ESCOLA POLITÉCNICA DA
UNIVERSIDADE DE SÃO PAULO
Engenharia Mecatrônica

GABRIEL AUGUSTO BIANCHI AZEVEDO FERREIRA
THIAGO FERNANDES FERRAZ

Processamento de Linguagem Natural para Consultas em Bancos de Dados Bibliográficos

São Paulo
2018

ESCOLA POLITÉCNICA DA
UNIVERSIDADE DE SÃO PAULO
Engenharia Mecatrônica

GABRIEL AUGUSTO BIANCHI AZEVEDO FERREIRA
THIAGO FERNANDES FERRAZ

Processamento de Linguagem Natural para Consultas em Bancos de Dados Bibliográficos

Monografia apresentada ao Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos da Escola Politécnica da Universidade de São Paulo como requisito obrigatório à obtenção do título de Engenheiro Mecatrônico.

Orientador:
Prof. Dr. Fábio G. Cozman

São Paulo
2018

ESCOLA POLITÉCNICA DA
UNIVERSIDADE DE SÃO PAULO
Engenharia Mecatrônica

GABRIEL AUGUSTO BIANCHI AZEVEDO FERREIRA
THIAGO FERNANDES FERRAZ

Processamento de Linguagem Natural para Consultas em Bancos de Dados Bibliográficos

Monografia apresentada ao Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos da Escola Politécnica da Universidade de São Paulo como requisito obrigatório à obtenção do título de Engenheiro Mecatrônico.

Aprovada em novembro de 2018.

_____

Orientador:
Prof. Dr. Fábio G. Cozman

Catalogação-na-publicação

FERREIRA, Gabriel
Processamento de Linguagem Natural para Consultas em Bancos de Dados Bibliográficos / G. FERREIRA, T. FERRAZ -- São Paulo, 2018.
71 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1.APRENDIZADO COMPUTACIONAL 2.PROCESSAMENTO DE LINGUAGEM NATURAL 3.ONTOLOGIAS 4.MOTORES DE BUSCA I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos II.t. III.FERRAZ, Thiago

AGRADECIMENTOS

Aos nossos familiares, estimados amigos, e a todos os que acreditaram que esta caminhada seria possível.

Resumo

Este trabalho busca unir técnicas de processamento de linguagem natural e ontologias para construir um mecanismo de busca de textos científicos em bases textuais bibliográficas do domínio específico de Engenharia de Petróleo. Um modelo de ontologia para descrever fichas catalográficas foi concebido, com o objetivo de estruturar a organização das fichas. Além disso, propomos uma classificação das frases dos resumos das fichas catalográficas, de modo a enriquecer a quantidade de informações da base de dados e facilitar a recuperação de informações. As frases dos resumos são categorizadas em quatro subcategorias que refletem a estrutura do discurso científico (Introdução, Objetivos, Métodos e Resultados/Conclusões). Uma base de dados contendo 278 artigos da área de Engenharia de Petróleo foi reunida e, as frases de seus resumos foram rotuladas manualmente para servirem como base de treino e validação para nossos algoritmos. No total, o *dataset (*que será disponibilizado publicamente*)* compreende mais de 2200 frases categorizadas. Nos experimentos, na base de teste, nosso melhor classificador foi a Regressão Logística e obteve uma acurácia de 86,4% com um F1-Score variando de 0,81 a 0,87, nas diversas classes. O sistema de buscas foi avaliado e obteve-se um mAP de 0,80 em um conjunto de consultas criadas para nossa base.

Palavras-chave: Processamento de Linguagem Natural, Aprendizado de Máquina, Busca de Documentos, Ontologias

Abstract

The present work aims at combining natural language processing techniques with ontologies in order to build a search engine for scholarly documents from the specific domain of Petroleum Engineering. We have created a model of ontology that describes the catalog records of scholarly documents (such as scientific papers, thesis, etc.). Besides, we propose a sentence classifier that will classify the sentences from the abstract of the catalog records. Our goal is to provide more data to the search engine and enhance data retrieval. The sentences are classified into four classes that reflect the scientific discourse structure (Introduction, Goals, Methods and Results/Conclusion). We produced a dataset containing 278 scholarly articles in the field of Petroleum Engineering. More than 2,200 sentences were labeled in order to be used as training and testing data. This dataset will be made publicly available. The classifier that yielded best results was Logistic Regression, with an accuracy of 86.4%, together with an F1-Score ranging from 0.81 to 0.87. The information retrieval system built on top of the classification system yielded a mAP of 0.80. We used queries constructed for our dataset in order to assess the precision.

Keywords: Natural Language Processing, Machine Learning, Document Retrieval, Ontologies

SUMÁRIO

1 - Introdução

Como defendido por Singh e Solanki (2016), vivemos atualmente num período de rápido desenvolvimento computacional, o que, com a ajuda da *web* e de bibliotecas digitais, impulsiona a geração e armazenamento de documentos textuais dos mais diversos tipos. Para que seja possível controlar e administrar esses repositórios textuais, é essencial o desenvolvimento e aprimoramento dos sistemas de recuperação de informação (Ladeira e Alvarenga, 2012). Um exemplo que ilustra essa situação são os repositórios de artigos científicos disponíveis *online*, que são cada vez mais frequentes e tornam-se cada vez maiores.

Em 2014, estimava-se que havia mais de 114 milhões de artigos científicos disponíveis na *web*, apenas na língua inglesa (Khabsa, Madian e Giles, 2014). Nesse sentido, cada vez mais, buscam-se desenvolver e aprimorar sistemas de busca de documentos dentro deste domínio. Do mesmo modo, cresce cada vez mais a disponibilidade de documentos no formato *machine-readable,* ou seja, documentos anotados de forma a permitir que um *software* os interprete sintática e semanticamente (Constantin et al., 2016). A construção desse tipo de documento está frequentemente associada ao uso de ontologias.

Neste trabalho apresentamos um sistema de buscas para a recuperação de artigos científicos, da área de Engenharia de Petróleo, que extrai informações não apenas do título ou palavras-chave, mas, que utiliza informações textuais presentes nos resumos (*abstracts)* das fichas catalográficas, explorando inclusive elementos semânticos das frases. O sistema categoriza as frases do resumo e permite ao usuário buscar termos ou assuntos dentro dos Objetivos, Resultados ou Métodos dos trabalhos. A escolha de um domínio específico para o desenvolvimento do sistema se deu com o intuito da limitação do vocabulário empregado nos documentos textuais e, logo, da aplicação de técnicas de processamento de linguagem natural de maneira eficiente sem a necessidade de montagem de uma base de dados demasiadamente extensa.

Para que isso seja possível, desenvolveu-se um classificador de frases, que, de maneira automática, classifica cada frase do resumo dos artigos em diferentes

categorias: 1) Introdução/Contextualização; 2) Objetivos; 3) Métodos; e 4) Resultados/Conclusões. O sistema de classificação de frases gera metadados que podem ser utilizados como complementos de informação para sistemas de busca, de recuperação de informações, de sumarização de documentos, interpretação de textos ou tomada de decisões. Dada a importância desta etapa de categorização e as numerosas aplicações que dela decorrem, este trabalho se foca no sistema de classificação de sentenças.

Com a finalidade de organizar os artigos em um banco de dados, cria-se uma ontologia para descrever suas fichas catalográficas. As ontologias nos ajudam a organizar o conhecimento, uma vez que sua principal função é identificar classes de objetos e a forma como estas se relacionam num domínio específico (Guarino et al., 2009). Em outras palavras, ontologias se prestam ao papel de organizar certo domínio de conhecimento de forma estruturada em classes hierárquicas, indicando as possíveis relações entre as entidades, bem como as características de cada classe. A associação de ontologias a bancos de dados textuais traz inúmeros benefícios ao processamento e recuperação de dados, uma vez que permite a estruturação destes em formatos computacionalmente tratáveis.

Nossa ontologia para descrição das ficha catalográficas possui dois níveis: um estrutural - que contém a essência da organização de uma ficha catalográfica - e, um nível semântico que aborda o conteúdo das frases e sentido das frases (categorização).

1.1 - Objetivos do trabalho

Os objetivos principais deste trabalho são:

a) Desenvolver uma estrutura eficiente para a descrição e organização de fichas catalográficas que atenda a nossos propósitos, bem como escolher uma ontologia semântica adequada para a classificação das frases de resumos de artigos científicos;
b) Criar uma base de dados de fichas catalográficas de artigos da área de Engenharia de Petróleo e anotar as frases dos resumos nas categorias escolhidas, gerando assim um conjunto de treinamento/testes;

c) Criar um classificador que permita a classificação automática das frases dos resumos, de modo a gerar metadados e informações adicionais a sistemas de busca ou recuperação de informações.
d) Implementar um sistema de busca de artigos científicos que utilize informações das frases dos resumos e se aproveite das categorizações das sentenças.

1.2 - Organização da monografia

Esta monografia organiza-se do seguinte modo: no Capítulo 2, apresenta-se uma revisão bibliográfica dos principais assuntos abordados por este trabalho, bem como aplicações e soluções já existentes e o “Estado da Arte”. O Capítulo 2 tem o objetivo de fornecer subsídios teóricos que permitam ao leitor acompanhar o restante do texto. No Capítulo 3, discute-se a organização do banco de dados de artigos científicos, bem como o desenvolvimento da ontologia utilizada, além disso dedica-se um espaço a abordar modelos de organização do discurso científico e de categorização de frases. Também no Capítulo 3, apresentam-se os métodos utilizados para a população do banco de dados: a seleção dos artigos e anotação das frases para a criação de um conjunto de treinamento e testes. No Capítulo 4, apresentam-se os experimentos de classificação de frases realizados, bem como seus resultados. O Capítulo 5 trata do desenvolvimento do sistema de buscas e recuperação de informações.

2 - Revisão bibliográfica

Neste capítulo, apresentamos uma revisão bibliográfica de temas centrais e caros ao nosso trabalho, de modo a proporcionar ao leitor leigo na área um entendimento básico dos assuntos necessários à compreensão desta monografia. A leitura deste capítulo pode ser feita antes da leitura do restante do texto, ou, conforme necessidade, enquanto o leitor avança nos demais capítulos. Em todo caso, este capítulo servirá como uma referência teórica, a ser consultada para garantir o bom entendimento do que vem a seguir.

Na Seção 2.1, fazemos uma rápida apresentação sobre a área de Processamento de Linguagem Natural. Na Seção 2.2 discutimos técnicas e métodos de classificação textual, bem como apresentamos alguns classificadores comumente utilizados em tarefas de classificação de textos. Os classificadores abordados são utilizados neste trabalho para os experimentos de classificação de frases, apresentados no Capítulo 4. Na Seção 2.3 conceituamos, de maneira breve, as Ontologias, e discutimos seu papel no processamento de textos. Na Seção 2.4, revisamos alguns trabalhos anteriores de classificação de frases de artigos científicos, discutindo técnicas utilizadas e resultados obtidos. A Seção 2.5 se dedica a explorar os conceitos de Recuperação de Informações (*Information Retrieval)*, e também Modelos de Linguagem, utilizados em nosso sistema de busca. Na Seção 2.6, fazemos uma breve definição de “Representações Vetoriais de Palavras”, apresentando um método consagrado. Tal método será utilizado mais adiante, no Capítulo 5. Dessa forma, é recomendado que se revisite a Seção 2.6 durante a leitura do Capítulo 5.

2.1 - Processamento de Linguagem Natural

De acordo com Nadkarni et al. (2011), o Processamento de Linguagem Natural, ou, em inglês, *Natural Language Processing* (NLP), iniciou-se na década de 1950, como uma intersecção entre as áreas de Inteligência Artificial e linguística. Diferenciava-se, à época, da área de Recuperação de Informação, ou Information Retrieval (IR), que empregava, em larga escala, métodos estatísticos para indexação

e buscas de grandes volumes de texto de maneira eficiente. Por muito tempo o NLP baseou-se em regras lógicas criadas manualmente, que levavam em conta esquemas gramaticais, sintáticos e usavam expressões regulares (RegEx) para buscas e classificações. Com o tempo, no entanto, ambas as áreas (IR e NLP) foram se aproximando, e o Processamento de Linguagem Natural baseado em regras criadas manualmente perdeu força, enquanto que as análises baseadas em Aprendizado de Máquina e estatística tornavam-se mais importantes e mostravam-se bem-sucedidas (Nadkarni et al., 2011).

No contexto do processamento de linguagem natural, “Análise Semântica” pode ser definida como o processo de mapear frases em linguagem natural para uma representação formal de seu significado passível de interpretação por uma máquina (Luz e Finger, 2017). Em outras palavras, é uma transformação que permite converter uma frase comum em uma representação matemática adequada, que capture seu significado e relações entre as entidades.

Certas abordagens tradicionais de NLP utilizam-se, muitas vezes, de modelos construídos manualmente, esquemas léxicos de alta qualidade e características linguísticas que são, via de regra, específicos a um certo tipo de domínio ou representação. (Dong e Lapata 2016). Esse tipo de especificidade pode tornar essas estratégias pouco adaptáveis a diferentes aplicações, ou seja, muitas vezes esses métodos são desenvolvidos para operarem em um contexto específico e controlado, limitados a um vocabulário pré-determinado de entrada.

Um exemplo de aplicação que utiliza o método supracitado foi desenvolvido por Singh e Solanki (2016), com o objetivo de traduzir frases em linguagem natural para consultas SQL. Esse sistema recebe uma consulta em linguagem natural e a processa seguindo os seguintes passos: 1) conversão do texto para caixa baixa; 2) separação da frase em palavras (*tokens);* 3) Remoção de *stop words* (palavras que não adicionam sentido); 4) Classificação dos *tokens* em substantivos, pronomes, verbos e variáveis numéricas / *strings*; 5) Determinação das relações entre as palavras, utilizando-se da classificação feita em 4; 6) Geração da query em linguagem SQL.

Como já explicado, esse tipo de procedimento é dependente do contexto e aplicação. Para a execução do passo 4, por exemplo, os autores criaram tabelas de

substantivos, pronomes e verbos que poderiam estar presentes nas consultas considerando-se o banco de dados que era objeto das consultas. O passo 5, por outro lado, utiliza-se de métodos estatísticos.

Há trabalhos na área de processamento de linguagem, como o de Luz e Finger (2017), em que os modelos linguísticos se apoiam exclusivamente em métodos de aprendizado de máquina, sem a dependência da elaboração de regras manualmente. Os autores treinaram redes neurais para traduzir frases do inglês para a linguagem SPARQL de consulta de bancos de dados. Nesse caso, não há dependência de regras léxico-linguísticas criadas manualmente, ou outros tipos de estruturas complexas propostas de forma manual. Em vez disso, essas abordagens se apoiam em uma base de dados com exemplos já previamente classificados ou anotados. Os métodos baseados em Aprendizado de Máquina possuem o intuito de eliminar a etapa de elaboração de estruturas para modelagem da linguagem. Esses modelos são aprendidos pelos algoritmos a partir dos dados.

2.2 - Sistemas de Classificação de textos

De acordo com Jurafsky e Martin (2019, no prelo), a classificação/categorização textual é uma tarefa que pode ser utilizada em uma gama grande de aplicações de processamento de linguagem natural. Seu objetivo é ser capaz de atribuir uma classe dentre um grupo discreto de classes para uma certa observação textual, a partir da extração de variáveis úteis para a realização dessa tarefa. Para isso, pode-se utilizar um conjunto de regras elaboradas à mão, o que tende a tornar o classificador frágil, à medida que as mudanças nos dados de entrada com o tempo não sejam contempladas pelas regras. Outra alternativa é a utilização de técnicas de aprendizado de máquina supervisionado, visando a aprendizagem do mapeamento de novas observações nas saídas corretas (classes), a partir de uma base de dados previamente rotulada. Ou seja, a partir de uma entrada x de *features* observadas e um conjunto fixo de classes de saída $Y = y_1, y_2, \ldots, y_M$ pré-estabelecido, deve-se retornar uma classe predita: $y \in Y$. Na seção seguinte discutiremos a escolha das *features* para processos de classificação de textos com mais detalhes.

Atualmente, métodos de classificação de textos são explorados em tarefas como a análise de sentimentos (*sentiment analysis*), que consiste da extração de sentimentos transmitidos pelo autor na escrita de, por exemplo, críticas literárias, textos políticos, comentários em redes sociais etc. Pak e Paroubek (2010) criaram um classificador de sentimentos baseados em frases publicadas no *microblog Twitter.* Para essa tarefa, utilizaram o classificador Naive Bayes e, como *features,* a *bag-of-words* (esses conceitos e ferramentas serão explicados mais adiante) para classificar as frases em três categorias: Positivas, Negativas ou Neutras. Outra aplicação comum da categorização textual é na detecção de spam (*spam detection*), que consiste da tarefa de classificação binária de e-mails entre as classes *spam* ou *não-spam* (Jurafsky e Martin, 2019, no prelo).

2.2.1 Features comumente usadas para classificação de textos

2.2.1.1 - Bag-of-Words

Para a aplicação dos métodos de aprendizado de máquina citados na seção anterior, primeiramente é preciso extrair variáveis (*features*) mensuráveis das observações (textos, frases, etc) em linguagem natural. Uma maneira simples de se fazer isso, é através da representação dos documentos textuais no formato *bag-of-words*, ou seja, como um conjunto desordenado de palavras, das quais extrai-se apenas a sua frequência de aparição no documento, ignorando suas respectivas posições (Jurafsky e Martin, 2019, no prelo).

