, com o pior resultado, o classificador *Naïve Bayes* com uma *Accuracy* de 0.32558 e *F1-Score* de 0.43 para a categoria *Body Temperature*.

Tabela 5.12: Resultados para *Body Temperature*

			Body Temperature	Other
Naïve Bayes	Accuracy 0.32558	Precision	0.28	1.00
		Recall	1.00	0.09
		F1-Score	0.43	0.17
SVM	Accuracy 0.83720	Precision	0.83	0.84
		Recall	0.45	0.97
		F1-Score	0.59	0.90
SVM Linear	Accuracy 0.69767	Precision	0.43	0.83
		Recall	0.55	0.75
		F1-Score	0.48	0.79
Regressão Logística	Accuracy 0.93023	Precision	0.79	1.00
		Recall	1.00	0.91
		F1-Score	0.88	0.95
Random Forest	Accuracy 0.97674	Precision	0.92	1.00
		Recall	1.00	0.97
		F1-Score	0.96	0.98
Árvore de Decisão	Accuracy 0.95348	Precision	1.00	0.94
		Recall	0.82	1.00
		F1-Score	0.90	0.97
K Vizinhos Mais Próximos	Accuracy 0.95348	Precision	0.85	1.00
		Recall	1.00	0.94
		F1-Score	0.92	0.97

5.2.4 Resultados para *Egg Temperature*

Como podemos observar na tabela 5.13 os classificadores *Random Forest* e Árvore de Decisão obtiveram os melhores resultados, tendo uma *Accuracy* de 0.81666 e 0.83333 respetivamente e com *F1-Score* de 0.62 e 0,67 para a categoria *Egg Temperature*. De seguida, o classificador SVM obteve uma *Accuracy* de 0.76666 com *F1-Score* de 0.42 e o K Vizinhos Mais Próximos esteve por valores de *Accuracy* muito próximos, nomeadamente, 0.75 com um *F1-Score* de 0.71. O classificador Regressão Logística obteve uma *Accuracy* de 0.7 com um *F1-Score* de 0.71 para a categoria de *Egg Temperature* e o classificador SVM Linear obteve uma *Accuracy* de 0.6666 com um *F1-Score* de 0.23. Por fim, o classificador *Naïve Bayes* conseguiu obter uma *Accuracy* de 0.56666 com um *F1-Score* de 0.58.

Tabela 5.13: Resultados para *Egg Temperature*

			Egg Temperature	Other
Naïve Bayes	Accuracy 0.56666	Precision	0.56	0.57
		Recall	0.60	0.53
		F1-Score	0.58	0.55
SVM	Accuracy 0.76666	Precision	0.56	0.80
		Recall	0.33	0.91
		F1-Score	0.42	0.85
SVM Linear	Accuracy 0.66666	Precision	0.27	0.76
		Recall	0.20	0.82
		F1-Score	0.23	0.79
Regressão Logística	Accuracy 0.7	Precision	0.69	0.71
		Recall	0.73	0.67
		F1-Score	0.71	0.69
Random Forest	Accuracy 0.81666	Precision	0.64	0.87
		Recall	0.60	0.89
		F1-Score	0.62	0.88
Árvore de Decisão	Accuracy 0.83333	Precision	0.67	0.89
		Recall	0.67	0.89
		F1-Score	0.67	0.89
K Vizinhos Mais Próximos	Accuracy 0.75	Precision	0.63	0.73
		Recall	0.80	0.53
		F1-Score	0.71	0.62

5.2.5 Resultados para *Fledging*

Como podemos observar na tabela 5.14 os classificadores Árvore de Decisão e *Random Forest* obtiveram os melhores resultados, tendo uma *Accuracy* de 0.9 e 0.84615 respetivamente e com *F1-Score* de 0.78 e 0,67 para a categoria de *Fledging*. De seguida, o classificador Regressão Logística obteve uma *Accuracy* de 0.825 com um *F1-Score* de 0.59 e, de seguida, o classificador K Vizinhos Mais Próximos obteve uma *Accuracy* de 0.725 com *F1-Score* de 0.35. O classificador *Naïve Bayes* conseguiu obter uma *Accuracy* de 0.71429 com um *F1-Score* de 0.57 e o classificador SVM Linear obteve uma *Accuracy* de 0.6666 com um *F1-Score* de 0.38. Por fim o classificador SVM apresentou *Accuracy* de 0.75, contudo o valor de *F1-Score* foi indefinido devido aos valores terem sido enviasados para a classe *Other* e, deste modo, não identificarem nenhum caso como sendo da categoria *Fledging*.

Tabela 5.14: Resultados para *Fledging*

			Fledging	Other
Naïve Bayes	Accuracy 0.71429	Precision	1.00	0.65
		Recall	0.40	1.00
		F1-Score	0.57	0.79
SVM	Accuracy 0.75	Precision	0.00	0.75
		Recall	0.00	1.00
		F1-Score	∞	0.86
SVM Linear	Accuracy 0.66666	Precision	0.36	0.79
		Recall	0.40	0.76
		F1-Score	0.38	0.77
Regressão Logística	Accuracy 0.825	Precision	0.71	0.85
		Recall	0.50	0.93
		F1-Score	0.59	0.89
Random Forest	Accuracy 0.84615	Precision	0.75	0.87
		Recall	0.60	0.93
		F1-Score	0.67	0.90
Árvore de Decisão	Accuracy 0.9	Precision	0.88	0.91
		Recall	0.70	0.97
		F1-Score	0.78	0.94
K Vizinhos Mais Próximos	Accuracy 0.725	Precision	0.43	0.79
		Recall	0.30	0.87
		F1-Score	0.35	0.83

5.2.6 Resultados para *Incubation*

Como podemos observar na tabela 5.15 os classificadores Árvore de Decisão e *Random Forest* obtiveram os melhores resultados, tendo uma *Accuracy* de 0.89361 e 0.875 respetivamente e com *F1-Score* de 0.80 e 0.75 para a categoria de *Incubation*. De seguida, o classificador Regressão Logística obteve uma *Accuracy* de 0.825 com um *F1-Score* de 0.87. Os classificadores K Vizinhos Mais Próximos, e SVM obtiveram o mesmo valor de *Accuracy* de 0.8125, diferenciando-se no valor de *F1-Score* em que o primeiro obteve 0.57 e o segundo 0.40. De seguida, o classificador SVM Linear obteve *Accuracy* de 0.70212 com *F1-Score* de 0.42 e, por fim, o classificador *Naïve Bayes* conseguiu obter uma *Accuracy* de 0.70588 com um *F1-Score* de 0.64.

Tabela 5.15: Resultados para *Incubation*

			Incubation	Other
Naïve Bayes	Accuracy 0.70588	Precision	0.56	0.83
		Recall	0.75	0.68
		F1-Score	0.64	0.75
SVM	Accuracy 0.8125	Precision	1.00	0.80
		Recall	0.25	1.00
		F1-Score	0.40	0.89
SVM Linear	Accuracy 0.70212	Precision	0.42	0.80
		Recall	0.42	0.80
		F1-Score	0.42	0.80
Regressão Logística	Accuracy 0.825	Precision	0.91	0.95
		Recall	0.83	0.97
		F1-Score	0.87	0.96
Random Forest	Accuracy 0.875	Precision	0.75	0.92
		Recall	0.75	0.92
		F1-Score	0.75	0.92
Árvore de Decisão	Accuracy 0.89361	Precision	0.77	0.94
		Recall	0.83	0.91
		F1-Score	0.80	0.93
K Vizinhos Mais Próximos	Accuracy 0.8125	Precision	0.67	0.85
		Recall	0.50	0.92
		F1-Score	0.57	0.88

5.2.7 Resultados para *Total Body Water*

Como podemos observar na tabela 5.16 o classificador Regressão Logística obteve o melhor valor de *Accuracy* de 0.95652 com *F1-Score* de 0.92. De seguida os classificadores *Random Forest* e K Vizinhos Mais Próximos obtiveram os mesmos valores, nomeadamente, 0.91304 de *Accuracy* e 0.83 de *F1-Score*. De seguida, o classificador Árvore de Decisão obteve uma *Accuracy* de 0.875 com *F1-Score* de 0.73. De seguida o classificador *Naïve Bayes* teve 0.82608 de *Accuracy* e 0.75 de *F1-Score* e, o classificador SVM Linear obteve 0.73913 de *Accuracy* e 0.50 de *F1-Score*. Por fim o classificador SVM apresentou *Accuracy* de 0.75, contudo o valor de *F1-Score* foi indefinido devido aos valores terem sido enviasados para a classe *Other* e, deste modo, não identificarem nenhum caso como sendo da categoria *Total Body Water*.

Tabela 5.16: Resultados para *Total Body Water*

			Total Body Water	Other
Naïve Bayes	Accuracy 0.82608	Precision	0.60	1.00
		Recall	1.00	0.76
		F1-Score	0.75	0.87
SVM	Accuracy 0.75	Precision	0.00	0.75
		Recall	0.00	1.00
		F1-Score	∞	0.8
SVM Linear	Accuracy 0.73913	Precision	0.50	0.82
		Recall	0.50	0.82
		F1-Score	0.50	0.82
Regressão Logística	Accuracy 0.95652	Precision	0.86	1.00
		Recall	1.00	0.94
		F1-Score	0.92	0.97
Random Forest	Accuracy 0.91304	Precision	0.75	0.92
		Recall	0.83	0.94
		F1-Score	0.83	0.93
Árvore de Decisão	Accuracy 0.875	Precision	0.80	0.89
		Recall	0.67	0.94
		F1-Score	0.73	0.92
K Vizinhos Mais Próximos	Accuracy 0.91304	Precision	0.83	0.94
		Recall	0.83	0.94
		F1-Score	0.83	0.94

5.2.8 Análise dos resultados

Para cada uma das categorias observámos os resultados de avaliação, nomeadamente *Accuracy* e *F1-Score*. Como pudemos verificar, quando experimentámos todas as categorias no mesmo sistema, os maiores valores de *Accuracy* foram 0.973534 para o classificador Árvore de Decisão e, 0.95463 para o classificador *Random Forest*.