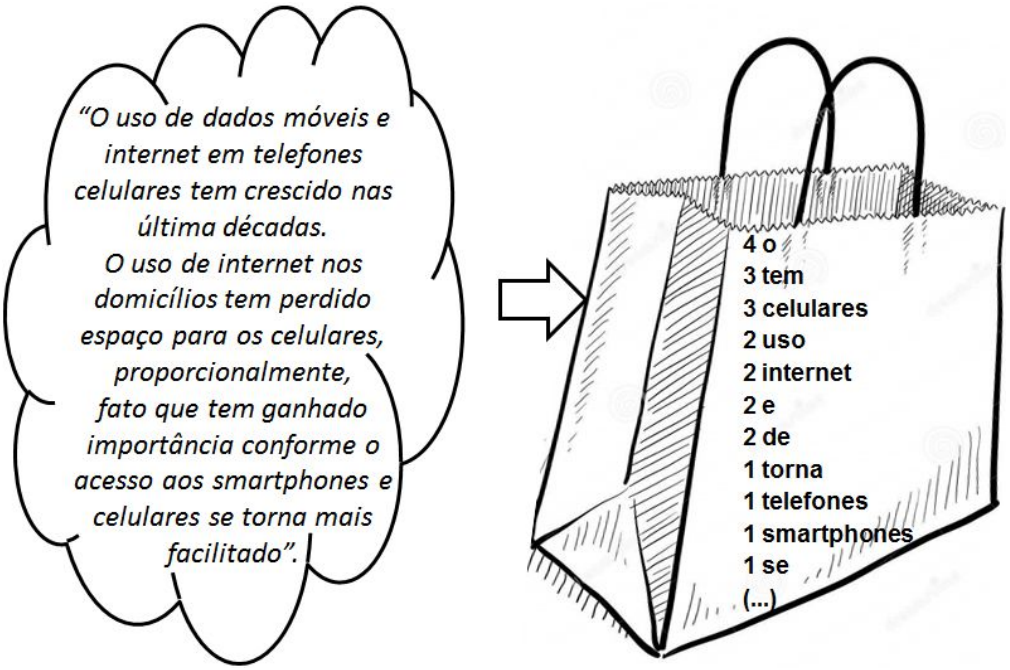
Dessa forma, o resultado dessa representação de documentos textuais pressupõe que o efeito de cada palavra na classificação é o mesmo sendo essa a primeira, décima ou última palavra do documento, ou seja, a sua posição não é considerada como uma variável importante mediante a aparição ou não da palavra no documento. A Figura 1 mostra um exemplo de extração de *features* de um excerto textual, utilizando *bag-of-words*. O resultado é uma relação das palavras que aparecem no texto, acompanhadas de sua frequência.

Em algumas aplicações, pode-se utilizar todas as palavras do texto neste processo, como ocorreu no exemplo da Figura 1. Em certas situações, contudo, deseja-se remover algumas palavras do texto antes da formação da *bag-of-words.* Muitas vezes, deseja-se retirar aquelas palavras que não adicionam sentido a frase,

conhecidas como *stop words.* São palavras como artigos, preposições, conjunções (“a”, “o”, “as”, “os”, “na”, “no”, “e”, etc.) e que servem de auxiliares na construção das frases, não representando, portanto, sua ideia central. Caso a hipótese seja de que esse tipo de palavra não auxiliará na classificação das frases, pode-se optar por sua remoção. Nesse caso, trata-se de um pré-processamento, anterior à extração das *features.*

As palavras que constam do modelo *bag-of-words* podem ser chamadas de vocabulário.

Figura 1 - Exemplo de construção de um conjunto bag-of-words a partir de um documento (frase ou conjunto de frases). À esquerda temos a frase original, e à direita, temos o saco (bag) de palavras, ou vocabulário, com suas respectivas frequências.



2.2.1.2 - N-gramas

Os N-gramas consistem de conjuntos de N palavras a serem usadas para formar o vocabulário do *bag-of-words.* No caso mais simples, como visto na seção anterior, temos os unigramas, ou seja, o vocabulário do *bag-of-words* é formado apenas por palavras individuais. Se estivéssemos utilizando bigramas (N=2), o

vocabulário seria composto de diversas sequências de duas palavras, como exemplificado na Tabela 1, abaixo:

Ao contrário das palavras avulsas (unigramas), os N-gramas (com N maior que 1) têm a capacidade de traduzir, de um certo modo, o contexto de ocorrência das palavras. Certas palavras quando próximas de outras transmitem ideias e sentidos completamente diferentes e, por isso, o uso de N-gramas pode ajudar substancialmente nas tarefas de classificação.

Tabela 1 - Exemplo de formação do vocabulário do bag-of-words utilizando-se de N-gramas de diferentes tamanhos. Os itens de cada vocabulário estão separados por ponto e vírgula.

Frase	As ações da BOVESPA caíram na manhã desta quarta-feira.
	Vocabulário
Unigrama	as; ações ; da ; BOVESPA ; caíram ; na ; manhã; desta; quarta-feira
Bigrama	as ações; ações da; da BOVESPA; BOVESPA caíram; caíram na; na manhã; manhã desta; desta quarta-feira;
Trigrama	as ações da; ações da BOVESPA; da BOVESPA caíram; BOVESPA caíram na; caíram na manhã ; na manhã desta; manhã desta quarta-feira

2.2.2 - Classificadores

Nessa Seção serão apresentados alguns classificadores comumente adotados no processo de classificação textual por métodos de aprendizado de máquina supervisionado e, que foram aplicados no presente trabalho, sendo as suas escolhas justificadas posteriormente.

2.2.2.1 - Naive Bayes

Como descrito por Jurafsky e Martin (2019, no prelo), o classificador Naive Bayes trata-se de um classificador probabilístico do tipo “Generativo” que, a partir de uma dada observação, retorna a classe mais provável de ter gerado tal conjunto de dados de entrada. Dessa forma, para uma observação de dados d, o classificador

retorna a classe $\hat{c}$ dentre todas as classes $c \in C$ que contenha a maior probabilidade a posteriori dado o documento, como indicado pela equação 2.1.

$$\hat{c} = \underset{c \in C}{argmax}\ P(c \mid d) \quad . \tag{2.1}$$

Utilizando-se a regra de Bayes descrita pela equação 2.2, é possível então escrever a equação 2.1 como mostrado pela equação 2.3.

$$P(x \mid y) = \frac{P(y \mid x)P(x)}{P(y)} \quad . \tag{2.2}$$

$$\hat{c} = \underset{c \in C}{argmax}\ P(c \mid d) = \underset{c \in C}{argmax}\ \frac{P(d \mid c)P(c)}{P(d)} \quad . \tag{2.3}$$

Dessa última, pode-se eliminar o denominador $P(d)$ uma vez que este não varia para cada classe $c \in C$, de forma que a classe retornada é a que maximiza o produto da probabilidade a priori $P(c)$ pela verossimilhança $P(d \mid c)$ como indicado pela equação 2.4.

$$\hat{c} = \underset{c \in C}{argmax}\ P(d \mid c)P(c) \quad . \tag{2.4}$$

O classificar de Naive Bayes enfim se utiliza de uma suposição simplificadora para computar a verossimilhança. No contexto de processamento de linguagem natural, representando-se um documento textual por um conjunto de variáveis (*features*) $f_1, f_2, \ldots, f_n$, assume-se a independência entre as probabilidades $P(f_i \mid c)$ dada a classe c, o que resulta na equação 2.5 que enfim pode ser aplicada para o cálculo de $\hat{c}$ indicado na equação 2.6.

$$P(d \mid c) = P(f_1, f_2, ..., f_n \mid c) = P(f_1 \mid c) \cdot P(f_2 \mid c) \cdot ... \cdot P(f_n \mid c) \quad . \tag{2.5}$$

$$\hat{c} = \underset{c \in C}{argmax}\ P(c) \cdot \prod_{k=1}^{n} P(f_i \mid c) \quad . \tag{2.6}$$

Na classificação textual, utilizando-se como *features* uma *bag-of-words*, teríamos:

$$P(d \mid c) = P(w_1|c) \cdot P(w_2|c) \cdot (\ldots) \cdot P(w_n|c), \tag{2.7}$$

onde $w_1, w_2, ..., w_n$ representam as palavras que aparecem em *d*. As probabilidades $P(w_i \mid c)$ podem ser estimadas dos dados a partir do cálculo da seguinte razão:

$$P(w_i \mid c) = \frac{N(w_i, c)}{\sum_{j=1}^{n} N(w_j, c)}\ , \tag{2.8}$$

onde $N(w_i, c)$ indica a quantidade de vezes que a palavra w_i ocorre em uma determinada classe c. A suavização de Laplace (Laplace Smoothing) pode ser utilizada de modo a se evitar probabilidades iguais a zero. Ela pode ser vista, de uma perspectiva Bayesiana, como o valor esperado de uma distribuição a posteriori, utilizando-se uma distribuição de Dirichlet simétrica (Jurafsky e Martin, 2019, no prelo). Neste caso teremos:

$$P(w_i \mid c) = \frac{N(w_i, c) + 1}{\sum_{j=1}^{n} N(w_j, c) + V}\ , \tag{2.9}$$

em que V representa o tamanho do vocabulário. Com esse ajuste, nenhuma probabilidade resultará em zero.

2.2.2.2 - Regressão Logística

Regressão Logística consiste de um classificador probabilístico do tipo "Discriminativo", ou seja, que tenta aprender quais *features* dos dados de treino são mais úteis para a discriminação entre o conjunto de classes possível para cada observação (Jurafsky e Martin, 2019, no prelo). Em termos da equação 2.1 apresentada na seção anterior, pode-se distinguir esse tipo de classificador dos

classificadores “Generativos”, por tentarem diretamente estimar as probabilidades a posteriori $P(c \mid d)$, em vez de se utilizarem da verossimilhança como no caso do naive Bayes.

Também de acordo com Jurafsky e Martin (2019, no prelo), dado um conjunto binário de classes, e documentos textuais representados por vetores $x_i = [x_{i1}, x_{i2}, ..., x_{in}]$ de n *features*, deseja-se estimar a probabilidade de uma dada observação pertencer à classe objetivo, ou seja, $P(y = 1 \mid x_i)$. Isso é feito pela Regressão Logística através do aprendizado (na base de treino), utilizando-se o algoritmo de gradiente descendente estocástico (*stochastic gradient descent*) e a perda de entropia cruzada (*cross-entropy loss*), de um vetor w de pesos e um termo b de viés que são utilizados em dados $x_j = [x_{j1}, x_{j2}, ..., x_{jn}]$ de teste como indicado pela equação 2.10.

$$z = \left(\sum_{k=1}^{n} w_k \, x_{jk} \right) + b \quad . \tag{2.10}$$

O termo z indicado acima pode variar na faixa: $]-\infty, \ +\infty[$, uma vez que os pesos w_k podem assumir quaisquer valores reais, desta forma, para que o valor computado seja compatível com uma probabilidade (a probabilidade $P(y = 1 \mid x_j)$ procurada), utiliza-se a função sigmóide (ou função logística) que mapeia valores reais para o intervalo $[0, \ 1]$ como mostrado pela equação 2.11.

$$P(y = 1|x_j) = \sigma(z) = \frac{1}{1 + e^{-z}} \quad . \tag{2.11}$$

A classificação então é realizada definindo-se um limite de decisão (usualmente 0.5) como descrito pela equação 2.12 em que $\hat{y}$ refere-se à classe predita pela Regressão Logística.

$$\hat{y} = \begin{cases} 1, \ se \ P(y = 1|x_j) > 0.5 \\ 0, \ se \ P(y = 1|x_j) \leq 0.5 \end{cases} \quad . \tag{2.12}$$

Para a realização do processo de classificação em um número $k > 2$ de classes, utiliza-se a Regressão Logística Multinomial, a qual estima para cada classe $c \in C$ a probabilidade $P(y = c \mid x_j)$. Isso é feito pelo cálculo de um vetor $z = [z_1, z_2, ..., z_k]$ onde cada termo é obtido de maneira semelhante ao indicado pela equação 2.10 e, pela aplicação do mesmo na função softmax (equação 2.13), que generaliza a função logística, recebendo um vetor com k valores arbitrários e mapeando-os em uma distribuição de probabilidades com cada termo pertencente ao intervalo $[0, 1]$ e, cuja soma é igual a 1.

$$softmax(z) = \left[\frac{e^{z_1}}{\sum_{i=1}^{k} e^{z_i}}, \frac{e^{z_2}}{\sum_{i=1}^{k} e^{z_i}}, \ldots, \frac{e^{z_k}}{\sum_{i=1}^{k} e^{z_i}} \right]. \quad (2.13)$$

Por fim, o cálculo de $P(y = c \mid x_j)$ é feito como apresentado pela equação 2.14 em que cada termo w_i e b_i são referentes à, respectivamente, um vetor de pesos e um viés ajustados na base de treino para cada classe e, o produto $w_i \,.\, x_j$ é detalhado na equação 2.15.

$$P(y = c|x_j) = \frac{e^{w_c \cdot x_j + b_c}}{\sum_{i=1}^{k} e^{w_i \cdot x_j + b_i}}, \quad (2.14)$$

$$w_i \cdot x_j = \sum_{l=1}^{n} w_{il}\ x_{jl} \quad . \quad (2.15)$$

2.2.2.3 - SVM

Desenvolvidos por Cortes e Vapnik (1995), Support Vector Machines (SVMs) tratam de métodos de classificação que se baseiam no cálculo de hiperplanos que maximizem a separação entre os dados (de treino) em um espaço dimensional superior utilizado por uma função Φ para o mapeamento dos pontos (x_i, y_i), $i = 1, ..., l$, onde $x_i \in R^n$ representa o vetor de *n features* da i-ésima observação da base de treino e, $y_i \in \{1, -1\}$ sua respectiva classe (Paula e Bonatti, 2015).

Isso é feito resolvendo-se o seguinte problema de otimização (equação 2.16), que está sujeito às restrições apresentadas pelas inequações 2.17 e 2.18 (Hsu et al., 2003):

$$min_{w,b,\varepsilon} \quad \frac{1}{2}w^T w + C\sum_{i=1}^{l}\varepsilon_i \quad , \tag{2.16}$$

$$y_i(w^T\phi(x_i) + b) \geq 1 - \varepsilon_i, \tag{2.17}$$

$$\varepsilon_i \geq 0, \tag{2.18}$$

onde w é o vetor normal ao hiperplano, b determina o afastamento do hiperplano da origem, $C > 0$ é um termo de regularização (penalização) que impõe um peso à minimização dos erros no conjunto de treinamento e, ε_i refere-se ao grau de erro de classificação de x_i. Como mostrado por Boser et al. (1992) pode-se então utilizar uma função de kernel: $K(x_i, x_j) = \Phi(x_i)^T\Phi(x_j)$ para computar o produto escalar entre x_i e x_j, sendo os kernels mais utilizados na literatura apresentados abaixo (Hsu et al., 2003):

- Linear: $K(x_i, x_j) = x_i^T x_j$;
- Polinomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \ \gamma > 0$;
- RBF (Radial Basis Function): $K(x_i, x_j) = exp\left(-\gamma \left\|x_i - x_j\right\|^2\right), \ \gamma > 0$;
- Sigmoidal: $K(x_i, x_j) = tanh(\gamma x_i^T x_j + r)$.

Nos quais γ, r e d são parâmetros dos kernels.

2.2.3 - Seleção de Features

Quando lida-se com um grande número de *features*, a seleção de subconjuntos de *features* é um tópico de extrema relevância na área de aprendizado

de máquina, uma vez que a redução dimensional dos dados pode contribuir positivamente em tarefas de classificação (Yun et al., 2007). Isso acontece pois, a seleção de variáveis, que sigam determinados critérios, tende a reduzir o custo do aprendizado, além de remover dados ruidosos, irrelevantes e/ou redundantes, provendo melhores acurácias (Tu et al., 2007).

Existe uma gama variada de funções de avaliação que podem ser computadas para o procedimento de seleção de *features* tais quais: o Ganho de Informação (*Information Gain*), a Frequência do Termo (*Term Frequency*), o Qui-quadrado de Pearson (*Pearson Chi-square*), a razão de possibilidades (*odd ratio*), o coeficiente de Gini (*Gini coefficient / index*), a Informação Mútua (*Mutual Information*), a Frequência nos Documentos (*Document Frequency*), dentre outras (Elssied et al., 2014).

2.2.3.1 - Valor-p baseado na estatística F

Elssied et al. (2014) descreve um método para seleção de *features* no qual, para cada variável X contínua realiza-se um teste F de ANOVA que verifica se as amostras de X para cada rótulo de Y possuem a mesma média.

Dessa forma, para um conjunto de N observações classificadas em um número J de classes tem-se:

- N_j = O número de observações em que $Y = j$.
- $\bar{x}_j$ = A média da amostra da variável X em que o rótulo $Y = j$.
- s_j^2 = A variância da amostra da variável X em que o rótulo $Y = j$:

$$s_j^2 = \sum_{i=1}^{N_j} \frac{(x_{ij} - \bar{x}_j)^2}{N_j - 1} \quad . \tag{2.19}$$

- $\bar{\bar{x}}$ = A média das médias das amostras da variável X pertencentes a cada uma das classes de Y:

$$\bar{\bar{x}} = \sum_{j=1}^{J} \frac{N_j \bar{x}_j}{N} \quad . \tag{2.20}$$

Pode-se então computar o Valor-p com base na estatística F pela seguinte equação:

$$Valor - p = Prob\{F(J-1, N-J) > F\}, \tag{2.21}$$

onde F é calculado como mostrado pela equação 2.22 e, $F(J-1,\ N-J)$ é uma variável aleatória que segue uma distribuição F de Fisher-Snedecor com J-1 e N-J graus de liberdade.

$$F = \frac{\sum_{j=1}^{J} \frac{N_j(\bar{x}_j - \bar{x})^2}{J-1}}{\sum_{j=1}^{J} \frac{(N_j-1)s_j^2}{N-1}}. \tag{2.22}$$

Por fim, pode-se ranquear as variáveis X de acordo com o Valor-p obtido em ordem crescente, ou mesmo de acordo com o valor F calculado em ordem decrescente (em caso de empate na ordenação pelo Valor-p).