Relativamente aos valores dos sistemas individuais para cada categoria, podemos verificar na figura 5.2 que os classificadores Árvore de Decisão, K Vizinhos Mais Próximos e *Random Forest* obtiveram os melhores resultados para a categoria *Body Mass*, sendo 0.95652 de *Accuracy* e 0.92 de *F1-Score*. Relativamente à categoria *Body Temperature*, o classificador que obteve melhores resultados foi o *Random Forest* com 0.97674 de *Accuracy* e 0.96 de *F1-Score*, como se pode verificar na figura 5.3. O classificador Árvore de Decisão obteve os melhores resultados para a categoria de *Egg Temperature*, com 0.83333 de *Accuracy* e 0,67 de *F1-Score* (ver figura 5.4), para a categoria *Fledging*, com 0.9 de *Accuracy* e 0,78 de *F1-Score* (ver figura 5.5) e para a categoria *Incubation*, com 0,89361 de *Accuracy* e 0,80 de *F1-Score* (ver figura 5.6). Por fim, para a categoria *Total Body Water* o classificador que obteve melhor resultado foi o Regressão Logística, obtendo 0.95652 de *Accuracy* e 0.92 de *F1-Score*, a ver na figura 5.7.

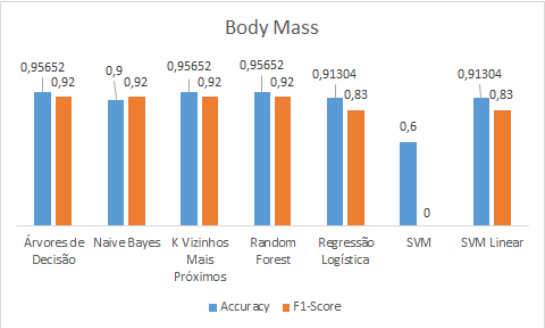

Figura 5.2: Resultados Body Mass

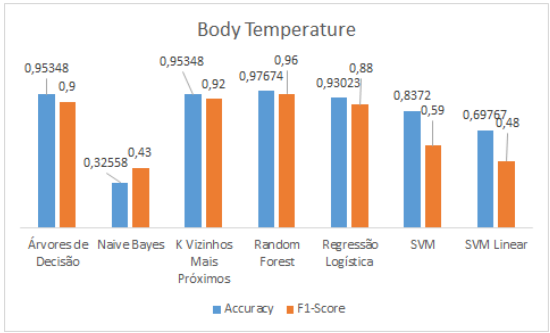

Figura 5.3: Resultados Body Temperature

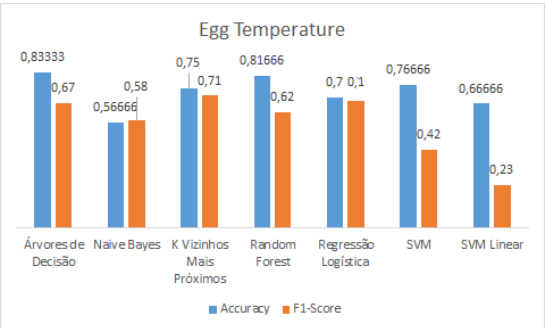

Figura 5.4: Resultados Egg Temperature

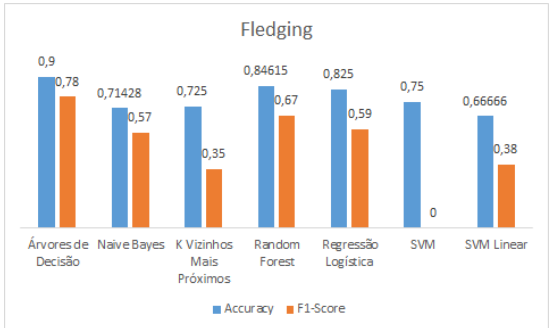

Figura 5.5: Resultados Fledging

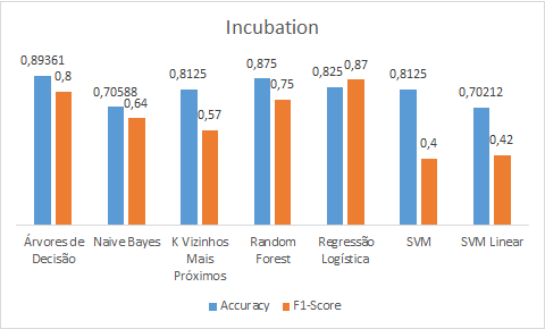

Figura 5.6: Resultados Incubation

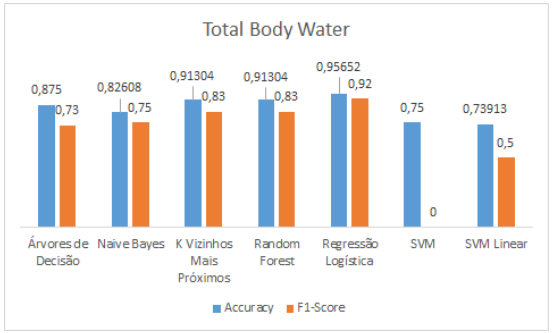

Figura 5.7: Resultados Total Body Water

Como podemos verificar os classificadores Árvore de Decisão e *Random Forest* foram os que obtiveram melhores resultados tanto no sistema com todas as categorias juntas como nos vários sistemas para cada categoria. Nestes últimos, o Árvore de Decisão obteve o melhor resultado nas categorias *Body Mass*, *Egg Temperature*, *Fledging* e *Incubation* e, por outro lado, o classificador *Random Forest* obteve nas categorias *Body Mass* e *Body Temperature*.

5.3 Avaliação depois da Seleção de Características

Tendo em conta os resultados obtidos no módulo anterior foi necessário escolher que sistemas iriamos usar para obter melhores resultados. Decidimos descartar o sistema que juntava todas as categorias devido à sua complexidade, pois existiria a possibilidade de melhorar para determinadas categorias e piorar para outras, seria necessário utilizar como métrica de referência a *Accuracy* e, como pudemos verificar existiram situações em que numa das categorias o valor de *F1-Score* era indefinido. Deste modo, se fossemos utilizar esta abordagem poderiamos estar a criar falsos valores.

Nos vários sistemas das várias categorias, decidimos realizar a Seleção de Características utilizando o classificador Árvore de Decisão e *f1-Score* comométrica de referência.

Apresentamos de seguida para cada categoria o máximo de valores de *F1-Score* tomados no processo de Seleção de Características e os resultados para o classificador Árvore de Decisão.

5.3.1 Resultados para *Body Mass*

Ao executar o algoritmo para seleção de características, foram contabilizados os valores de *F1-Score* que o classificador Árvore de Decisão obteve em cada iteração enquanto percorria a sequência de características durante os vários momentos para a sua constituição. No caso da categoria *Body Mass* em todas as cento e vinte iterações iniciais a avaliação foi de 0.92 e para as restantes 1.0 de *F1-Score*.

Como é possível verificar na tabela 5.21, no total foram definidas 436 características e foram selecionadas 136 nesta seleção.

Tabela 5.17: Características para *Body Mass*

Grupo de Características	Características selecionadas	Todas as Características
Ocorrência de palavras	45	295
Total de palavras	1	1
Distância entre valor e palavras específicas	40	58
Intervalo com números parametrizados	20	28
Distância a números parametrizados	30	54
Total	136	436

Ao avaliar os candidatos com a nova seleção de características, o classificador Árvore de Decisão obteve para *Accuracy* 0.95652 e *F1-Score* de 0.92, como podemos verificar na tabela 5.18.

Tabela 5.18: Resultados para *Body Mass*

	Árvore de Decisão		
	Accuracy:0.95652		
	Precision	Recall	F1-Score
Body Mass	0.86	1.00	0.92
Other	1.00	0.94	0.97

5.3.2 Resultados para *Body Temperature*

Ao executar o algoritmo para seleção de características, foram contabilizados os valores de *F1-Score* que o classificador Árvore de Decisão obteve em cada iteração enquanto percorria a sequência de características durante os vários momentos para a sua constituição. No caso da categoria *Body Temperature* a primeira característica foi obtida com máximo de 0.8461 de *F1-Score* e as restantes com 1.0 de *F1-Score*.

Como é possível verificar na tabela 5.19, no total foram definidas 617 características e foram selecionadas 255 nesta seleção. Das características selecionadas as que tiveram mais impacto foram várias palavras ocorridas nos candidatos.

Tabela 5.19: Características para *Body Temperature*

Grupo de Características	Características selecionadas	Todas as Características
Ocorrência de palavras	246	476
Total de palavras	1	1
Distância entre valor e palavras específicas	3	58
Intervalo com números parametrizados	5	28
Distância a números parametrizados	0	54
Total	255	617

Ao avaliar os candidatos com a nova seleção de características, o classificador Árvore de Decisão obteve 0.97674 de *Accuracy* com 0.95 de *F1-Score*, como podemos verificar na tabela 5.20.

Tabela 5.20: Resultados para *Body Temperature*

	Árvore de Decisão		
	Accuracy:0.97674		
	Precision	Recall	F1-Score
Body Temperature	1.00	0.91	0.95
Other	0.97	1.00	0.98

5.3.3 Resultados para *Egg Temperature*

Ao executar o algoritmo para seleção de características, foram contabilizados os valores de *F1-Score* que o classificador Árvore de Decisão obteve em cada iteração enquanto percorria a sequência de características durante os vários momentos para a sua constituição. No caso da categoria *Egg Temperature* a primeira característica foi selecionada com máximo de 0.7647 de *F1-Score*, a segunda e a terceira com máximo de 0.896 e, as restantes, com máximo de 0.9375 de *F1-Score*.

Como é possível verificar na tabela 5.21, no total foram definidas 436 características e foram selecionadas 136 nesta seleção. Das características selecionadas, as diversas palavras encontradas nos candidatos foram mais significativos.

Tabela 5.21: Características para *Egg Temperature*

Grupo de Características	Características selecionadas	Todas as Características
Ocorrência de palavras	321	520
Total de palavras	0	1
Distância entre valor e palavras específicas	5	58
Intervalo com números parametrizados	6	28
Distância a números parametrizados	0	54
Total	332	661

Ao avaliar os candidatos com a nova seleção de características, o classificador Árvore de Decisão obteve 0.93333 de *Accuracy* com 0.88 de *F1-Score*, como podemos verificar na tabela 5.22.

Tabela 5.22: Resultados para *Egg Temperature*

	Árvore de Decisão		
	Accuracy:0.93333		
	Precision	Recall	F1-Score
Egg Temperature	0.79	1.00	0.88
Other	1.00	0.91	0.95

5.3.4 Resultados para *Fledging*

Ao executar o algoritmo para seleção de características, foram contabilizados os valores de *F1-Score* que o classificador Árvore de Decisão obteve em cada iteração enquanto percorria a sequência de características durante os vários momentos para a sua constituição. No caso da categoria *Fledging* na primeira iteração a característica selecionada obteve como valor máximo de F1-Score de 0.8571, a segunda obteve 0.9000 para F1-Score. As duzentas características seguintes foram obtidas com o máximo de 0.9523 e, para as restantes iterações a avaliação foi de 1.0 de *F1-Score*.

Como é possível verificar na tabela 5.21, no total foram definidas 436 características e foram selecionadas 136 nesta seleção.