2.2.4 - Avaliação de Classificadores

A avaliação empírica tem papel central em sistemas de aprendizado de máquina, processamento de linguagem natural e recuperação de informação (*information retrieval*), nos quais usualmente se utilizam indicadores unidimensionais sintéticos para estimações de performance (Goutte e Gaussier, 2005).

Os principais indicadores calculados nesse processo, quando trata-se de classificadores, são os seguintes: Acurácia, Precisão, Recall e F1-Score, cujas definições são apresentadas nos itens 2.2.4.1 a 2.2.4.4. A Figura 2 mostra uma matriz de confusão, que é uma forma de se visualizar graficamente *True Positives* (TP), *False Positives* (FP), *True Negatives* (TN) e *False Negatives* (FN). Estes indicam, para um processo supervisionado de classificação:

- TP (verdadeiros-positivos): a quantidade de atribuições de uma classe correta y a observações pertencentes a essa mesma classe ;
- FP (falsos-positivos): a quantidade de atribuições de uma classe y a observações não pertencentes a essa classe ;

- TN (verdadeiros-negativos): a quantidade de atribuições de classes diferentes de y a observações não pertencentes a y;
- FN (falsos-negativos): a quantidade de atribuições de classes diferentes de y a observações pertencentes a y.

Figura 2 - Matriz de confusão para classificação de base de treino em classes y ou não y.

CLASSE PREVISTA		CLASSE REAL	
		Classe y	**Classe Não y**
	Classe y	True Positives (TP)	False Positives (FP)
	Classe Não y	False Negatives (FN)	True Negatives (TN)

2.2.4.1 - Acurácia

A acurácia mede o total de acertos de classificação sobre o total de observações classificadas pelo classificador, ou seja, a proporção total de acertos, como mostrado pela equação 2.23.

$$Acurácia = \frac{TP + TN}{TP + FP + TN + FN}. \tag{2.23}$$

2.2.4.2 - Precisão

A precisão consiste da fração das atribuições a uma determinada classe que, de fato, estão corretas, medindo dessa forma a certeza de que a atribuição dessa classe é feita da maneira correta pelo classificador, ignorando a perda de informação, ou seja, as atribuições de outras classes às observações (Paula e Bonatti, 2015).

$$Precisão = \frac{TP}{TP + FP}. \tag{2.24}$$

2.2.4.3 - Recall

O *recall* (ou sensibilidade) avalia a capacidade de um classificador de não perder informações relevantes, ou seja, sua habilidade de conseguir classificar corretamente todas as observações pertencentes a uma determinada classe, aceitando-se a atribuição da mesma às observações não pertinentes.

$$Recall = \frac{TP}{TP + FN}. \tag{2.25}$$

2.2.4.4 - F1-Score

O F1-Score trata-se de uma medida que leva em consideração tanto a precisão quanto o *recall* ao mesmo tempo e, com igual peso, sendo descrita pela média harmônica desses dois índices, como mostrado pela equação 2.26 (onde Pr refere-se à precisão e, Rec ao Recall).

$$F1 - Score = \frac{2PrRec}{Pr + Rec}. \tag{2.26}$$

2.3 - Ontologias

As ontologias se prestam a organizar o conhecimento, identificando classes de objetos e a forma como estas se relacionam num domínio específico (Guarino et al., 2009). Por meio delas, o conhecimento pode ser estruturado em classes hierárquicas, em que existem relações entre as entidades, e definem-se características de cada classe. A associação de ontologias a bancos de dados textuais traz inúmeros benefícios ao processamento e recuperação de dados, uma vez que permite a estruturação destes em formatos computacionalmente tratáveis.

Pode-se classificar ontologias em leves ou pesadas (Furtado, 2017), ou seja, em duas grandes categorias relativas à maneira com que descrevem o domínio a que são aplicadas. As ontologias leves se propõe a apresentar estruturas mais gerais para o mapeamento de seus domínios, resultando em um número de classes menor e relações mais simples entre essas, uma vez que não apresentam um nível de detalhamento tão profundo. Já as ontologias pesadas tendem a mapear domínios

muito específicos de maneira detalhada, apresentando em sua estrutura uma grande quantidade de classes e relações entre as mesmas.

A necessidade de utilização de cada um dos tipos de ontologia supracitados varia de acordo com a aplicação. Na estratégia de associação entre ontologias e técnicas de processamento de linguagem natural baseados em aprendizado de máquina, o uso de ontologias mais pesadas necessita de uma quantidade maior de dados anotados, o que tende a resultar em resultados piores que o uso de ontologias leves e abrangentes (Furtado, 2017).

2.3.1 - Ontologias para descrição de documentos científicos

"Publicação Semântica" pode ser definida como a atividade de melhorar um documento (por exemplo um artigo científico), por meio de anotações semânticas, provendo um meio de se entender o significado das informações publicadas e permitir a criação de vínculos com outros documentos relacionados (Ruiz-Iniesta e Corcho, 2014). Esse tipo de prática pode ser realizada apoiando-se em ontologias, que permitem descrever a estrutura dos documentos, bem como certas camadas de seu conteúdo, metadados, relações e informações relacionadas. Nesta seção fazemos uma breve revisão bibliográfica de outros trabalhos que se propuseram a apresentar ontologias para a descrição de trabalhos científicos, bem como para a estruturação do discurso científico. Também nesta seção apresentamos uma proposta de ontologia estrutural adequada a ser usada em nosso trabalho.

De acordo com Constantin et al. (2016), a estruturação dos textos científicos segundo uma ontologia pode auxiliar na compreensão dos documentos em questão, tanto por parte dos usuários, como por máquinas, além de evidenciar um critério de extrema relevância para a avaliação de contribuições científicas que é a organização coerente da narrativa textual descrita.

Ruiz-Iniesta e Corcho (2014) relatam que, em trabalhos anteriores, na produção de ontologias para a descrição de documentos escolares e científicos, a organização das classes, muitas vezes, abrange as grandes seções da documentação, como por exemplo na DoCO (Constantin et al., 2016) em que aparecem classes como "seção", "capítulo" e "glossário", ou na BIBO (*Bibliographic Ontology Specification*) em que se podem encontrar as classes "artigo acadêmico",

“livro” e “capítulo” dentre sua organização hierárquica. Classes como essas são muitos úteis quando se está observando e categorizando artigos e livros científicos completos. São, contudo, menos úteis quando se olha apenas para o corpo do texto ou para o resumo.

Ainda dentro da revisão de publicações, feita por Ruiz-Iniesta e Corcho (2014), percebe-se que existem ontologias que, tentam descrever a estruturação do discurso científico. Essas ontologias se estendem até o nível de parágrafos, frases ou sentenças, classificando-as em subcategorias. Ontologias como a Deo (Shotton e Perroni, 2015), a CISP (Liakata e Soldatova, 2007) e a EXPO (Soldatova, King e Clare, 2007) encaixam-se nessa categoria, apresentando um elevado nível de detalhamento, por se tratarem de estruturas que visam a ser aplicadas às documentações completas. Esse nível de minúcia pode ser utilizado no presente trabalho para o detalhamento da maior fonte de informação das fichas catalográficas: o resumo. A utilização de todas as classes de qualquer uma das ontologias citadas anteriormente é aplicável, porém, tornaria a nossa ontologia muito pesada para a aplicação em um extrato de texto tão reduzido. Dessa forma propõe-se, no Capítulo 3, a redução do número de classes para a organização do discurso científico dentro dos resumos das fichas catalográficas.

2.4 - Trabalhos anteriores sobre classificação de frases em artigos

A classificação de frases é uma tarefa que pode beneficiar muitos outros processos de mineração de textos, isso porque ela é capaz de gerar dados úteis e computacionalmente tratáveis, os quais podem ser utilizados em outras etapas, como por exemplo extração de informações, sumarização e resposta automática a perguntas (Agarwal e Yu, 2009). Há na literatura diversos trabalhos que tratam do problema de classificação automática de frases em artigos científicos. Nesse sentido, há um padrão muito utilizado conhecido como IMRAD para a categorização das frases do discurso científico, esse padrão compreende às classes: Introdução, Métodos, Resultados e Discussão.

Muitos trabalhos dessa área se concentram em textos das áreas médicas e biológicas, como por exemplo o trabalho de Agarwal e Yu (2009), que rotularam manualmente frases de artigos da base PUBMED nas categorias do IMRAD com o

objetivo de treinar algoritmos de aprendizado computacional para classificá-las. As anotações para a formação da base de dados foram feitas por duas pessoas de maneira independente e a cada frase foi também associada uma nota que era a confiança que se tinha naquela anotação. A matriz de confusão foi gerada e as frases em que não houve concordância de anotação foram descartadas. Segundo os autores, a maior parte das discordâncias era causada por frases ambíguas, em que não fica muito claro a que classe pertencem, ou frases que poderiam pertencer a mais de uma categoria.

Como métodos de aprendizado de máquina, Agarwal e Yu (2009) testaram os seguintes métodos: multinomial Naive Bayes, Naive Bayes e SVM, sendo que o primeiro foi o que deu os melhores resultados. No caso do multinomial Naive Bayes, os autores utilizaram como *features* a *bag-of-words*, sem a remoção de *stop-words,* uma vez que notaram que elas são importantes para distinguir certas categorias de outras. Além disso, os autores ainda adicionaram *features* binárias como: a presença ou não de citações na frase e a presença ou não de verbos no presente ou passado (o que, segundo eles, ajudava a distinguir as frases de introdução e resultados). Os autores reportaram uma acurácia de 91,25%.

Outros autores se dedicaram a mesma tarefa de classificação automática de frases, porém aplicadas especialmente à categorização de frases de *abstracts* de trabalhos científicos (que é o foco desta monografia), como por exemplo McKnight e Srinivasan (2003) ou Yamamoto e Takagi (2005). Os primeiros relataram uma acurácia de 89,2% e um F-score variando de 52-86% usando SVMs e classificando as frases segundo o padrão IMRAD. Os segundos relataram um F-Score variando de 73-89%. Esses autores notaram que o uso de *feature* de posição para cada frase (que avalia a localização relativa de uma frase dentro do resumo) pode auxiliar na classificação, uma vez que muitos *abstracts* são estruturados e seguem uma ordem lógica (por exemplo, as conclusões vêm normalmente depois dos métodos, e assim por diante).

2.5 - Recuperação de Informações e Modelos de Linguagem

Esta Seção apresenta conceitos básicos da área de Recuperação de Informações, bem como introduz o tópico de Modelos de Linguagem e Suavização

de Probabilidades. Estas técnicas serão utilizadas no Capítulo 5, que trata do sistema de buscas implementado.

De acordo com Manning, Raghavan e Schütze (2008), Recuperação de Informações (*Information Retrieval*) trata-se de encontrar (recuperar) material (geralmente documentos) de natureza não estruturada e que satisfaçam a uma necessidade ou restrição informacional. Comumente esses dados não estruturados estão em grandes coleções ou bancos de dados, usualmente mantidos em computadores.

A partir desta definição proposta pelos autores supracitados, cabe-nos fazer algumas observações pertinentes. Por dados “não estruturados”, entende-se dados que não estão numa forma prontamente “acessível” aos programas de computador ou, em outras palavras, dados que não sejam prontamente computacionalmente tratáveis. Textos são bons exemplos de dados não estruturados: é possível de se extrair informação útil deles, no entanto, processamentos devem ser realizados anteriormente. Bancos de dados relacionais são, por outro lado, exemplos de dados bem estruturados, em que as informações já estão prontas para serem consultadas através de consultas padronizadas.

Para nossas finalidades, vamos trabalhar com uma definição de “Documento” simples, porém efetiva: trata-se de um texto, composto por palavras, símbolos ou números e que pode ter qualquer tamanho/extensão. No caso de nosso trabalho, os documentos são os resumos dos artigos científicos ou as frases dos resumos.

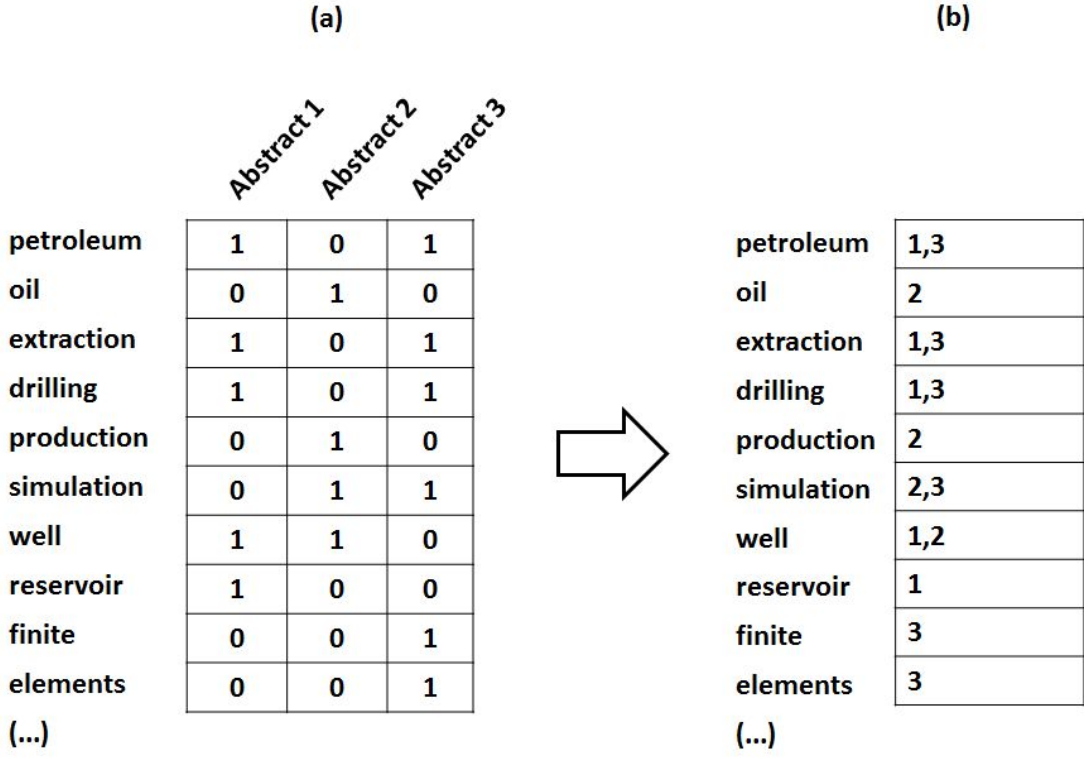
2.5.1 - Busca Booleana

Por “Restrição” ou “Necessidade Informacional” entende-se aquilo que se deseja buscar, ou as informações que se deseja recuperar. Em outras palavras, é a especificação da consulta. Tal especificação pode ser feita de diversos modos. A forma mais básica de recuperar informações – e especificar as Restrições - é a Booleana, ou Lógica (Manning, Raghavan e Schütze, 2008), em que uma lista de termos é escolhida partindo-se de um vocabulário pré-estabelecido. Estes termos são ligados por conjunções e predicados lógicos (“e”, “ou”, “não”, etc.) e retornam-se os documentos que satisfizerem tais condições. Para realizar tal consulta, é importante que haja uma matriz de Termos do Vocabulário vs. Documentos, que

indique os termos que estão presentes ou não em cada um dos documentos da base (Figura 3.a). Para economia de espaço e memória, pode-se representar essa matriz esparsa na forma invertida: para cada termo, grava-se em qual(is) documentos ele aparece (Figura 3.b).

Seguindo o exemplo da Figura 3, suponha, que a consulta a esse sistema de representação Booleana seja: `"drilling" E "extraction"`. Os documentos que satisfizerem essa condição serão retornados como relevantes (Abstract 1 e Abstract 3, no caso). Se, por outro lado a expressão booleana fosse `("oil" OU "petroleum") E ("simulation")` os resultados seriam Abstract 2 e Abstract 3.

Figura 3 – (a) Esquema mostrando uma matriz de Vocabulário vs. Documentos como exemplo. Os documentos são Abstracts (colunas), e a esquerda vemos os termos do vocabulário (linhas). "1" indica presença do termo no respectivo Abstract e "0" indica ausência; (b) Esquema mostrando a representação vetorial desta matriz, como forma de economizar-se espaço e memória: grava-se, para cada termo, o número dos documentos em que ele está presente.



2.5.2 - Buscas usando *scoring*

Consultas binárias não necessariamente retornam bons resultados, por alguns motivos:

a) não capturam o contexto ou ideia dos termos buscados: apenas "combinações exatas" são consideradas;

b) não é capaz de ordenar os resultados, pois não atribui uma pontuação (*score)* para os resultados a fim de diferenciar os mais relevantes dos menos relevantes.

Para endereçar a questão b), Manning, Raghavan e Schütze (2008) defendem a definição de um indicador de relevância para a busca, de modo que este seja capaz de indicar a "similaridade" entre uma consulta (*query)* e um documento, e não apenas retornar Verdadeiro ou Falso. Ora, se dispusermos de uma pontuação para os resultados, podemos ordená-los e mostrar aos usuários apenas os N resultados mais relevantes.

Uma forma de se atribuir uma pontuação (*scoring)* aos resultados das consultas é utilizando-se de um modelo probabilístico e calcular qual a probabilidade de um certo documento ser relevante ou estar relacionado à busca especificada. Na busca probabilística, dada uma condição ou *consulta (query)* **q***,* e um documento **D**, associa-se a este par (consulta, documento) uma probabilidade de que o documento D seja relevante, dada a consulta. Em vez de atribuirmos um valor binário, como na busca Booleana, busca-se um valor de probabilidades que nos ajude a ordenar os resultados, distinguindo os menos relevantes daqueles mais relevantes.