Tabela 5.23: Características para *Fledging*

Grupo de Caracteristicas	Características selecionadas	Todas as Características
Ocorrência de palavras	194	268
Total de palavras	0	1
Distância entre valor e palavras específicas	13	58
Intervalo com números parametrizados	9	28
Distância a números parametrizados	8	54
Total	224	409

Ao avaliar os candidatos com a nova seleção de características, o classificador Árvore de Decisão obteve 0.95 de *Accuracy* com 0.90 de *F1-Score*, como podemos verificar na tabela 5.24.

Tabela 5.24: Resultados para *Fledging*

	Árvore de Decisão		
	Accuracy:0.95		
	Precision	Recall	F1-Score
Fledging	0.90	0.90	0.90
Other	0.97	0.97	0.97

5.3.5 Resultados para *Incubation*

Ao executar o algoritmo para seleção de características, foram contabilizados os valores de *F1-Score* que o classificador Árvore de Decisão obteve em cada iteração enquanto percorria a sequência de características durante os vários momentos para a sua constituição. No caso da categoria *Incubation* na primeira iteração a característica selecionada obteve como valor máximo de F1-Score de 0.8695 e para as restantes iterações a avaliação foi de 0.9166 de *F1-Score*.

Como é possível verificar na tabela 5.25, no total foram definidas 507 características e foram utilizadas 417 nesta seleção.

Tabela 5.25: Características para *Incubation*

Grupo de Características	Características selecionadas	Todas as Características
Ocorrência de palavras	330	366
Total de palavras	0	1
Distância entre valor e palavras específicas	20	58
Intervalo com números parametrizados	28	28
Distância a números parametrizados	39	54
Total	417	507

Ao avaliar os candidatos com a nova seleção de características, o classificador Árvore de Decisão obteve 0.91489 de *Accuracy* com 0.82 de *F1-Score*, como podemos verificar na tabela 5.26.

Tabela 5.26: Resultados para *Incubation*

	Árvore de Decisão		
	Accuracy:0.91489		
	Precision	Recall	F1-Score
Incubation	0.90	0.75	0.82
Other	0.92	0.97	0.94

5.3.6 Resultados para *Total Body Water*

Ao executar o algoritmo para seleção de características, foram contabilizados os valores de *F1-Score* que o classificador Árvore de Decisão obteve em cada iteração enquanto percorria a sequência de características durante os vários momentos para a sua constituição. No caso da categoria *Total Body Water* em todas as iterações a avaliação foi de 0.90909 de *F1-Score*.

Como é possível verificar na tabela 5.27, no total foram definidas 409 características e foram utilizadas 224 nesta seleção.

Tabela 5.27: Características para *Total Body Water*

Grupo de Características	Características selecionadas	Todas as Características
Ocorrência de palavras	100	179
Total de palavras	1	1
Distância entre valor e palavras específicas	40	58
Intervalo com números parametrizados	20	28
Distância a números parametrizados	50	54
Total	299	320

Ao avaliar os candidatos com a nova seleção de características, o classificador Árvore de Decisão obteve 0.91666 de *Accuracy* com 0.80 de *F1-Score*, como podemos verificar na tabela 5.28.

Tabela 5.28: Resultados para *Total Body Water*

	Árvore de Decisão		
	Accuracy:0.91666		
	Precision	Recall	F1-Score
Total Body Water	1.00	0.67	0.80
Other	0.90	1.00	0.95

5.4 Análise de Resultados Final

Analisando os valores obtidos nos sistemas de cada categoria, podemos verificar que na fase de classificação sem proceder à Seleção de Características os classificadores que obtiveram melhores resultados foram o Árvore de Decisão e o *Random Forest*, obtendo um máximo de *Accuracy* de 0.95652 e 0.97674 respetivamente, o primeiro na categoria Body Mass e o segundo na categoria Body Temperature. O mínimo de valor que obtiveram foi de 0.83333 e 0.81666, respetivamente, na categoria *Egg Temperature*. Como o classificador Árvore de Decisão obteve bons resultados para mais características, decidimos utilizá-lo para a Seleção de Características.

Comparando os resultados obtidos, para a categoria *Body Mass* (ver figura 5.8) com a realização da seleção de características os valores de *Accuracy* e *F1-Score* mantiveram-se. Para as restantes categorias, *Body Temperature* (ver figura 5.9) , *Egg Temperature* (ver figura 5.10), *Fledging* (ver figura 5.11, *Incubation* (ver figura 5.12) e *Total Body Water* (ver figura 5.13) houve uma melhoria nos resultados, convergindo para uma melhoria nas métricas de avaliação de 2% a 10%. Apesar dos valores para a categoria *Body Mass* não terem sofrido alterações, podemos observar que várias características existentes não eram relevantes e que com menos características foi possível obter o mesmo resultado.

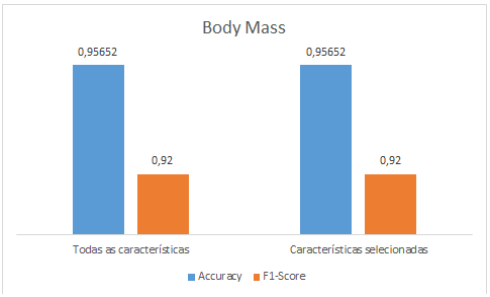

Figura 5.8: Resultados Body Mass

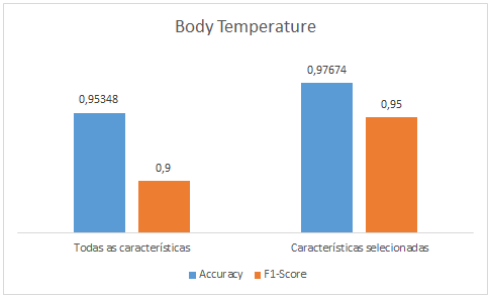

Figura 5.9: Resultados Body Temperature

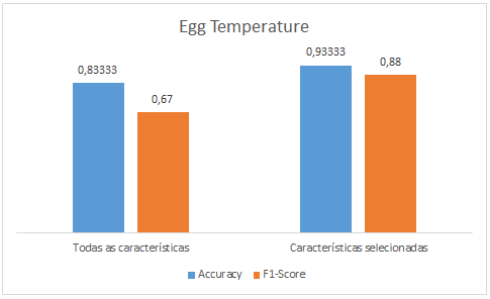

Figura 5.10: Resultados Egg Temperature

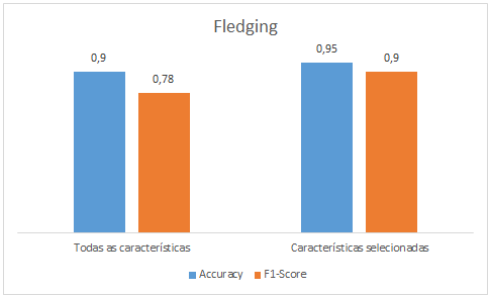

Figura 5.11: Resultados Fledging

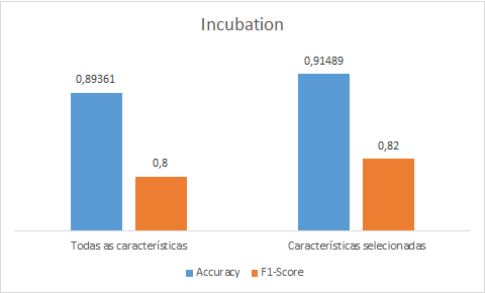

Figura 5.12: Resultados Incubation

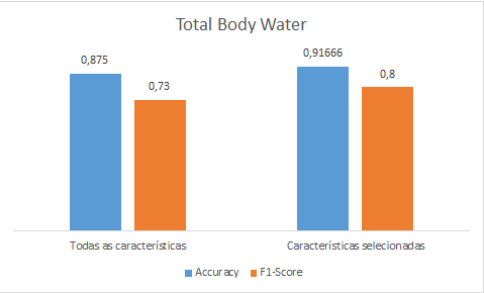

Figura 5.13: Resultados Total Body Water

5.5 Sumário

Neste capítulo apresentámos os resultados obtidos com o nosso sistema relativamente à classificação sem realização de Seleção de Características e depois da sua aplicação com o classificador Árvore de Decisão. Por fim, fizemos uma análise comparativa dos valores obtidos.

Capítulo 6

Conclusões

Para dar resposta às necessidades de informação da Biologia e tomando como exemplo o grupo taxonómico das aves em que a maioria das bases de dados existentes apenas agregam informação sobre a sua fisionomia (valores de massa, comprimento), sendo raros os dados da sua fisiologia (temperatura média do corpo ou conteúdo hídrico), este trabalho surge com o principal objetivo de dar resposta à questão: "Será possível construir um sistema que possa extrair dados de determinadas espécies de aves a partir de artigos científicos?".

Para dar resposta à questão formulada, criámos um sistema que, ao receber artigos científicos e um documento com a descrição das características a extrair, procede à extração da informação relevante para preencher uma Base de Conhecimento. O sistema segue uma abordagem híbrida, complementando a abordagem segundo regras com a abordagem segundo Aprendizagem Automática e para a análise dos textos são utilizadas técnicas de Processamento de Língua Natural, Expressões Regulares e vários classificadores.

O sistema é constituído por três módulos: a extração de candidatos, classificação e seleção de características. No primeiro módulo são extraídos todos os candidatos para cada categoria, de seguida na classificação são definidas características para analisar os candidatos e estes são utilizados como dados nos vários classificadores. No último módulo são selecionadas as características relevantes para melhorar os resultados obtidos nos classificadores.

Realizámos experiências com vários classificadores utilizando um sistema com todas as categorias e vários sistemas para cada categoria individual. Os dois classificadores que obtiveram melhores resultados tanto no sistema com todas as categorias como nos sistemas de categorias individuais foram a Árvore de Decisão e *Random Forest*. De seguida, decidimos realizar a Seleção de Características com o classificador Árvore de Decisão nos sistemas de cada categoria e os valores obtidos tiveram em consideração a maximização da métrica de *F1-Score*. O resultado foi o aumento de *Accuracy* e aumento de candidatos identificados como pertencente à categoria. Assim sendo foi possível identificar 164 candidatos verdadeiros.

Em suma, nas duas abordagens experimentadas, o classificador que obteve melhores resultados foi Árvore de Decisão e, concluímos ser este o classificador mais indicado para o nosso problema.

Demonstrámos assim que é possível construir um sistema para a extração de dados a partir de artigos científicos no domínio da Biologia. Contudo, ainda não é possível ter um sistema completamente automático tornando-se relevante um utilizador humano que possa resolver ambiguidades nos resultados.