2.5.3 - Modelos de Linguagem

Uma das maneiras mais simples de se realizar este tipo de consulta é utilizando-se um Modelo de Linguagem de unigramas, como o apresentado por Zhai e Lafferty (2004). Neste tipo de abordagem, estimamos a probabilidade de um documento, dadas as palavras da busca, a partir da frequência de ocorrência das palavras no documento:

$$P(D|q) = \frac{P(q|D) \cdot P(D)}{P(q)}, \tag{2.27}$$

onde **D** é um documento e **q** é a consulta (conjunto de palavras). $P(D)$ pode ser assumido uniforme em toda a base (o que significa dizer que todos os documentos têm a mesma relevância à priori). Do mesmo modo, para fins de ranqueamento, ao compararmos diversos documentos, $P(q)$ se manterá constante, já que independe de **D**. Assim, temos que:

$$P(D|q) \propto P(q|D). \tag{2.28}$$

Também de acordo com Zhai e Lafferty (2004) a probabilidade da consulta **q** (composta de palavras $w_1, ..., w_n$), dado o documento, pode ser modelada por um Modelo de Linguagem de unigrama:

$$P(q|D) = \prod_{i}^{n} P(w_i|D), \tag{2.29}$$

$$P(q|D) = P(w_1|D) \cdot P(w_2|D) \cdot (\dots) \cdot P(w_n|D). \tag{2.30}$$

Sendo o modelo escolhido o de unigramas - que considera apenas a frequência das palavras individualmente, independente de sua ordem -, o estimador de máxima verossimilhança para $P(w \mid D)$ é a frequência da palavra w no documento D. Tal frequência é definida como a função *tf (Term Frequency),* ou frequência do termo, que é a divisão entre o número de vezes que a palavra aparece no documento D pelo total de palavras deste:

$$\hat{P}(w|D) = tf(w, D) = \frac{N(w, D)}{\sum_{w' \in D} N(w', D)}, \tag{2.31}$$

em que *N(w, D)* indica o número de ocorrências da palavra *w* no documento *D*. Este estimador, no entanto, não dá conta dos casos em que a palavra *w* está ausente no documento D: neste caso, a probabilidade $P(w \mid D)$ será zero. Consequentemente, toda a expressão de $P(D \mid q)$ será zerada, na presença de um termo igual a zero (vide eq. 2.29). Este fato é usualmente contornado utilizando-se uma forma de "suavização" (*smoothing)* das probabilidades. Usualmente, o estimador de máxima verossimilhança $\widehat{P}$ (eq. 2.31) subestimará a probabilidade de uma palavra que não esteja presente no documento. A suavização tem como principal propósito atribuir probabilidade não nula a tais palavras (ausentes do modelo), bem como melhorar a acurácia da estimação da probabilidade de palavras em geral (Zhai e Lafferty, 2004).

O método de suavização mais simples é o de Laplace, que já foi visto em seção anterior (vide Equação 2.8). Trata-se de uma correção que soma uma ocorrência a todas as frequências, evitando probabilidades iguais a zero. Outros métodos de suavização mais sofisticados, comumente substituem a frequência de palavras não observadas em um certo documento D por sua frequência em toda a coleção de documentos C (Chen e Goodman, 1998). Um exemplo é o método Jelinek–Mercer proposto por Jelinek e Mercer (1980):

$$P_\lambda(w|D) = (1-\lambda)P(w|D) + \lambda P(w|C). \qquad (2.32)$$

Este método de suavização estima a probabilidade $P(w \mid D)$ como sendo uma combinação linear entre a frequência da palavra w no documento D e na coleção C. O método possui um parâmetro λ que deve ser ajustado empiricamente.

2.6 - Representação de Palavras no Espaço Vetorial

Esta Seção visa a introduzir o conceito de representação vetorial de palavras. Essa técnica será utilizada posteriormente, no Capítulo 5, no processo de recuperação de informações. Caso o leitor deseje, poderá ler esta Seção após a leitura do Capítulo 5, para complementar seu entendimento.

A chamada representação Word2Vec (Mikolov et al., 2013a) tem sido muito utilizada em aplicações de NLP recentemente. Trata-se de uma forma de construir

uma representação vetorial para palavras, de modo que seja possível calcular distâncias e inferir a similaridade entre termos e palavras. Parte-se de um *corpus*, que é um conjunto de textos ou fragmentos de textos. O conjunto de todas as palavras distintas que aparecem no corpus constituem um vocabulário. Constrói-se uma rede neural (Figura 4) para realizar a seguinte tarefa: dada uma palavra de entrada (*input word*), a saída deve ser a probabilidade de que cada uma das outras palavras do vocabulário ocorra “próximo” da palavra de entrada em uma frase. Essa proximidade é determinada pela escolha de um tamanho de vizinhança (usualmente 5 palavras adiante e 5 palavras anteriores). Para essa tarefa, a rede é treinada no *corpus*, de modo que aprenda as relações de vizinhança entre palavras. Por exemplo, se considerarmos um conjunto de textos que contenham notícias da época da Guerra Fria, dada a palavra “Soviética” de entrada, a palavra “União” deverá ter uma probabilidade grande de ocorrer próximo desta, enquanto que a palavra “Americano” uma probabilidade menor.

Suponha um corpus com 10.000 palavras em seu vocabulário. A entrada e saída da rede são representadas por um vetor de 10 mil posições, em que cada posição refere-se a uma palavra. Na entrada, o número 1 em certa posição indicará que aquele vetor corresponde à representação daquela palavra, e todas as outras posições do vetor de entrada conterão zeros (ver Figura 4). Na saída, cada posição terá um valor em ponto flutuante, representando a probabilidade de ocorrência daquela palavra em uma vizinhança em torno da *input word.*

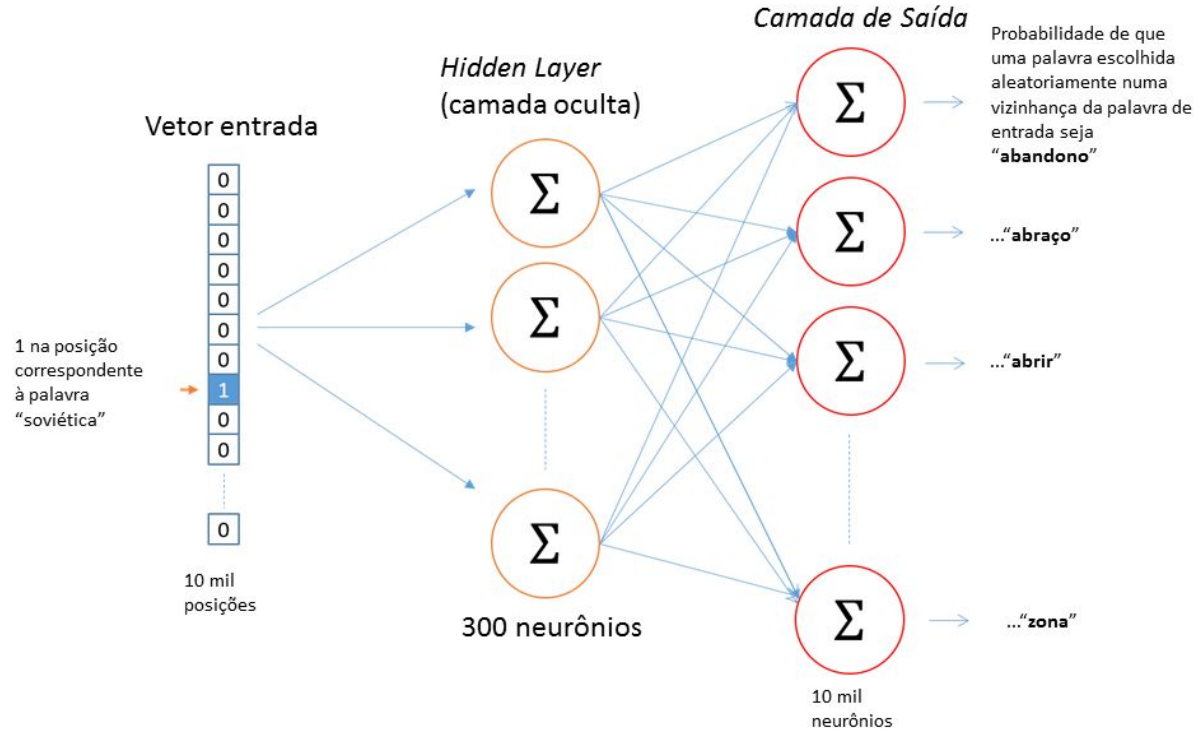
Essa rede é construída e treinada para realizar tais estimações, mas aquilo que nos interessa verdadeiramente é a representação intermediária de cada palavra na camada oculta (Hidden Layer), vide Figura 4. Essa representação é aprendida pela rede no processo de treinamento. No caso da figura, temos uma rede com N = 300 neurônios na camada oculta. Essa camada oculta nada mais é do que uma forma de representação intermediária da palavra de entrada. Ao criar uma rede e “designá-la” certa tarefa através do treinamento, ela deve aprender a representar as palavras de entrada em um espaço vetorial de dimensão N, onde N é o número de neurônios da camada oculta. A ideia deste processo é análoga (mas não idêntica) ao processo de *encoding-decoding,* explicado na seção anterior.

Esta representação intermediária é a chamada representação vetorial das palavras, daí o nome: “Word-to-Vec(tor)”. A partir daí, é possível de se criar um dicionário (*embedding)*, o qual relaciona cada palavra do vocabulário a um vetor em um espaço real N-dimensional. No caso da figura, N = 300, e, logo, cada palavra seria representada por um vetor de dimensão 300.

Mais formalmente, sendo **V** o tamanho do vocabulário e **N** o número de neurônios, podemos dizer que a camada oculta é uma função *f* da palavra de entrada e escrever:

$$f : \mathbb{R}^V \rightarrow \mathbb{R}^N \ . \tag{2.33}$$

Figura 4 - Esquema mostrando a estrutura da rede neural e exemplo do seu funcionamento para uma rede com 300 neurônios na camada oculta. Neste caso, a representação vetorial das palavras pertencerá a um espaço vetorial de dimensão 300.



Fonte: Adaptado de McCormick (2016).

A representação vetorial supracitada captura regularidades sintáticas e semânticas que podem ser observadas através de operações matemáticas simples, Mikolov et al. (2013a), por exemplo, demonstra que essas regularidades entre as

representações vetoriais de palavras que compartilham alguma relação particular, são observadas como vetores constantes em casos como (sendo *vec(i)* a representação vetorial da palavra *i*):

$$vec(\text{"}apple\text{"}) - vec(\text{"}apples\text{"}) \approx vec(\text{"}car\text{"}) - vec(\text{"}cars\text{"}) \approx vec(\text{"}family\text{"}) - vec(\text{"}families\text{"}).$$

Outro exemplo trata da relação de gênero (masculino/feminino) que é automaticamente capturada pelo modelo e permite que operações como a citada abaixo sejam possíveis: $vec(\text{"}king\text{"}) - vec(\text{"}man\text{"}) + vec(\text{"}woman\text{"}) \approx vec(\text{"}queen\text{"}).$

O cálculo da "proximidade" entre duas palavras, ou "similaridade", é feito, usualmente, pela Similaridade de Cossenos, que pode ser definida como:

$$sim(w_1, w_2) = \frac{\vec{v}(w_1) \cdot \vec{v}(w_2)}{|\vec{v}(w_1)||\vec{v}(w_2)|}, \tag{2.34}$$

em que *v(w) = vec(w),* ou a representação vetorial da palavra. Este número varia entre 0 e 1, onde 1 indica palavras iguais e, quanto mais próximo de zero, menor a similaridade entre as palavras.

3 - Base de dados

Neste capítulo descrevemos a formação de nossa base de dados e os métodos que foram utilizados para a anotação das frases e para o controle de qualidade. Também detalhamos a escolha das categorias das frases e da estrutura utilizada para a organização dos dados.

Na Seção 3.1, apresentamos a forma de organização dos dados e as categorias de classificação das frases. Na Seção 3.2, discutimos a coleta de dados e seu processo de rotulação.

3.1 - Ontologia estrutural e semântica proposta

Para formular nossa ontologia estrutural de fichas catalográficas, nos apoiamos em diversos outros trabalhos (supracitados nas seções 2.3.1 e 2.4), que criaram estruturas para descrever trabalhos científicos e categorias para a classificação de suas frases. Tentou-se incorporar à nossa ontologia tanto o lado estrutural da descrição e organização da ficha catalográfica, quanto uma parte mais relacionada ao discurso científico, o que foi feito através da inclusão de subclasses de frases.

A partir de evidências da maioria dos repositórios bibliográficos consultados, identifica-se em uma ficha catalográfica típica as seguintes classes: “Autor”, “Área do conhecimento”, “Orientador”, “Título”, “Data de publicação”, “Resumo” e “Palavra-Chave”. Essas classes foram incorporadas à nossa ontologia.

Além disso, nos preocupamos em descrever com mais detalhes o conteúdo da seção “resumo” das fichas catalográficas, dividindo-o em frases e classificando-as em subclasses. Para as subclasses, tomamos como base o padrão IMRAD, que contempla Introdução, Métodos, Resultados e Discussão. Tal padrão é utilizado por diversos autores na classificação de frases de trabalhos científicos, tal como visto na seção 2.4. A grande diferença é que a maioria destes trabalhos se propõe a classificar frases de textos científicos completos, ao passo que nós trabalhamos apenas com os resumos dos artigos. Dessa forma, ao observarmos diversos resumos de textos de engenharia, concluímos que, para nossos propósitos,

a seguinte divisão seria mais efetiva, sobretudo no que diz respeito à recuperação de dados relevantes: “Introdução/Revisão Bibliográfica”, “Objetivos”, “Métodos”, e “Conclusão/Resultados”. Em suma, retirou-se a classe “Discussão”, agrupando-a com “Conclusão/Resultados” e adicionou-se a classe “Objetivos”.

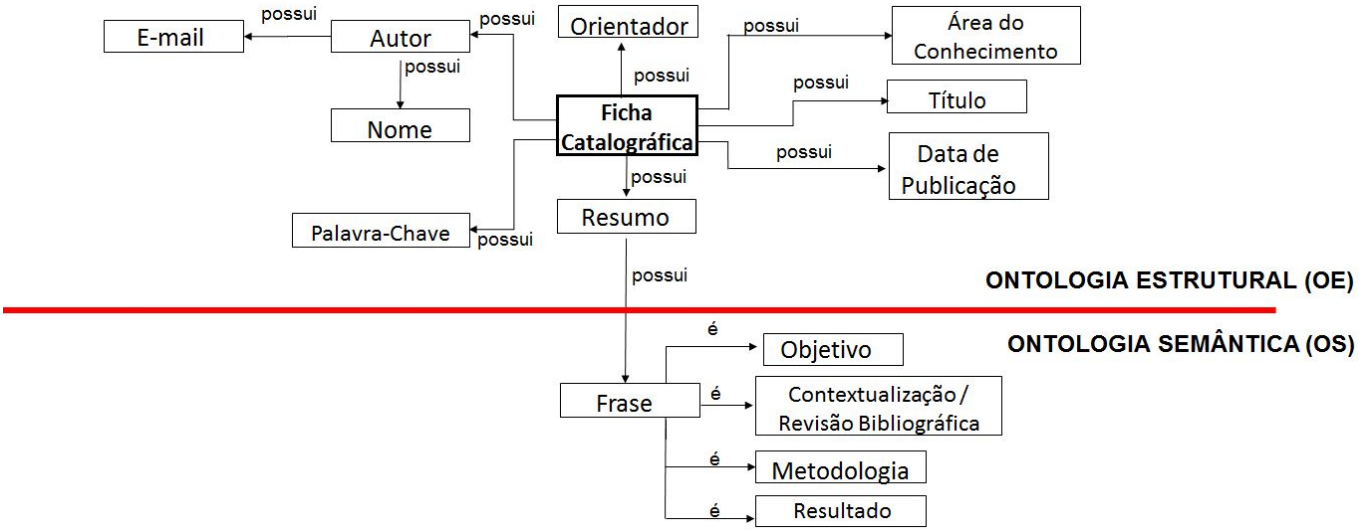
É muito relevante e, extremamente frequente, o aparecimento dos Objetivos do trabalho em um resumo. Por outro lado, um resumo, muitas vezes, não discute o tema, devido a uma restrição de espaço: ele deve ser enxuto. Por isso, a classe Discussão é menos importante e menos frequente nos resumos.

Temos dois níveis de informações em nossa base: uma Ontologia Estrutura (OE) e uma Ontologia Semântica (OS). A OE compreende a organização das fichas catalográficas, com seus dados cadastrais básicos: “Autor”, “Área do conhecimento”, “Título”, etc. Já a OS compreende a classificação das frases do *abstract,* e relaciona-se com o sentido do discurso e função das sentenças no texto, por isso, é chamada semântica. A Figura 5 ilustra essa organização.

3.2 - Coleta de dados e método de rotulação

Inicialmente coletamos 278 fichas catalográficas de artigos científicos, dentro da área de exploração de petróleo, extraídos da seguinte revista científica *Journal of Petroleum Exploration and Production Technology* (de 2011 até 2018) e também da base IEEE Xplore. Os resumos das fichas catalográficas foram separados em frases, totalizando 2420 frases. A separação foi realizada utilizando-se a biblioteca NLTK (*Natural Language Toolkit*) em Python (Bird et al., 2009). Todos os artigos e resumos coletados estavam em língua inglesa. Alguns resumos foram eliminados por estarem mal escritos (erros de escrita, de digitação gramática e confusos). Após a eliminação dos referidos resumos, restaram 2263 frases “válidas”, as quais foram classificadas de maneira independente por parte dos integrantes da dupla dentre as categorias da ontologia semântica proposta (“Introdução / Contextualização”, “Objetivo”, “Método”, “Resultados e Conclusões”).

Figura 5 - Diagrama mostrando a ontologia estrutural da ficha catalográfica desenvolvida para este trabalho, bem como sua relação com a ontologia semântica. A ontologia estrutural (OE) baseia-se em diversos trabalhos, que foram revisados na seção 2.3.1. A ontologia semântica (OS) baseia-se no padrão IMRAD, com algumas modificações.



Para o processo de anotação também foram estabelecidas algumas regras e orientações, de modo a se minimizar o efeito de frases ambíguas ou frases que poderiam ser classificadas em mais de uma categoria. A Tabela 2 mostra as principais características de cada classe, estabelecendo critérios e orientações seguidas pelos anotadores no processo de rotulação.

Outra prática adotada foi o estabelecimento da seguinte “prioridade” para as rotulações: primeiramente Objetivos, depois Resultados, então Métodos e, por fim, Introdução. Em outras palavras, caso uma frase se enquadre em duas categorias, deveria-se classificá-la de acordo com aquela que viesse em primeiro lugar na hierarquia estabelecida (vide Figura 6).

De acordo com esta hierarquia, caso um anotador estivesse em dúvida se certa frase deveria ser classificada como Introdução ou Métodos, por exemplo, deveria classificá-la como Métodos.

Figura 6 - Esquema mostrando a prioridade das classes uma sobre as outras.

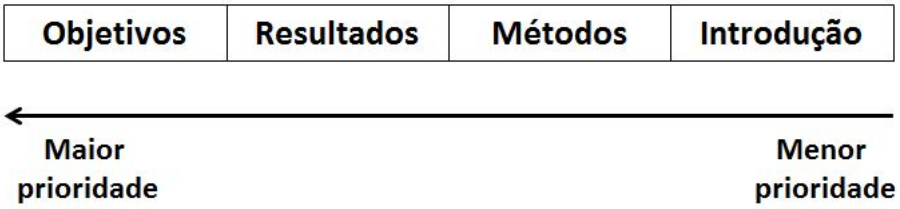


Tabela 2 - Descrição das classes utilizadas para a rotulação das frases dos resumos dos artigos.

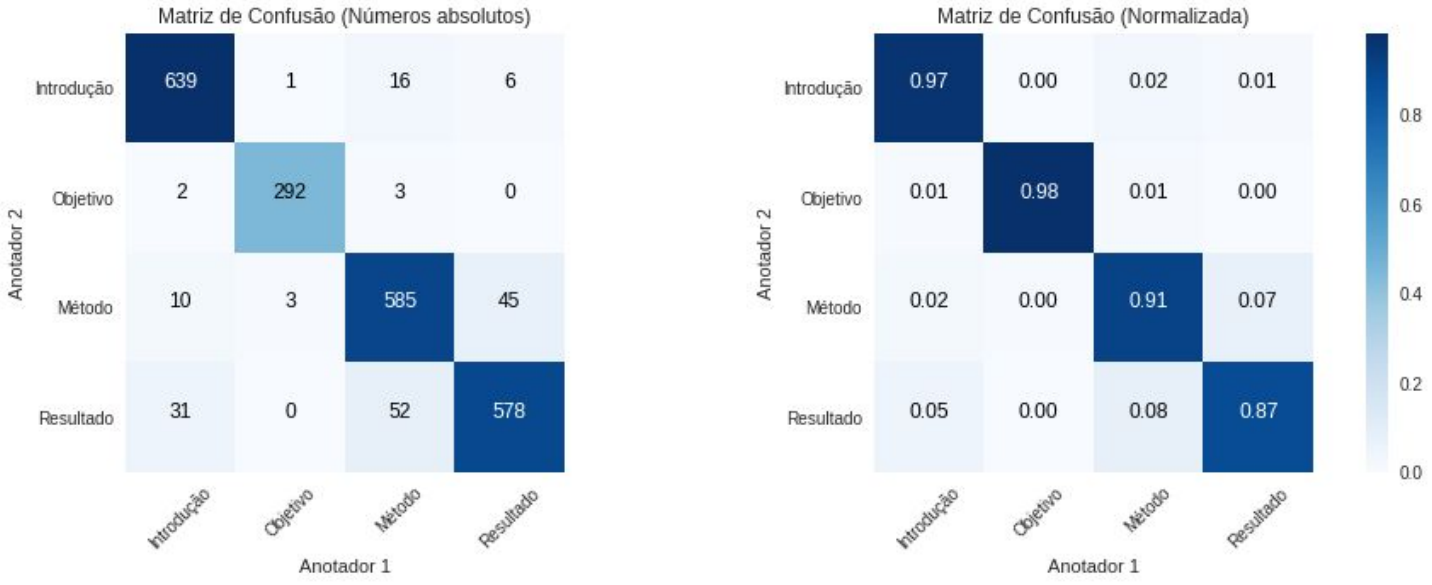
Classe	Descrição
Introdução	Inclui contextualizações sobre o tema do artigo, que, geralmente, são frases em sentido amplo e geral e, que não se referem explicitamente aos resultados do trabalho.
Objetivo	Explicita os objetivos do trabalho e as motivações que levaram os autores para a realização do trabalho. Inclui frases como "This work", "This thesis", "The purpose of this paper is..." mas não restrito a elas.
Método	Explicita os métodos utilizados para a realização da pesquisa: explicação de experimentos realizados pelos autores, ferramentas utilizadas, softwares, entre outros.
Resultado / Conclusão	Explicita os resultados obtidos através dos métodos propostos ou as conclusões tiradas a partir do estudo. Pode incluir dados numéricos de resultados ou não.

Tabela 3 - Matriz de confusão mostrando as anotações de cada anotador, e como essas anotações se confundem nas diferentes classes. A maior parte das confusões é causada por frases ambíguas.

		Anotador 1			
		Introdução	**Objetivo**	**Método**	**Resultado**
Anotador 2	**Introdução**	**639**	1	16	6
	Objetivo	2	**292**	3	0
	Método	10	3	**585**	45
	Resultado	31	0	52	**578**

Após a etapa de rotulação e cruzamento das anotações na matriz de confusão (Tabela 3 e Figura 7), de um total de 2263 frases válidas, 2094 tiveram coincidência na rotulação, o que corresponde a 92,5% das frases (diagonal principal da matriz de confusão). Portanto, houve 169 frases (7,5%) em que não houve coincidência: cada anotador classificou de uma maneira. Essas frases, devido à ambiguidade, foram retiradas da base. Por fim, nossa base rotulada de frases passou a contar com 2094 frases, todas anotadas de maneira igual por ambos os anotadores. A escolha de retirar tais frases foi feita de modo a se evitar trabalhar com frases ambíguas, e que pudessem atrapalhar o aprendizado dos classificadores. Agarwal e Yu (2009) obtiveram melhores resultados de classificação removendo as frases ambíguas de sua base, em seu trabalho de classificação de frases em artigos de ciências biológicas.

Figura 7 - Representações gráficas da matriz de confusão das rotulações (Anotador 1 x Anotador 2), em forma de números absolutos (à esquerda) e normalizada (à direita).



O Coeficiente kappa (Cohen's kappa coefficient ou kappa Score) é muito utilizado nas ciências sociais e médicas para medir a concordância entre dois “avaliadores” ou especialistas, sobretudo no que diz respeito a análises qualitativas e categóricas (Manning, Raghavan e Schütze, 2008). Ele é dito ser mais robusto que um simples cálculo de acurácia por levar em conta, também, a probabilidade de que

os juízes concordem por acaso. Por isso, o utilizamos como referência para ter um parâmetro para afirmar que nossa base era suficientemente não ambígua e que a tarefa de classificação poderia ser avaliada posteriormente de maneira confiável.

O coeficiente kappa pode ser calculado como segue (eq. 3.1):

$$\kappa = \frac{p_o - p_e}{1 - p_e}\ , \tag{3.1}$$

em que p_o indica a proporção de concordância entre os avaliadores e p_e é a probabilidade hipotética de que os avaliadores concordem entre si por acaso, calculada a partir dos dados.

De acordo com McHugh (2012), o coeficiente kappa deve ser interpretado como mostrado na Tabela 4.

Tabela 4 - Valores de referência para interpretação do coeficiente kappa.

Coeficiente kappa	**Nível de Concordância**	**% dos dados que são confiáveis**
0,0 - 0,20	Nenhum	0-4%
0,21 - 0,39	Mínimo	4-15%
0,40 - 0,59	Fraco	15-35%
0,60 - 0,79	Moderado	35-63%
0,80 - 0,90	Forte	64-81%
> 0,9	Quase perfeito	82-100%

Fonte: McHugh (2012).

O cálculo do coeficiente kappa em nossos dados resultou em um valor de 0,897, o que sugere que estamos na faixa de “forte nível de concordância”, muito próximos da faixa de concordância “quase perfeita”.

4 - Experimentos de classificação

Neste capítulo explicamos os experimentos realizados com relação ao classificador de frases. Exploraremos os métodos utilizados para a escolha das *features* e também os diferentes classificadores escolhidos para os testes.

4.1 - Conjuntos de treino e de teste

Para a realização experimentos, observou-se a separação dos dados disponíveis em dois conjuntos: um de treino e outro de testes. Essa separação tem objetivo de evitar-se o sobreajuste (*overfitting*) dos classificadores aos dados. A separação praticada foi de 80%-20% (treino-teste). Dessa forma, das 2094 frases não-ambíguas (vide seção 3.2), 1675 foram utilizadas para os treinos e 419 para os testes, todas selecionadas de forma aleatória.

A separação foi feita de modo estratificado, ou seja, de modo a manter as proporções entre as classes iguais tanto no conjunto de testes quanto no conjunto de treinamento. Isso é importante pois há uma classe que possui quantidade de observações significativamente menor (classe Objetivos), como pode-se ver na Tabela 5, que mostra a proporção dos dados em toda a base (Treino + Teste).

Tabela 5 – Proporção entre as classes nos dados de toda base (2094 frases não ambíguas).

Classe	Número Absoluto	Proporção (%)
Introdução	639	30,5
Objetivos	292	14
Métodos	585	28
Resultado	578	27,5
Total	2094	100%

A separação dos dados de forma estratificada serve ao propósito de preservar as proporções entre as classes de maneira aproximada, de modo que o processo de

treinamento não seja prejudicado e, também, para que não faltem exemplos suficientes das classes menos numerosas para a validação.

4.2 – Pré-processamento

O pré-processamento dos dados trata-se de um tratamento anterior à extração das *features.* Este tratamento contempla a remoção da pontuação das frases, a transformação das palavras para caixa baixa e a substituição dos números pela expressão "#ISNUMBER#", de modo que todos os números sejam contabilizados na mesma coluna do *bag-of-words.* Este último tratamento dos números é opcional, e foi realizado em alguns dos experimentos (naqueles em que há tal indicação). A troca dos números por essa expressão visa a criar uma *feature* que é a contagem da quantidade de números existentes na frase e verificar se ela é de fato um atributo relevante para o problema. Exemplos de pré-processamento estão presentes na Tabela 6.

Tabela 6 – Tabela que exibe exemplos de pré-processamento de frases, com remoção de pontuação, transformação das palavras para caixa-baixa e processamento de números.

Frase Original	Frase pré-processada
(I) "In this regard, RFID (Radio Frequency Identification) technology is currently considered as one of the leading enabling technologies of the Internet of Things (IoT)."	"in this regard rfid radio frequency identification technology is currently considered as one of the leading enabling technologies of the internet of things iot"
(II) "The measurement equipment transmits pseudo-random noise test signals from two antennas at a 2.47 GHz carrier with a signal bandwidth of approximately 25 MHz."	"the measurement equipment transmits pseudorandom noise test signals from two antennas at a **#ISNUMBER#** ghz carrier with a signal bandwidth of approximately **#ISNUMBER#** mhz"

Fontes das frases :

(I) HUANG, W. et al. Simultaneous coherent and random noise attenuation by morphological filtering with dual-directional structuring element. **IEEE Geosci. Remote Sens. Lett**, v. 14, n. 10, p. 1720-1724, 2017.

(II) GAAFAR, M.; MESSIER, G. G. Petroleum refinery multiantenna propagation measurements. **IEEE Antennas and Wireless Propagation Letters**, v. 15, p. 1365-1368, 2016.

4.3 – Features consideradas

Abaixo listamos as features que foram consideradas para nossos experimentos. Nos vários experimentos variamos os classificadores utilizados, bem como o conjunto de atributos escolhidos.

a) Bag-of-words e N-gramas

A *feature* de *bag-of-words* de N-gramas (vide seção 2.2) é um atributo base para nossas classificações. Ao longo dos experimentos, variamos o tamanho dos N-gramas (1 palavra, 2 palavras, 3 palavras, etc.), para verificar qual das combinações respondia com melhores resultados.

b) Posição

Este atributo corresponde à posição de uma determinada frase no *abstract.* Essa *feature* varia linearmente de 0 a 1, sendo que “0” indica que a frase é a primeira do resumo, enquanto “1” indica que é a última. Esta feature é importante pois, há uma certa correlação entre a posição da frase e sua categoria. Por exemplo, a introdução tende a vir antes dos métodos, e estes, por sua vez, antes das conclusões e resultados.

c) Frequência de PoS-tags (classes gramaticais)

PoS-tags são as *Part-of-Speech Tags*, ou seja, etiquetas que indicam as funções gramaticais das palavras nas frases (ex: verbo, advérbio, adjetivo, pronome, conjunção, substantivo, artigo, etc). Essa classificação é feita por uma ferramenta, que já está previamente treinada em uma grande base de dados da língua inglesa. Para nosso caso, utilizamos a ferramenta NLTK (Bird et al., 2009), que possui função para realizar a análise gramatical. De posse das *tags* de uma certa frase, fazemos a contagem da frequência de cada uma delas, por exemplo, *n* verbos no presente, *m* verbos no passado e *p* adjetivos. Essa contagem dá origem também a um conjunto de *features*.

4.4 - Classificadores utilizados e avaliação

Para os testes de classificação foram escolhidos três classificadores com base nas particularidades de nossa base de dados, bem como no estado da arte para o problema de classificação textual evidenciado no capítulo 2: Naive Bayes, SVM e Regressão Logística. Em cada teste foram computados os valores de acurácia, precisão, recall e F1-Score da aplicação dos mesmos no conjunto de treino descrito no item 4.1, afim da comparação de suas performances preditivas.

Para o classificador Naive Bayes optou-se pela configuração Multinomial, adequada para a utilização da frequência das palavras na representação de documentos textuais (Agarwal e Yu, 2009). Como demonstrado por McCallum e Nigam (1998), a configuração Multinomial supera a configuração que segue uma distribuição multivariada de Bernoulli, em que cada palavra é tida como uma variável booleana, ou seja, tendo em cada observação apenas a presença ou ausência das palavras contabilizadas.

Tendo sido aplicado pela primeira vez no contexto de classificação textual por Mosteller e Wallace (1964), o classificador Naive Bayes exige o ajuste de poucos parâmetros graças às simplificações de independência das variáveis dada a classe das observações a que pertencem, o que é de extrema utilidade na representação de documentos textuais por *bag-of-words*, por exemplo, uma vez que essas apresentam grandes quantidades de palavras, logo, *features*, que comumente iriam exigir o ajuste de uma proporção mais elevada de parâmetros. Apesar disso, tais suposições de independência condicional assumidas levam o classificador a apresentar desvantagens como a superestimação de certas evidências para casos em que os dados contenham variáveis extremamente correlacionadas. Mesmo assim, trabalhos como o de Wang e Manning (2012) mostram que para documentos textuais curtos (como é o caso das frases que desejamos classificar neste trabalho) o classificador Naive Bayes funciona muito bem.

A utilização de SVMs (Support Vector Machines) se deu por serem, assim como Naive Bayes, classificadores usualmente empregados para a obtenção de parâmetros base de desempenho em tarefas de classificação textual (Wang e Manning, 2012). Dessa forma, foi escolhido o *kernel* linear por apresentar performance superior a outras configurações em problemas onde o espaço dos

dados de entrada apresenta um número elevado de dimensões (Jochims, 1999), como é o caso da representação de frases por *bag-of-words*, em que cada palavra (dentro de um vasto vocabulário) é uma *feature.*

Por fim, avaliou-se a classificação com a aplicação de Regressão Logística que, assim como Naive Bayes, é um classificador probabilístico, no entanto apresentando diversas vantagens sobre esse, principalmente em face de variáveis com alta correlação entre si. Por exemplo, para duas *features* f_1 e f_2 perfeitamente correlacionadas, a regressão logística irá distribuir entre os termos w_1 e w_2 (que multiplicam as variáveis) o peso no processo de classificação, enquanto o classificador Naive Bayes as tratará de forma independente multiplicando as suas probabilidades (seus pesos) no cálculo de máxima verossimilhança (Jurafsky e Martin, 2019, no prelo).