6.1 Limitações e recomendações para trabalhos futuros

O presente trabalho respondeu à questão central levantada e contribuiu para a extração de vários exemplos corretos, contudo, existem alguns pontos que podem ser aprofundados.

O primeiro ponto incide sobre a ferramenta utilizada para o leitor de *pdf*. Uma das grandes limitações deste trabalho referiu-se à interpretação da ferramenta sobre o texto proveniente dos documentos *.pdf*. Assim sendo, sugerimos a utilização de uma ferramenta mais poderosa que permita diferenciar o texto em várias colunas, as imagens e os gráficos e que permita obter a informação de modo correto, sem códigos incompreensíveis.

Na fase do classificador, tinhamos poucos exemplos para cada categoria, o que limitou vários resultados obtidos no módulo de classificação. Assim sendo, sugerimos a obtenção de mais exemplos positivos, talvez obtidos utilizando o método de *Distant Supervision* [31], como o utilizado nas resoluções dos participantes da competição de povoamento de Bases de Conhecimento. Deste modo, permitiria maior confiança nos resultados e, consequentemente, poder realizar a Seleção de Características tendo em conta os valores da métrica de avaliação *Accuracy*.

Num trabalho futuro também seria interessante implementar uma interface para a classificação para ser utilizada pelo utilizador.

Bibliografia

[1] Akella, L., Norton, C., Miller, H.: Netineti: discovery of scientific names from text using machine learning methods. BMC Bioinformatics 13, 211 (2012)

[2] Ananiadou, S., Kell, D.B., Tsujii, J.i.: Text mining and its potencial application in systems biology. Trends in Biotechnology 24(12), p571–579 (2006)

[3] Angeli, G., Gupta, S., Premkumar, M., Manning, C., Tibshirani, C.R., Wu, J.Y., Wu, S., Zhang, C.: Stanford's distantly supervised slot filling systems for kbp 2014 (2015)

[4] Asma Ben Abacha, P.Z.: A hybrid approach for the extraction of semantic relations from medline abstracts. In: Gelbukh, A. (ed.) Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science, vol. 6609, pp. 139–150. Springer Berlin Heidelberg (2011)

[5] Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edn. (2012)

[6] Bentor, Y., Viswanathan, V., Mooney, R.: University of texas at austin kbp 2014 slot filling system: Bayesian logic programs for textual inference. In: Proceedings of the TAC-KBP 2014 Workshop (2014)

[7] Berry, M.W., Kogan, J.: Text mining : applications and theory. Chichester, U.K. Wiley, Boston, MA, USA, 1 edn. (2010)

[8] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, J., VanderPlas, J., Joly, A., Holt, B., Varoquaux, G.: API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning. pp. 108–122 (2013)

[9] Campbell, N.: Biology: Concepts & Connections. Pearson/Benjamin Cummings (2006)

[10] Craven, M., Kumlien, J.: Constructing biological knowledge bases by extracting information from text sources. In: Proceedings of the International Conference on Intelligent Systems for Molecular Biology (1999)

[11] Davis, J., Goadrich, M.: OThe Relationship Between Precision-Recall and ROC Curve. In: Proceedings of the 23rd International Conference on Machine Learning. ACM. New York, NY, USA. pp.233–240 (2006)

[12] Department of Statistics: Chapter 12 Logistic Regression. September (2016). http://www.stat.cmu.edu/ cshalizi/uADA/12/lectures/ch12.pdf

[13] Emden, M.H.V., Kowalski, R.A.: The semantics of predicate logic as a programming language. Journal of the ACM 23, p.569–574 (1976)

[14] Govindan, V.K., Shivaprasad, A.P.: Character recognition-a review. Pattern Recogn. 23, p.671–683 (1990)

[15] Goyvaerts, J., Levithan, S.: Regular Expressions Cookbook - Detailed Solutions in Eight Programming Languages, Second Edition. O'Reilly (2012)

[16] Han, J., Kamber, M.,Pei, J.:Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA(2011)

[17] Hanisch, D., Fundel, K., Mevissen, H., Zimmer, R., Fluck, J.: Prominer:rule-based protein and gene entity recognition. BMC Bioinformatics 6, 14 (2005)

[18] Henerey,R.: Classification. Chapter 2. pp.6–16.Machine Learning, Neural and Statistical Classification. Ellis Horwood. USA (1994)

[19] Hirschman, L., Morgan, A.A., Yeh, A.S.: Rutabaga by ant other name: extracting biological names. pp. 247–259. No. 35, Academic Press (2002)

[20] Hong, Y., Wang, X., Chen, Y., Wang, J., Zhang, T., Zheng, J., Yu, D., Li, Q., Zhang, B., Wang, H., Pan, X., Ji, H.: Rpi blender tac-kbp2014 knowledge base population system. In: Proceedings of the TAC-KBP 2014 Workshop (2014)

[21] Humphreys, K., Demetriou, G., Gaizauskas, R.: Two applications of information extraction to biological science journal articles: Enzyme interactions and protein structures. In: Proceedings of the Pacific Symposium on Biocomputing (PSB-2000). pp. 505–516 (2000)

[22] Indurkhya, N., Damerau, F.J.: Handbook of Natural Language Processing. Chapman & Hall/CRC, 2nd edn. (2010)

[23] Jiang, J.: Information extraction from text. In: Mining Text Data, pp. 11–41 (2012)

[24] Kaufmann, M., Gaizauskas, R., Wakao, T., Humphreys, K., Cunningham, H., Wilks, Y.: University of sheffield: Description of the lasie system as used for muc-6 (1995)