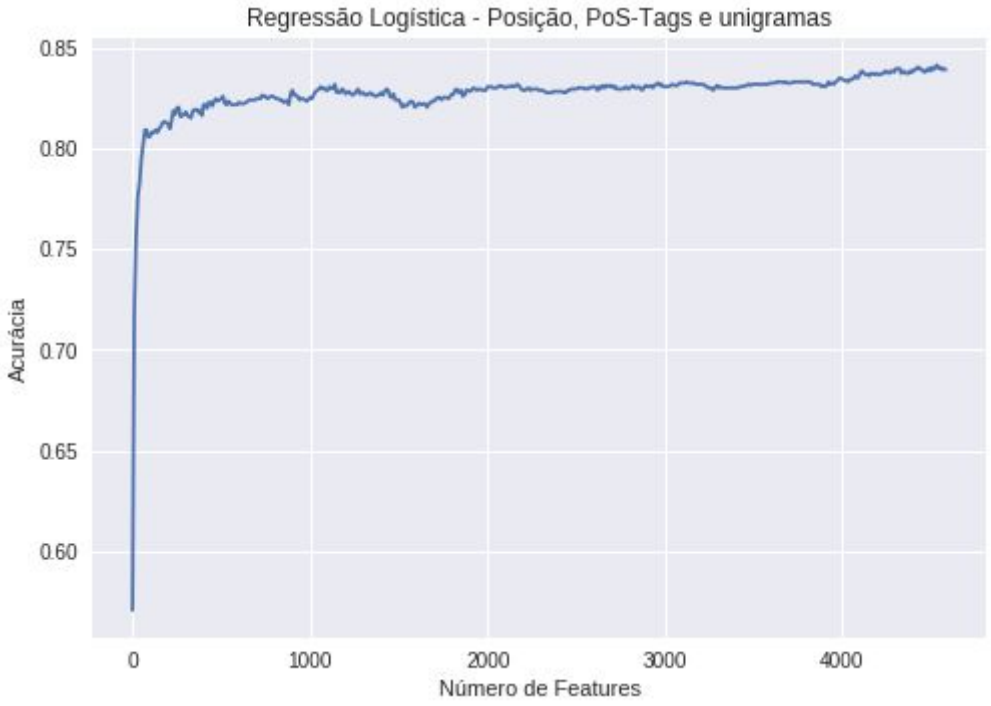
4.5 - Resultados dos testes de classificação

A maior parte dos classificadores utilizados têm, por construção, um funcionamento binário, ou seja, se prestam a resolver problemas onde existem duas classes: uma positiva e outra negativa. No caso de problemas multi-classe (em que existem mais de duas classes) uma das possibilidades é tratar a classificação com um esquema *One-vs-All (*um contra todos*)*. Ou seja, o classificador é treinado N vezes para as N classes, considerando, em cada um desses treinamento, uma classe específica como positiva e as outras N-1 classes como exemplos negativos. Dessa forma, obtemos N classificadores, cada um especializado em identificar uma classe específica. Esta é uma abordagem muito popular na literatura e, apesar de simples, obtêm relativo sucesso nas tarefas de classificação multi-classe (Galar et al., 2011). Dessa forma, essa foi nossa abordagem escolhida. Portanto, daqui em diante, quando se fala de classificação, entende-se que ela é feita por meio de um processo de *One-Vs-All.*

O Quadro 1 apresenta os valores de precisão, *recall*, *F1-Score* e acurácia obtidos pelos três classificadores indicados na seção 4.4 para 12 configurações das bases de treino e teste em relação à utilização da posição das frases, do pré-processamento de *strings* numéricas, das *PoS-Tags* e do número de n-gramas (apenas unigramas ou, unigramas e bigramas) na geração de *features*.

O número de *features* indicado em cada configuração foi obtido pela aplicação do método de seleção indicado no item 2.2.3.1 e, pela avaliação empírica por validação cruzada (5-Fold) na base de treino da quantidade de variáveis que maximiza a acurácia de cada classificador. A Figura 8 mostra um exemplo deste procedimento, que foi aplicado a todos os casos apresentados no Quadro 1 , com o objetivo de se obter um número adequado de *features* para cada situação. Na Figura 6, é possível de se notar que, variando-se o número de *features,* a acurácia atinge valores maiores.

Figura 8 - Variação da acurácia da Regressão Logística em função do número de features selecionadas para a configuração da base de dados que inclui a posição das frases, suas respectivas PoS-Tags e, apenas unigramas na geração de features por bag-of-words. Avaliação feita a partir de validação cruzada (5-Fold) na base de Treino. Seleção de features feita pelo método do teste-F.



Quadro 1 - Quadro informativo sobre a performance de cada classificador na base de testes, sob diversas condições distintas de dataset. P, R e F1 indicam respectivamente: precisão, recall e F1-Score. I, O, M e R indicam as categorias de frases, respectivamente: Introdução, Objetivos, Métodos e Resultados. NB e RL indicam Naive-Bayes e Regressão Logística. Nas 3 primeiras colunas, "N" indica ausência da feature e "S" indica a sua presença.

(continua)

Dataset					Classificadores									
						Naive Bayes			SVM			Regressão Logist.		
Position	PoS-Tags	Number	# Features	Ngram Range		P	R	F1	P	R	F1	P	R	F1
N	N	N	NB = 401; SVM = 3701; RL = 3801	(1, 1)	I	0,67	0,77	0,72	0,65	0,78	0,71	0,70	0,80	0,74
					O	0,78	0,66	0,71	0,87	0,78	0,82	0,79	0,72	0,76
					M	0,72	0,76	0,74	0,80	0,80	0,80	0,82	0,83	0,83
					R	0,67	0,59	0,63	0,70	0,59	0,64	0,71	0,62	0,66
					Méd	0,70	0,70	0,70	0,74	0,73	0,73	0,75	0,75	0,75
					Acu.	0,699			0,733			0,747		
N	N	N	NB = 1351; SVM = 4901; RL = 2101	(1, 2)	I	0,68	0,80	0,74	0,65	0,85	0,74	0,68	0,81	0,74
					O	0,75	0,76	0,75	0,91	0,72	0,81	0,88	0,74	0,80
					M	0,75	0,74	0,75	0,77	0,76	0,76	0,83	0,77	0,80
					R	0,72	0,58	0,64	0,73	0,57	0,64	0,68	0,63	0,65
					Méd	0,72	0,72	0,72	0,74	0,73	0,73	0,75	0,74	0,74
					Acu.	0,718			0,730			0,740		
S	N	N	NB = 301; SVM = 4080; RL = 4320	(1, 1)	I	0,74	0,72	0,73	0,84	0,87	0,85	0,86	0,87	0,86
					O	0,80	0,67	0,73	0,88	0,78	0,83	0,86	0,74	0,80
					M	0,70	0,77	0,73	0,81	0,82	0,81	0,84	0,85	0,85
					R	0,69	0,70	0,69	0,83	0,84	0,83	0,83	0,87	0,85
					Méd	0,72	0,72	0,72	0,83	0,83	0,83	0,85	0,85	0,85
					Acu.	0,721			0,833			0,847		
S	N	N	NB = 701; SVM = 8801; RL = 741	(1, 2)	I	0,72	0,72	0,72	0,84	0,86	0,85	0,85	0,87	0,86
					O	0,78	0,72	0,75	0,96	0,74	0,83	0,83	0,78	0,80
					M	0,73	0,75	0,74	0,77	0,85	0,81	0,83	0,77	0,80
					R	0,69	0,69	0,69	0,83	0,83	0,83	0,80	0,87	0,83
					Méd	0,72	0,72	0,72	0,84	0,83	0,83	0,83	0,83	0,83
					Acu.	0,721			0,831			0,828		
S	S	N	NB = 401; SVM = 3901; RL = 4500	(1, 1)	I	0,75	0,73	0,74	0,84	0,88	0,86	0,85	0,90	0,87
					O	0,78	0,67	0,72	0,86	0,76	0,81	0,86	0,76	0,81
					M	0,75	0,78	0,76	0,84	0,84	0,84	0,88	0,85	0,87
					R	0,70	0,74	0,72	0,85	0,86	0,85	0,87	0,89	0,88
					Méd	0,74	0,74	0,74	0,85	0,84	0,84	0,86	0,86	0,86
					Acu.	0,740			0,845			0,864		
S	S	N	NB = 1101; SVM = 8801; RL = 8000	(1, 2)	I	0,72	0,78	0,75	0,85	0,88	0,86	0,87	0,88	0,88
					O	0,81	0,76	0,79	0,95	0,72	0,82	0,95	0,69	0,8
					M	0,76	0,78	0,77	0,77	0,85	0,80	0,81	0,86	0,83
					R	0,72	0,67	0,70	0,84	0,83	0,83	0,83	0,87	0,85
					Méd	0,75	0,75	0,75	0,84	0,83	0,83	0,85	0,85	0,85
					Acu.	0,747			0,833			0,847		
S	N	S	NB = 211; SVM = 4110; RL = 3951	(1, 1)	I	0,75	0,70	0,72	0,84	0,86	0,85	0,85	0,86	0,86
					O	0,82	0,69	0,75	0,87	0,78	0,82	0,84	0,74	0,79
					M	0,67	0,74	0,70	0,83	0,82	0,82	0,85	0,85	0,85
					R	0,67	0,71	0,69	0,82	0,85	0,84	0,84	0,88	0,86
					Méd	0,72	0,71	0,71	0,84	0,84	0,84	0,84	0,84	0,84
					Acu.	0,711			0,835			0,845		

Quadro 1 - Quadro informativo sobre a performance de cada classificador na base de testes, sob diversas condições distintas de dataset. P, R e F1 indicam respectivamente: precisão, recall e F1-Score. I, O, M e R indicam as categorias de frases, respectivamente: Introdução, Objetivos, Métodos e Resultados. NB e RL indicam Naive-Bayes e Regressão Logística. Nas 3 primeiras colunas, "N" indica ausência da feature e "S" indica a sua presença.

(conclusão)

Dataset					Classificadores									
						Naive Bayes			SVM			Regressão Logist.		
Position	PoS-Tags	Number	# Features	Ngram Range		P	R	F1	P	R	F1	P	R	F1
S	N	S	NB = 661; SVM = 8401; RL = 8810	(1, 2)	I	0,73	0,70	0,71	0,83	0,85	0,84	0,85	0,88	0,86
					O	0,76	0,71	0,73	0,96	0,74	0,83	0,93	0,71	0,80
					M	0,72	0,75	0,74	0,78	0,84	0,81	0,82	0,84	0,83
					R	0,70	0,72	0,71	0,83	0,84	0,84	0,82	0,87	0,85
					Méd	0,72	0,72	0,72	0,84	0,83	0,83	0,84	0,84	0,84
					Acu.		0,721			0,831			0,840	
N	S	S	NB = 501; SVM = 3801; RL = 4001	(1, 1)	I	0,68	0,77	0,72	0,69	0,76	0,72	0,71	0,80	0,75
					O	0,78	0,69	0,73	0,85	0,76	0,80	0,77	0,76	0,77
					M	0,77	0,79	0,78	0,75	0,81	0,78	0,81	0,82	0,82
					R	0,68	0,61	0,65	0,71	0,61	0,66	0,73	0,62	0,67
					Méd	0,72	0,72	0,72	0,73	0,73	0,73	0,75	0,75	0,75
					Acu.		0,718			0,733			0,752	
N	S	S	NB = 1201; SVM = 6301; RL = 5301	(1, 2)	I	0,66	0,80	0,72	0,69	0,81	0,75	0,68	0,87	0,76
					O	0,79	0,76	0,77	0,91	0,72	0,81	0,93	0,67	0,78
					M	0,77	0,75	0,76	0,84	0,79	0,77	0,76	0,78	0,77
					R	0,72	0,58	0,64	0,73	0,61	0,67	0,73	0,59	0,66
					Méd	0,72	0,72	0,72	0,75	0,74	0,74	0,75	0,74	0,74
					Acu.		0,718			0,740			0,740	
S	S	S	NB = 361; SVM = 3451; RL = 3901	(1, 1)	I	0,74	0,73	0,74	0,85	0,87	0,86	0,86	0,88	0,87
					O	0,79	0,66	0,72	0,86	0,76	0,81	0,88	0,76	0,81
					M	0,73	0,78	0,75	0,83	0,84	0,83	0,86	0,85	0,86
					R	0,71	0,74	0,73	0,85	0,88	0,86	0,84	0,88	0,86
					Méd	0,74	0,74	0,73	0,85	0,85	0,85	0,86	0,86	0,86
					Acu.		0,735			0,847			0,857	
S	S	S	NB = 1121; SVM = 1141; RL = 811	(1, 2)	I	0,72	0,79	0,75	0,85	0,84	0,85	0,86	0,88	0,87
					O	0,83	0,78	0,80	0,90	0,79	0,84	0,85	0,79	0,82
					M	0,78	0,79	0,78	0,80	0,83	0,81	0,86	0,83	0,84
					R	0,74	0,67	0,70	0,84	0,86	0,85	0,83	0,86	0,84
					Méd	0,76	0,75	0,75	0,84	0,84	0,84	0,85	0,85	0,85
					Acu.		0,754			0,838			0,850	

Observa-se que o melhor resultado obtido, em termos de acurácia, foi de 86,4%, que se deu para o teste realizado com a base de dados contendo a posição das frases e, a frequência das *PoS-Tags* e dos unigramas como atributos de cada frase. Resultados muito próximos a esse (acurácia de 85,7%) foram obtidos para a

base contendo as mesmas *features* em conjunto com o pré-processamento das strings numéricas.

A variação do número de atributos selecionados se deu em função da busca pela combinação, em cada caso, que gerasse o melhor desempenho para cada classificador (na base de treino). Dessa forma, nota-se que o algoritmo de Naive Bayes obteve seu máximo para uma média de 693 *features*, enquanto as *Support Vector Machines* e Regressões Logísticas necessitaram respectivamente de médias de 5116 e 4186.

A Tabela 7 ilustra os resultados de classificação obtidos para a configuração de maior acurácia apresentada anteriormente. Para cada uma das classes atribuídas pelo classificador foram selecionados três exemplos de observações da base de teste, dos quais dois foram corretamente classificados e um não.

Tabela 7 - Exemplos de frases extraídas da base de teste, seus rótulos reais e rótulos preditos utilizando-se Regressão Logística aplicada ao dataset contendo a posição e, as frequências de PoS-Tags e unigramas.

(continua)

Frase	Real	Predito
(I) “In the Tahe oilfield in China, heavy oil is commonly lifted using the light oil blending technology.”	Introdução / Contextualização	Introdução / Contextualização
(II) “Carbon dioxide is considered to have high potential to improve the production efficiency of the reservoir.”	Introdução / Contextualização	Introdução / Contextualização
(III) “Xujiahe Formation is a set of terrestrial clastic rocks with low compositional maturity, low cement content, and medium textural maturity.”	Introdução / Contextualização	Métodos
(IV) “The objective of the paper is to optimize the total cost per unit time in light of critical factors which involve spills or leakage, system failure and maintenance cost for optimum extraction.”	Objetivos	Objetivos
(V) “This paper investigates the potential of high acyl GLG as additive for drilling mud.”	Objetivos	Objetivos

Tabela 7 - Exemplos de frases extraídas da base de teste, seus rótulos reais e rótulos preditos utilizando-se Regressão Logística aplicada ao dataset contendo a posição e, as frequências de PoS-Tags e unigramas.

(conclusão)

Frase	Real	Predito
(VI) "The present work sought to determine whether or not a commercially available simulator could accurately simulate results from core flooding experiments."	Objetivos	Resultados / Conclusões
(VII) "The gas response of was studied as a function of the particle size, the temperature, and the gas flow rate."	Métodos	Métodos
(VIII) "These flow rates were used to characterize the performance of the jet pump."	Métodos	Métodos
(IX) "The properties are controlled at such values that the mud provides optimum performance."	Métodos	Introdução / Contextualização
(X) "Furthermore, it has been concluded that aquifer strength has a little effect on coning behavior during oil production process."	Resultados / Conclusões	Resultados / Conclusões
(XI) "Final results clearly indicate that substantial electric power savings are possible if production control is executed by VSDs instead of the present practice of using surface chokes."	Resultados / Conclusões	Resultados / Conclusões
(XII) "Furthermore, simulations using the Well-X model suggest an incremental oil recovery factor of 14.7% OOIP due to surfactant-polymer flooding."	Resultados / Conclusões	Métodos

Fontes das frases :

(I) ZHU, M. et al. Research on viscosity-reduction technology by electric heating and blending light oil in ultra-deep heavy oil wells. **Journal of Petroleum Exploration and Production Technology**, v. 5, n. 3, p. 233-239, 2015.

(II) GHANI, A.; KHAN, F.; GARANIYA, V. Improved oil recovery using CO2 as an injection medium: a detailed analysis. **Journal of Petroleum Exploration and Production Technology**, v. 5, n. 3, p. 241-254, 2015.

(III) GONG, L. et al. Reservoir characterization and origin of tight gas sandstones in the Upper Triassic Xujiahe formation, Western Sichuan Basin, China. **Journal of Petroleum Exploration and Production Technology**, v. 6, n. 3, p. 319-329, 2016.

(IV) SHAH, N.; MISHRA, P. Oil production optimization: a mathematical model. **Journal of Petroleum Exploration and Production Technology**, v. 3, n. 1, p. 37-42, 2013.

(V) GAO, C. H. Preliminary evaluation of gellan gum as mud thickener. **Journal of Petroleum Exploration and Production Technology**, v. 6, n. 4, p. 857-861, 2016.

(VI) RAI, S. K.; BERA, A.; MANDAL, A. Modeling of surfactant and surfactant–polymer flooding for enhanced oil recovery using STARS (CMG) software. **Journal of Petroleum Exploration and Production Technology**, v. 5, n. 1, p. 1-11, 2015.

(VII) RATHORE, D.; KURCHANIA, R.; PANDEY, R. K. Gas sensing properties of size varying CoFe 2 O 4 nanoparticles. **IEEE Sensors Journal**, v. 15, n. 9, p. 4961-4966, 2015.

(VIII) NANDA, R.; GUPTA, S.; SHUKLA, A. K. N. Experimental setup for performance characterization of a jet pump with varying angles of placement and depth. **Journal of Petroleum Exploration and Production Technology**, v. 1, n. 2-4, p. 107-110, 2011.

(IX) SAFI, B. et al. Physico-chemical and rheological characterization of water-based mud in the presence of polymers. **Journal of Petroleum Exploration and Production Technology**, v. 6, n. 2, p. 185-190, 2016.

(X) AZIM, R. A. Evaluation of water coning phenomenon in naturally fractured oil reservoirs. **Journal of Petroleum Exploration and Production Technology**, v. 6, n. 2, p. 279-291, 2016.

(XI) TAKACS, G. How to improve poor system efficiencies of ESP installations controlled by surface chokes. **Journal of Petroleum Exploration and Production Technology**, v. 1, n. 2-4, p. 89-97, 2011.

(XII) BU, P. X. et al. Simulation of single well tracer tests for surfactant–polymer flooding. **Journal of Petroleum Exploration and Production Technology**, v. 5, n. 4, p. 339-351, 2015.

4.6 - Discussão

Como já previamente discutido na Seção 4.4, o presente trabalho focou-se nos algoritmos de Naive Bayes, SVM e Regressão Logística para a classificação, devido à frequente utilização dos dois primeiros em outros trabalhos consultados - que serviram de referência para a performance esperada nesse tipo de tarefa (Wang e Manning, 2012) - e, pela natureza probabilística, semelhante à Naive Bayes,

porém, com diversas vantagens sobre esse, da Regressão Logística, que, inclusive, alcançou os melhores resultados de acurácia nos experimentos realizados.

Técnicas comuns de processamento de linguagem natural como o uso de lematizadores e a remoção de *stop-words*, que ajudam na remoção de potenciais ruídos para a classificação textual (Salvetti et al., 2004) não foram amplamente exploradas pois testes iniciais indicaram uma queda significativa da performance de classificação dos algoritmos utilizados, demonstrando a importância do conteúdo completo das sentenças (em relação às palavras e suas inflexões) na seleção de *features*.

Dos resultados apresentados na Seção 4.5 é possível extrair diversas conclusões a respeito das particularidades de nossa base de dados e, da influência delas no processo de classificação.

Primeiramente, como mostrado pelos trabalhos de McKnight e Srinivasan (2003) e Yamamoto e Takagi (2005), é notável que a posição das frases consiste de uma *feature* extremamente relevante para a predição das classes propostas, dada a natureza estruturada e ordenada dos *abstracts*. Como apresentado no Quadro 1, sempre que utilizada nos datasets, a posição gerou um aumento significativo da acurácia máxima alcançada pelos classificadores. Isso fica evidente, por exemplo, ao observar-se os testes com Regressão Logística a partir de um dataset sem pré-processamento de string numéricas, sem contagem de *PoS-Tags* e com a geração de *features* apenas a partir de unigramas, em que a adição da *feature* de posição a essa configuração gerou um aumento de aproximadamente 10% de acurácia com uma seleção de apenas 301 atributos da base (100 atributos a menos do que o selecionado na configuração sem a posição para obtenção da acurácia máxima de 74,7%).

Outro ponto importante a ser ressaltado é a contribuição da frequência de aparição das classes gramaticais (*PoS-Tags*) de cada frase como *features*. Apesar de não ser uma abordagem tradicional para o problema de classificação textual, que usualmente se utiliza dessas *tags* apenas para a filtragem de palavras pertencentes a determinadas classes (Paula e Bonatti, 2015), ela contribuiu positivamente para o aumento da acurácia dos três classificadores, demonstrando o poder discriminativo dessas *features* dentro de nossa base de dados.

A maior acurácia, obtida com o uso de Regressão Logística e de um *dataset* composto das posições das frases, *PoS-Tags* e frequência dos unigramas como *features*, foi de 86,4% se aproximando muito dos 89,2% relatados por McKnight e Srinivasan (2003), que realizaram tarefas semelhantes.

Em nosso melhor resultado, o *F1-Score* obtido variou na faixa de 0,81 a 0,87, para o conjunto de testes considerando as diversas classes. McKnight e Srinivasan (2003) relataram um *F1-Score* variando de 0,52 a 0,86*,* enquanto Yamamoto e Takagi (2005) relataram um *F1-Score* variando de 0,73-0,89 em tarefas de classificação também semelhantes.

Em nosso trabalho, notamos, ainda, que a precisão variou na faixa 0,85 - 0,88, e o *recall*, variou entre 0,76 e 0,90, para as classes consideradas, em nosso melhor resultado.

5 - Busca e Recuperação de Informações

Nos dois capítulos anteriores exploramos a rotulação de frases de resumos de artigos científicos e, sobretudo, métodos para sua classificação de maneira automática. Essa classificação tem como subproduto a geração de metadados a respeito das frases dos textos: geram-se informações estruturadas que antes estavam indisponíveis, a partir dos dados puramente textuais. Essas informações adicionais, em formato estruturado e computacionalmente tratável, podem servir a diversas aplicações, como mencionado no Capítulo 1. A partir de agora, exploraremos uma das aplicações possíveis para o uso destes dados, que é a de Recuperação de Informações (*Information Retrieval*).

5.1 - Modelo de Linguagem adotado e Suavização

Para a tarefa de busca, neste trabalho, utilizamos um Modelo de Linguagem de unigramas, tal como apresentado na Seção 2.5. Na mesma seção, também discutimos a importância da Suavização nos Modelos de Linguagem: atribuir probabilidade não nula a palavras ausentes do modelo, bem como melhorar a acurácia da estimação da probabilidade de palavras em geral (Zhai e Lafferty, 2004). Como já discutido, diversos métodos de suavização estimam a probabilidade $P(w \mid D)$ utilizando-se de $P(w \mid C)$, ou seja, relacionam a frequência da palavra w no documento D com sua frequência na coleção de documentos C.

Esta abordagem, no entanto, não consegue capturar a relação entre palavras com sentido similar ou que são usadas em contextos semelhantes. Por exemplo, se nossa busca (*query*) contiver a palavra "*fuel*" e um certo documento não possuir tal palavra, mas possuir palavras com sentido semelhante como "*gasoline*" ou "*diesel*" elas não serão consideradas tão importantes e receberão, ao contrário, uma pontuação que é proporcional a sua frequência em toda a Coleção: $P(w|C)$.

Para abordar essa questão, Ganguly et al. (2015) apresentam um método que une o Modelo de Linguagem com o uso de *embeddings* (representações vetoriais de palavras), com o intuito de dar maior importância a palavras dos documentos que são usadas em contextos semelhantes àquelas que aparecem na consulta. A

representação de palavras utilizada é a Word2Vec (Mikolov et al., 2013a), já apresentada e discutida na Seção 2.6. De acordo com Ganguly et al. (2015), a probabilidade de transformar um termo *t'* em um termo *t ,* dado um documento D pode ser expressa como:

$$P(t|t', D) = \frac{sim(t, t')}{\sum_{t'' \in D} sim(t, t'')} = \frac{sim(t, t')}{\sum(D)} \quad . \qquad (5.1)$$

Na equação 5.1, $sim(x, y)$ indica a Similaridade de Cossenos, anteriormente definida pela equação 2.34. Os autores destacam que a probabilidade de selecionar um termo *t',* dado um termo *t* é proporcional à similaridade entre os termos (calculada por 2.34). Essa estimativa pode ser utilizada na suavização da probabilidade $P(w|D)$, fornecendo informações adicionais especialmente quando a palavra *w* está ausente do documento. Este tipo de estimativa tenta fornecer informações contextuais adicionais, buscando capturar a relação entre as palavras. Assim, palavras "próximas" em sentido àquelas da busca serão favorecidas, em detrimento de palavras mais "distantes". Consequentemente, um documento que não contenha exatamente as palavras buscadas, mas que possua palavras relacionadas, terá chances de ser sinalizado como relevante.

Propomos um modelo de suavização que utiliza a similaridade entre a palavra buscada e a palavra "mais similar do documento", para evitar-se a atribuição de probabilidade zero, no caso de a palavra consultada estar ausente. Baseado na equação 5.1 teríamos:

$$P_{max}(t'|w, D) = \frac{1}{\sum_{w' \in D} sim(w, w')} \max_{t' \in D} [sim(w, t')] \quad . \qquad (5.2)$$

A equação 5.2 denota, portanto, a probabilidade *máxima* de se transformar um termo *w* da busca em um termo *t'* do documento *D*, dados *w* e *D.* Em outras palavras, a probabilidade de se transformar um termo *w* da busca no termo mais similar a ele, no documento.

Assim, tomando as equações 5.2 e 2.31 e combinando-as linearmente como na equação 2.32, podemos obter uma estimativa para $P(w|D)$ que leve em conta o contexto e relação entre palavras:

$$\hat{P}_\beta(w|D) = (1-\beta)\frac{tf(w,D)}{\sum_{w' \in D} tf(w',D)} + \beta \frac{1}{\sum_{w' \in D} sim(w,w')} \max_{t' \in D}[sim(w,t')] \quad . \qquad (5.3)$$

Na equação 5.3, a combinação linear possui um parâmetro β ($0 \leq \beta \leq 1$), que regula a importância da ocorrência das palavras exatas da consulta, em oposição à importância do contexto e da ocorrência de palavras semelhantes às da busca (mas não idênticas). Quanto mais próximo de 0, maior a importância que se dá a ocorrência exata de *w* no documento D. Ao contrário, quanto mais próximo de 1, maior a contribuição exercida por palavras similares a *w,* mas não idênticas a esta. Por conseguinte, para atribuirmos pontuação a um certo documento, poderemos utilizar a equação 2.29 e a 5.3 para escrever:

$$score(D) = P(q|D) \propto P(D|q) = \prod_{w \in q} \hat{P}_\beta(w|D) \quad . \qquad (5.4)$$

5.2 - Combinando Modelo de Linguagem e classificação de frases

O equacionamento da probabilidade de um documento dada uma *query* (equação 5.4) usando Modelos de Linguagem pode ser complementado pelas informações e metadados gerados pelas etapas de classificação apresentadas anteriormente. É possível, por exemplo, utilizarmos a classificação de frases para permitirmos ao usuário filtrar os termos da busca por categorias de frase. Dessa forma, ele será capaz de buscar um certo tema ou palavras nos Objetivos dos artigos ou ainda em seus Métodos, Resultados, e assim por diante. Para isso, podemos calcular a probabilidade de uma dada frase pertencer a cada uma das categorias utilizando-nos dos classificadores obtidos na no capítulo anterior.

No caso do classificador utilizando Regressão Logística (que foi aquele que forneceu os melhores resultados) as probabilidades são obtidas no esquema de *One-Vs-All*, ou seja, calcula-se, em princípio, a probabilidade de cada classe de maneira independente, considerando tal classe como positiva e as demais como negativas: $P(Y = 1|X)$ (vide equação 2.11). Posteriormente, normalizam-se essas probabilidades entre si, dividindo-se cada uma pela somatória de todas elas.

Assim, se assumirmos que cada frase é um documento independente, e estivermos interessados em considerar a probabilidade de a frase pertencer à classe γ desejada, podemos re-escrever a pontuação da equação 5.4 como:

$$\begin{aligned} score(D) = P(D|q,\gamma) \propto P(q,\gamma|D) = P(q|D) \cdot P(\gamma|D) \\ = \Big[\prod_{w \in q} \hat{P}_\beta(w|D)\Big] \cdot P(\gamma|D) \end{aligned} \tag{5.5}$$

Em nossos testes, utilizamos a equação 5.5 para calcular a pontuação de cada frase dos resumos de cada artigo considerado. Os artigos são então avaliados, e recebem uma pontuação que corresponde à pontuação da frase de máxima probabilidade dentro daquele artigo.

Considere um artigo A_i, com frases $s_1, s_2, ..., s_k \, \varepsilon \, A_i$. Cada frase recebe uma pontuação calculada de acordo com a equação 5.5. O artigo receberá a pontuação de modo que: $score(A_i) = max(score(s_j)),\ s_j \, \varepsilon \, A_i$. Ou seja, em outras palavras, o artigo terá a pontuação de sua frase mais relevante.

Em nossa abordagem, estabelecemos que, para um artigo ser considerado relevante, sua pontuação mínima deveria ser aquela de um artigo que não possuísse nenhuma das palavras buscadas mas possuísse palavras com similaridade mínima de 0,5 com as da busca. Dessa forma, substituindo-se essas condições na equação 5.4, teremos:

$$P_{min\beta}(q|D) = \prod_{w \in q} \left(\beta \cdot \frac{0.5}{\sum_{w' \in D} sim(w, w')} \right) \tag{5.6}$$

Obviamente, para uma mesma *query q,* essa pontuação da equação 5.6 variará de documento para documento (*D*), que em nosso caso são as frases dos resumos de cada artigo.

5.3 - Método de avaliação do sistema de buscas

Para validação deste método, coletamos nova base de dados, diferente daquela utilizada para o treino do classificador de frases. Trata-se de uma base coletada da revista *Journal of Petroleum Exploration and Production Technology,* contendo 320 artigos de variados temas relacionados à extração e produção de petróleo.

Embora esteja fora do escopo deste trabalho fornecer uma avaliação exaustiva dos métodos de busca obtidos, alguns testes foram realizados com o objetivo de se demonstrar a efetividade do sistema. Sobretudo, deseja-se demonstrar o potencial deste tipo de abordagem, e as diversas aplicações que dela podem decorrer.

Para a avaliação dos resultados duas métricas são relevantes: a precisão e o *recall* (sensibilidade). A precisão (*precision*) indica a fração dos documentos classificados como relevantes que realmente é relevante, ou seja, é uma forma de estimar a probabilidade do documento ser relevante, dado que foi classificado dessa forma: $P(relevante \mid recuperado)$. O *recall (*ou sensibilidade), por outro lado, indica qual parcela dos documentos relevantes foi recuperada, ou seja: $P(recuperado \mid relevante)$.

A importância da sensibilidade e da precisão para a avaliação de sistemas de busca é relativa ao tipo de aplicação considerada. Certas aplicações como buscas na *web* tendem a favorecer a precisão, pois o usuário está muito mais interessado em ter resultados relevantes na primeira página do que obter *todos* os resultados relevantes possíveis. Em outras aplicações como sistemas de buscas fiscais ou jurídicos, que envolvem tarefas mais minuciosas, a sensibilidade tem papel mais importante, pois não se pode tolerar perder certos resultados (Manning, Raghavan e Schütze, 2008).

A sensibilidade para o sistema de buscas não será calculada neste trabalho, pois, nossa maior preocupação é avaliar se os resultados retornados são de fato

úteis, e, não verificar se *todos* os resultados úteis são encontrados. De fato, em um sistema de buscas como o nosso, quanto maior a base de dados, menor a importância do *recall*. Por isso a precisão (equação 5.7) será suficiente para nossas avaliações:

$$precision = \frac{TP}{TP + FP} = \frac{(nro.\ docs.\ relevantes)}{(nro.\ docs\ recuperados)}. \quad (5.7)$$

Uma das maneiras comuns de se avaliar a precisão de um sistema é utilizando-se da mAP (*Mean Average Precision,* ou Precisão Média). Ela é calculada avaliando-se a precisão (Equação 5.7) de diversas consultas (ou necessidades informacionais) $q_1, q_2, ..., q_m \, \varepsilon \, Q$ e tomando-se a média (Manning, Raghavan e Schütze, 2008).

$$MAP = \frac{1}{|Q|} \sum_{q \in Q} precision(q) \quad . \quad (5.8)$$

A equação 5.8 mostra uma forma de representação do cálculo do mAP, em que $|Q|$ indica o número de buscas consideradas na avaliação do mAP e $precision(q)$ indica a precisão (equação 5.8) calculada para a consulta *q* sobre a base de dados considerada.

5.4 - Experimentos e Resultados

Para nossos experimentos, foi necessário utilizar-se de um modelo de *embedding* (ou dicionário), como explicado na Seção 2.6. Essa necessidade surge para que possamos utilizar o método proposto por Ganguly et al. (2015), cuja ideia central é resumida pela equação 5.1. Escolhemos utilizar um modelo já pré-treinado em textos coletados da *Wikipedia*, que é modelo chamado *"Common Crawl",* construído por Pennington, Socher e Manning (2014). Esse *embedding* foi treinado em 42 bilhões de *tokens*, possuindo um vocabulário de tamanho 1,9 milhões, vetores de 300 dimensões e um total de 1,75 GB.

Um conjunto de 50 consultas foi avaliado. Esse número é considerado adequado por Manning, Raghavan e Schütze (2008), em suas discussões sobre avaliação de sistemas de Recuperação de Informações. As consultas consideradas nos testes foram sugeridas por alguns voluntários, de maneira independente, após estes realizarem uma inspeção rápida dos assuntos contidos na base. Eles foram orientados a fornecer consultas constituídas por palavras-chave, contendo de uma a três palavras cada. Deveriam, também, indicar, para cada consulta, uma ou mais classes de frases em que estavam interessados: Introdução, Objetivos, Métodos ou Resultados. A Tabela 8 mostra alguns exemplos de consultas consideradas.

A avaliação dos resultados foi feita pelos autores, que, de maneira independentemente, assinalaram, para cada resultado retornado, se ele era relevante ou não. Os resultados só foram considerados relevantes quando foram assinalados desta forma pelos dois avaliadores. Caso contrário, foram considerados irrelevantes.

A pontuação, calculada pela equação 5.5 para cada documento (dada uma busca), depende de um parâmetro β (vide equação 5.3). Esse parâmetro é obtido empiricamente. Através da avaliação qualitativa e testes em algumas consultas (diferentes das 50 consideradas para o teste), chegou-se à conclusão de que valores em torno de 0,1 a 0,3 para β forneciam bons resultados. Para nossos testes, utilizamos o valor de $\beta = 0,2$.