[25] Koning, D., Sarkar, I., Moritz, T.: Taxongrab: Extracting taxonomic names from text. Biodiversity Informatics 2 (2005)

[26] Krallingera, M. , Erhardtb, R. and Valencia, A.: Text-mining approaches in molecular biology and biomedicine. Drug Discovery Today. 10, p. 439–445 (2005)

[27] Lee Epstein, G.K.: The rules of inference. The University of Chicago Law Review 69, 1–133 (2002)

[28] Linné, C.v., Salvius, L.: Caroli linnaei...systema naturae per regna tria naturae :secundum classes, ordines, genera, species, cum characteribus, differentiis, synonymis, locis. 1, 881 (1758)

[29] Mallory, E., Zhang, C., Ré, C., Altman, R.B.: Large-scale extraction of gene interactions from full text literature using deepdive. BMC Bioinformatics pp. 1–8 (2015)

[30] Min, B., Grishman, R.: Challenges in the knowledge base population slot filling task. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12). European Language Resources Association (ELRA) (2012)

[31] Mintz, M., Bills, S.,Snow, R., Jurafsky, D.: Distant Supervision for Relation Extraction Without Labeled Data. Association for Computational Linguistics. Stroudsburg, PA, USA (2009)

[32] Nguyen, T.H., He, Y., Pershina, M., Li, X., Grishman, R.: New york university 2014 knowledge base population systems. In: Proceedings of the TAC-KBP 2014 Workshop (2014)

[33] Pazienza, M.T. (ed.): Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology, International Summer School, Lecture Notes in Computer Science, vol. 1299. Springer (1997)

[34] Peters, S., Zhang, C., Livny, M., Ré, C.: A machine-compiled macroevolutionary history of phanerozoic life. CoRR abs/1406.2963, pp.671–683 (2014)

[35] Piskorski, J., Yangarber, R.: Information extraction: Past, present and future. In: 2º (ed.) Multisource, multilingual information extraction and summarization, theory and applications of natural language processing. Springer (2013)

[36] Ramos, J.: Using TF-IDF to Determine Word Relevance in Document Queries(1999) http://www.cs.rutgers.edu/ mlittman/courses/ml03/iCML03/papers/ramos.pdf

[37] Richardson, M., Domingos, P.: Markov logic networks. Mach. Learn. 62, pp.107–136 (2006)

[38] Roth, B., Barth, T., Chrupala, G., Gropp, M., Klakow, D.: Relationfactory: A fast, modular and effective system for knowledge base population. In: Proceedings of the 14th Conference of the European Chapter of the Association or Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden. pp. 89–92 (2014)

[39] Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson Education, 2 edn. (2003)

[40] Sarawagi, S.: Information Extraction, vol. 1. Now Publishers Inc., Hanover, MA, USA (2008)

[41] Shalev-Shwartz, S., Ben-David, S.: Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, New York, NY, USA (2014)

[42] Smola, A.J., Vishwanathan, S.: Introduction to Machine Learning. Cambridge University Press (2008)

[43] Surdeanu, M., Heng, J.: Overview of the english slot filling track at the tac2014 knowledge base population evaluation. In: Proceedings of the TAC-KBP 2014 Workshop (2014)

[44] Surdeanu, M., Ji, H.: Overview of the english slot filling track at the tac2014 knowledge base population evaluation. In: Proceedings of the TAC KBP 2014 workshop (2015)

[45] Tang, J., Alelyani, S., Liu, H.: Feature Selection for Classification: A Review. Data Mining and Knowledge Discovery Series (2014)

[46] Thessen, A.E., Cui, H., Mozzherin, D.: Applications of natural language processing in biodiversity science. Advances in Bioinformatics(2012)

[47] Thessen, A.E., Parr, C.S.: Knowledge extraction and semantic annotation of text from the encyclopedia of life. vol. 9, p. e89550. Public Library of Science (2014)

[48] Rückstieß, T., Osendorfer, C., van der Smagt, P.: Sequential Feature Selection for Classification. In: AI 2011: Advances in Artificial Intelligence: 24th Australasian Joint Conference, Perth, Australia, December 5-8, 2011. Proceedings, pp.132–141 , Springer Berlin Heidelberg (2011)

[49] Vibhav, G., Dechter, R.: SampleSearch: A scheme that searches for consistent samples. In Proceedings of AISTATS (2007)

[50] Witten, I. H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems). Morgan Kaufmann Publishers Inc. San Francisco, CA, USA (2005)

[51] Zhai, C.,Lafferty, J.: A Study of Smoothing Methods for Language Models Applied to Information Retrieval. ACM Trans. Inf. Syst.New York, NY, USA (2004)

[52] Tilman, D.: Causes, consequences and ethics of biodiversity. Nature 405, 208–211 (2000)

[53] Zhang, C.: DeepDive: A Data Management System forAutomatic Knowledge Base Construction. Ph.D. thesis, University of Wisconsin-Madison

[54] Zhang, C., Govindaraju, V., Borchardt, J., Foltz, T., Ré, C., Peters, S.: Geodeepdive: statistical inference using familiar data-processing languages. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013. pp. 993–996 (2013)

[55] Zhou, X., Zhang, X., Hu, X.: Maxmatcher: Biological concept extraction using approximate dictionary lookup. In: PRICAI. vol. 4099, pp. 1145–1149. Springer (2006)