Nosso sistema de busca impõe uma pontuação mínima para os resultados, por meio da equação 5.6. Resultados abaixo dessa pontuação não são exibidos (ou seja, são considerados irrelevantes através do critério estabelecido). Além disso, um máximo de 10 resultados por consulta foi determinado. Ou seja, embora pudessem existir mais resultados relevantes, apenas os 10 mais relevantes são exibidos para avaliação. Um mínimo de resultados por consulta não foi imposto.

A avaliação da base de 50 consultas forneceu um resultado de **mAP** de **0,80** com um **desvio-padrão** (amostral) de **17,1%**. Percebeu-se que os resultados mostram, em primeiro lugar, resultados que contém todas as palavras da consulta. Após isso, mostram-se os resultados que possuem algumas das palavras da consulta e palavras relacionadas. Na Figura 9 é possível ver isso. Ela mostra alguns resultados de uma das consultas testadas. As palavras buscadas eram *“reduce,*

costs" nas classes Objetivos e Resultados. Percebe-se que no terceiro resultado as palavras mais similares encontradas foram "*eliminate*" e "*expenditures*", que são palavras que possuem significado similar ao pretendido, o que mostra o sucesso da generalização de nosso algoritmo.

Figura 9 - Exemplo de resultados obtidos por nosso sistema de recuperação de informações. Os resultados mostram: título, referência bibliográfica, frase mais relevante e, em vermelho, os termos mais próximos daqueles especificados na busca.

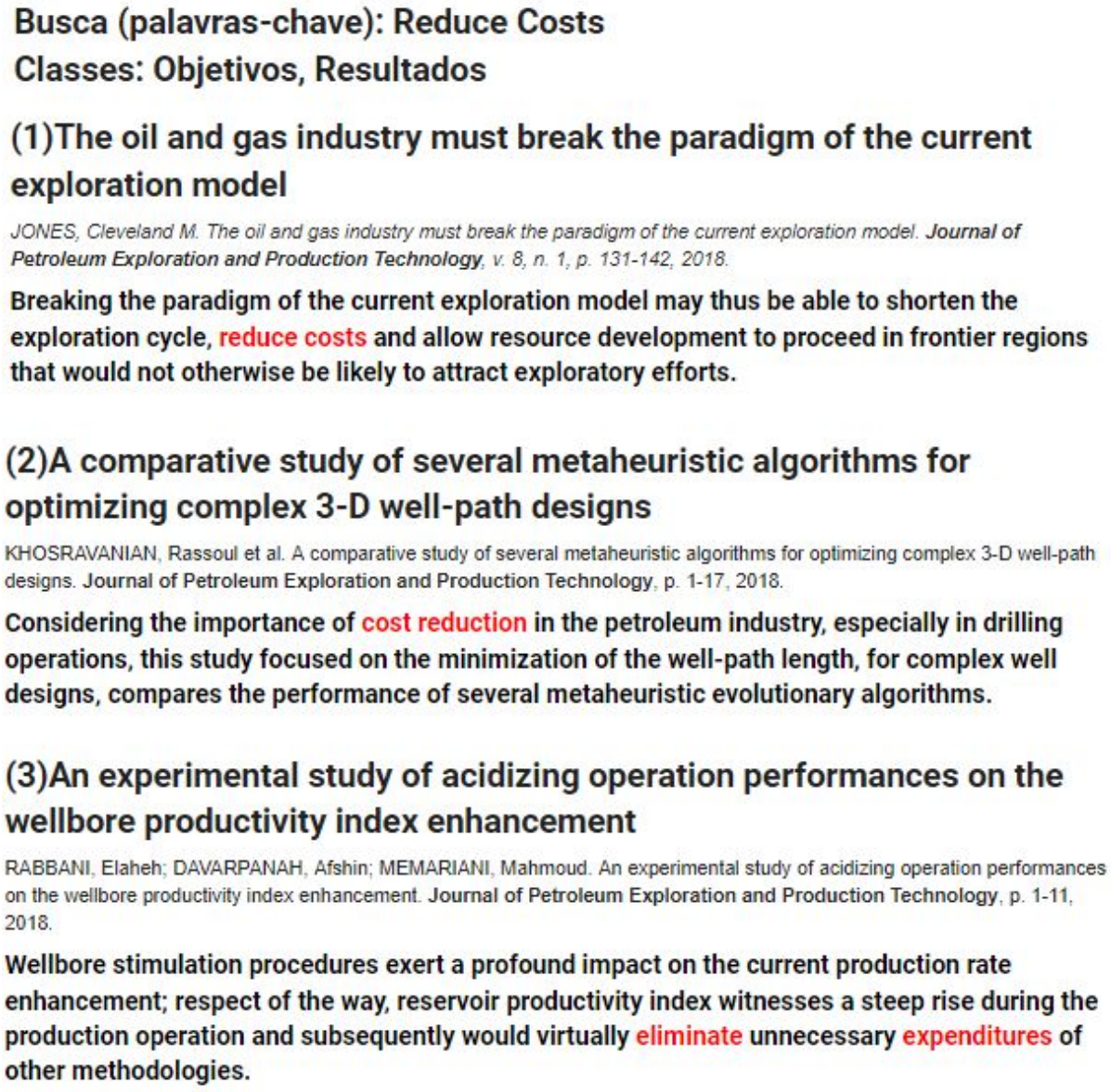


Tabela 8 - Exemplos de consultas utilizadas na avaliação do sistema de Recuperação de Informações.

Palavras-chave da busca	**Classes de interesse**
Reduce, Costs	Objetivos, Resultados
Increase, Production	Objetivos, Resultados
Oil, Recovery, Evaluation	Objetivos, Resultados
Predict, Production, Rate	Objetivos, Resultados
Drilling, Improve	Objetivos, Resultados
Fourier, Transform	Métodos, Resultados
Flow, Model	Objetivos, Métodos, Resultados
Permeability, Measure	Objetivos, Resultados
Finite, Elements	Métodos
Polymer, Injection	Métodos

6 - Conclusões e Próximos Passos

Com relação à classificação de frases, nossos resultados se equiparam ou superam os resultados de outros trabalhos semelhantes, como discutido na seção 4.6. Nosso sistema foi capaz de classificar frases em categorias do discurso científico, em textos de resumos de artigos científicos da área de Engenharia de Petróleo. Atingimos uma acurácia de 86% (comparável aos trabalhos analisados), e um F1-Score de 0,81-0,88 (superior aos trabalhos analisados). Uma das contribuições do nosso trabalho é ter feito tal tarefa em textos de Engenharia, uma vez que, na literatura, a quase totalidade dos trabalhos disponíveis sobre o assunto trata da classificação de frases em artigos da área das Ciências Biológicas ou Médicas.

Os principais pontos para desenvolvimento futuro em relação ao sistema de classificação são:

- O aumento do número de observações rotuladas da base de treino, em especial, da classe “Objetivos” visando a melhora da acurácia de classificação e, do *recall* dessa classe que apresentou valores inferiores em comparação às demais (0,76 para o melhor classificador).
- A anotação das frases de treino por pesquisadores independentes e, avaliação da concordância de atribuição de classe entre esses e os autores deste trabalho, eliminando-se o viés introduzido pela anotação realizada simplesmente pelos autores. Tal prática também proporciona a avaliação da qualidade da anotação de maneira mais precisa e confiável, como mostrado por Agarwal e Yu (2009).
- O teste de outros algoritmos de classificação, como as combinações de Naive Bayes e SVM propostas por Wang e Manning (2012), visando o aumento da acurácia.

Em relação ao sistema de busca, demonstramos que foi possível unir informações obtidas automaticamente pelo classificador de frases a algoritmos de Recuperação de Informação baseados no uso de Modelos de Linguagem. Também

incluímos no sistema de buscas o uso de *embeddings* para suavização do cálculo das probabilidades, com o objetivo de tratar, de maneira mais apropriada, os casos de documentos que não contém as palavras da busca, mas sim palavras semelhantes, e que são também relevantes à consulta.

Os resultados dos testes realizados mostram qualitativamente e quantitativamente que o resultado pretendido foi alcançado: as consultas retornam em média 80% de resultados relevantes e, além disso, consegue-se encontrar documentos que possuem palavras diferentes das buscadas, mas com o mesmo sentido. Além disso, o filtro por categorias de frases classificadas automaticamente mostrou-se, também, uma ferramenta promissora.

Com relação ao sistema de Recuperação de Informações, podemos dizer que os trabalhos futuros envolvem:

- Realização de testes com maior número de consultas, sendo a relevância dos documentos anotada por especialistas da área de Engenharia de Petróleo. A anotação de todos os documentos (Relevante vs. Irrelevante), para cada consulta, seria um trabalho dispendioso e demorado, porém permitiria o cálculo do *Recall,* informação que não possuímos no momento.
- Variação de parâmetros do algoritmo de busca, tanto β quanto os parâmetros de pontuação mínima dos documentos.
- Testes com diferentes modelos *embeddings,* treinados em diferentes *corpora*, para a representação vetorial de palavras.
- Testes com representação da consulta e dos documentos em forma vetorial. A utilização de algoritmos e heurísticas apropriadas para calcular-se a similaridade entre uma consulta e um documento (ambos em forma vetorial) constitui, também, um método interessante de busca, tal como defendem: Manning, Raghavan e Schütze (2008) no Capítulo 6 ("*Scoring, term weighting and the vector space model*"), ou Mikolov et al. (2013b), o qual apresenta uma abordagem mais moderna para o tema. Uma abordagem introdutória interessante é apresentada Singhal et al. (1996).

Referências

ABERER, K.I.; CUDRÉ-MAUROUX, P.; HAUSWIRTH, M. The chatty web: emergent semantics through gossiping. In: **Proceedings of the 12th international conference on World Wide Web**. ACM, 2003. p. 197-206.

AGARWAL, S.; YU, H. Automatically classifying sentences in full-text biomedical articles into Introduction, Methods, Results and Discussion. Bioinformatics, v. 25, n. 23, p. 3174-3180, 2009.

BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Biblioteca Digital de Teses e Dissertações da USP. Disponível em: <http://www.teses.usp.br/>. Acesso em 18 de junho de 2018.

BIRD, S.; KLEIN, E.; LOPER, E. **Natural language processing with Python: analyzing text with the natural language toolkit**. O'Reilly Media, Inc, 2009.

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: **Proceedings of the fifth annual workshop on Computational learning theory**. ACM, 1992. p. 144-152.

CHEN, S. F. AND GOODMAN, J. An empirical study of smoothing techniques for language modeling. Tech. Rep. TR-10-98, Harvard University. 1998.

CONSTANTIN, A. et al. The document components ontology (DoCO). *Semantic web*, v. 7, n. 2, p. 167-181, 2016.

CORTES, C.; VAPNIK, V. Support-vector networks. Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.

DONG, L.; LAPATA, M. Language to logical form with neural attention. *arXiv preprint arXiv:1601.01280*, 2016.

ELSSIED, N. O. F.; IBRAHIM, O.; OSMAN, A. H. A novel feature selection based on one-way anova f-test for e-mail spam classification. Research Journal of Applied Sciences, Engineering and Technology, v. 7, n. 3, p. 625-638, 2014.

FURTADO, P. H. T. Interpretação automática de relatórios de operação de equipamentos. Dissertação (Mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro. 2017.

GALAR, M. et al. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. Pattern Recognition, v. 44, n. 8, p. 1761-1776, 2011.

GANGULY, D. et al. Word embedding based generalized language model for information retrieval. In: **Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval**. ACM, 2015. p. 795-798.

GOUTTE, C.; GAUSSIER, E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In: **European Conference on Information Retrieval**. Springer, Berlin, Heidelberg, 2005. p. 345-359.

GUARINO, N.; OBERLE, D.; STAAB, S. What is an ontology?. In: **Handbook on ontologies**. Springer, Berlin, Heidelberg, 2009. p. 1-17.

HSU, C.-W. et al. **A practical guide to support vector classification**. 2003.

JOACHIMS, T. Transductive Inference for Text Classification Using Support Vector Machines. ICML, vol. 99, pp. 200–209, 1999.

JELINEK, F.; MERCER, R. Interpolated estimation of markov source parameters from sparse data. In: **Pattern Recognition in Practice**. 1980. p. 381–402.

JURAFSKY, D.; MARTIN, J. H. **Speech and language processing**. London: Pearson, 2019. No prelo. Disponível em: <https://web.stanford.edu/~jurafsky/slp3/>. Acesso em: 1 nov. 2018.

KHABSA, M.; GILES, C. L. The number of scholarly documents on the public web. PloS one, v. 9, n. 5, p. e93949, 2014.

LADEIRA, A. P.; ALVARENGA, L. **PROCESSAMENTO DE LINGUAGEM NATURAL: em busca de evidências temáticas nas publicações nacionais e contemporâneas**. 2009.

LUZ, F. F.; FINGER, M. Semantic Parsing Natural Language into SPARQL: Improving Target Language Representation with Neural Attention. *arXiv preprint arXiv:1803.04329*, 2018. Disponível em <https://arxiv.org/abs/1803.04329/>. Acesso em: 1 nov. 2018.

MANNING, C. D., RAGHAVAN, P.; SCHÜTZE, H. **Introduction to Information Retrieval**. Cambridge : Cambridge University Press, 2008.

MCCALLUM, A.; NIGAM, K. A comparison of event models for naive Bayes text classification, In: **AAAI-98 Workshop on Learning for Text Categorization**. Madison, Wisconsin. The AAAI Press. p. 41-48, 1998.

MCCORMICK, C. *Word2Vec Tutorial* - The Skip-Gram Model. 2016. Disponível em: <http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/>. Acesso em: 29 out. 2018.

MCHUGH, M. Interrater reliability: the kappa statistic. Biochemia medica vol. 22,3.. p. 276-82. 2012.

MCKNIGHT, L.; SRINIVASAN, P. Categorization of sentence types in medical abstracts. In: **AMIA Annual Symposium Proceedings**. American Medical Informatics Association, 2003. p. 440.

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013a. Disponível em: <https://arxiv.org/abs/1301.3781>. Acesso em: 1 nov. 2018.

MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. In: **Advances in neural information processing systems**. 2013b. p. 3111-3119.

MIKOLOV, T.; YIH, W-T.; ZWEIG, G. Linguistic regularities in continuous space word representations. In: **Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. 2013. p. 746-751.

MILLER, E. An introduction to the resource description framework. *Bulletin of the Association for Information Science and Technology*, v. 25, n. 1, p. 15-19, 1998.

MOSTELLER, F.; WALLACE, D. Inference and disputed authorship: The Federalist. 1964.

NADKARNI, P. M.; OHNO-MACHADO, L.; CHAPMAN, Wendy W. Natural language processing: an introduction. Journal of the American Medical Informatics Association, v. 18, n. 5, p. 544-551, 2011.

PAULA, A. G.; BONATTI, R. Development of email classifier in Brazilian Portuguese using feature selection for automatic response. Dissertação (Engenharia Mecatrônica) - Escola Politécnica, Universidade de São Paulo. São Paulo. 2015.

PAK, A.; PAROUBEK, P. Twitter as a corpus for sentiment analysis and opinion mining. In: **LREc**. 2010. p. 1320-1326.

PENNINGTON, J.; SOCHER, R.; MANNING, C. Glove: Global vectors for word representation. In: **Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)**. 2014. p. 1532-1543

RUIZ-INIESTA, A.; CORCHO, O. A review of ontologies for describing scholarly and scientific documents. In: **SePublica**. 2014.

SALVETTI, F.; LEWIS, S.; REICHENBACH, C. Automatic opinion polarity classification of movie. Colorado research in linguistics, v. 17, n. 1, p. 2, 2004.

SINGH, G.; SOLANKI, A. An algorithm to transform natural language into sql queries for relational databases. *Selforganizology*, v. 3, n. 3, p. 100-116, 2016.

SINGHAL, A., BUCKLEY, C., and MITRA, M. Pivoted document length normalization. In **Proc. SIGIR**, ACM Press. pp. 21-29. 1996

SHOTTON, D.; PERONI, S. The Discourse Elements Ontology. 2016. Disponível em: <https://sparontologies.github.io/deo/current/deo.html>. Acesso em: 16 jun. 2018.

SOLDATOVA, L.; KING, R.; CLARE, A. EXPO - Ontology of scientific experiments. 2007. Disponível em: <http://expo.sourceforge.net/>. Acesso em: 16 jun. 2018.

SOLDATOVA, L.; LIAKATA, M. *CISP the proposed Core Information about Scientific Papers*. 2007. Disponível em: <https://www.researchgate.net/publication/277992477_CISP_-_the_proposed_Core_Information_about_Scientific_Papers>. Acesso em: 16 jun. 2018.

SOUMYA, G.; SHIBILY, J. Text Classification by Augmenting Bag of Words (BOW) Representation with Co-occurrence Feature. IOSR Journal of Computer Engineering (IOSR-JCE) . Volume 16, Issue 1. p. 34-38. 2014.

TU, C.-J., CHUANG L.-Y., CHANG J.-Y., YANG C.-H. Feature selection using PSO-SVM. International Journal of Computer Science, 2007.

WANG, S.; MANNING, C. D. Baselines and bigrams: Simple, good sentiment and topic classification. In: **Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2**. Association for Computational Linguistics, 2012. p. 90-94.

YAMAMOTO, Y.; TAKAGI, T. A sentence classification system for multi biomedical literature summarization. In: **Data Engineering Workshops**, 2005. 21st International Conference on. IEEE, 2005. p. 1163-1163.

YUN, C. et al. An experimental study on feature subset selection methods. In: **cit. IEEE**, 2007. p. 77-82.

ZHAI, C.; LAFFERTY, J. A study of smoothing methods for language models applied to information retrieval. ACM Transactions on Information Systems (TOIS), v. 22, n. 2, p. 179-214, 2004.