RECUPERAÇÃO DA INFORMAÇÃO E A IMPORTÂNCIA DO PRÉ-PROCESSAMENTO

Alex Marino Gonçalves de Almeida[1], Natália Aparecida Beirão Leite[2], Ricardo Fabrício Ramos[3]

Resumo
A categorização de documentos consiste na classificação dos mesmos em uma ou mais categorias existentes, de acordo com os assuntos ou conceitos presentes em seus conteúdos. A aplicação mais comum da categorização de documentos é a indexação de documentos para os Sistemas de Recuperação de Informação visando uma melhor recuperação destes documentos. Porém, são também utilizados na categorização de mensagens e notícias. Para que a classificação seja realizada de forma satisfatória é necessário que os documentos a serem classificados passem por um processo de estruturação, determinado pré-processamento, a fim de otimizar seu conteúdo para análise dos algoritmos classificadores. A finalidade deste trabalho é demonstrar, por meio de experimentos, a importância do pré-processamento na categorização de documentos, uma vez que este influencia diretamente nos resultados classificadores. Para isto foram realizadas análises de um conjunto de documentos com as ferramentas Statistica12 para o pré-processamento e *Weka* para a classificação. A importância do pré-processamento foi determinada com análise dos resultados obtidos por meios dos algoritmos classificadores SMO, Naive Bayes e J48.
Palavras-chave: Classificação de Texto, Recuperação da Informação, Aprendizado de Máquina.

Abstract
The categorization of documents consists the classification thereof in one or more catagories according to the subjects or concepts present in its contents. The most common application of categorization of documents is document indexing for information retrieval systems to improve the recovery of these documents. But they are also used in categorizing messages and news as described in the work it is also used in filtering information in the summarization of texts, among others. For classification is performed satisfactorily it is necessary for the documents to be classified undergo a process of structuring, certain pre- processing in order to optimize their content for analysis of classifiers algorithms. The purpose of this study is to show through experiments the importance of pre - processing the categorization of documents since this directly influences the results classifiers. For this analysis were performed a set of documents with the Statistica12 tools for pre -processing and *Weka* for classification. The importance of pre- processing was determined by analysis of the results obtained by means of classifiers SMO algorithms Naive Bayes and J48.
Keywords: Text Classification, Information Retrieval, Machine Learning.

[1] Mestre em Ciência da Computação pela Universidade Estadual de Londrina-UEL; professor da Faculdade de Tecnologia de Ourinhos-FATEC. E-mail: alex.marino@fatecourinhos.edu.br.
[2] Graduada em Segurança da Informação pela Faculdade de Tecnologia de Ourinhos-FATEC. E-mail: nathaliabeirao@hotmail.com.
[3] Graduado em Segurança da Informação pela Faculdade de Tecnologia de Ourinhos-FATEC. E-mail: ricardo.fabricio@hotmail.com.br.

Introdução

Ao pesquisar no dicionário o sentido do vocábulo recuperação, encontramos: “ato ou efeito de recuperar-se” e, para recuperar: “recobrar o perdido” em outras palavras é o ato ou efeito de se buscar aquilo que procura. No entanto, quando aplicado as áreas relacionadas à tecnologia da informação (TI), tem seu sentido alterado. A recuperação da informação (RI) consiste em recuperar informações a respeito de um assunto desejado, e não simplesmente recuperar documentos que satisfaçam sentenças de consulta (TAN et al., 1999).

Os sistemas de informação (SI) e de RI são essenciais na ciência desta, pois seus objetivos são os de facilitar o acesso ao que necessita. Contudo um dos grandes desafios encontrados na RI é como atender as necessidades de informação do usuário de forma rápida e precisa. Várias pesquisas foram e continuam sendo realizadas com o propósito de aumentar a precisão dos resultados de forma que o usuário possa encontrar os documentos que atendam as suas necessidades. A grande maioria das pesquisas tem como base o uso das palavras, ou seja, o seu radical, como meio de acesso à informação pelos sistemas automatizados de RI e tem sido a base da maioria, senão todos, os modelos de RI implantados até hoje. Apesar de alguns deles terem alcançado relativo sucesso na melhoria da precisão de resultados de uma busca, a meta principal da RI, que é a obtenção de todos os documentos pertinentes a uma consulta não foi atingida.

Trataremos de forma sucinta, como a categorização de documentos junto com a RI vem simplificando o método de busca de informação, de uma forma geral abordaremos o processo para a realização da categorização de documentos, incluindo a coleta de dados, o pré-processamento e todos os seus respectivos assuntos.

O objetivo deste trabalho é demonstrar a importância, em mineração de texto, em identificar tópicos relevantes considerando um conjunto de documentos (*corpus*) e então vincular a uma ou mais categorias predefinidas. As regras para enquadrar estes documentos e as categorias são determinadas de forma manual por especialistas. Essas regras, ou conjunto de regras, especificam características no qual os documentos devem satisfazer para adequar-se a determinada categoria.

Para realizar esta classificação por meio de aprendizado de máquina supervisionado, é necessário executar diversas operações de forma a torná-lo mais conciso. Tais operações recebem o nome de pré-processamento e consistem basicamente na *tokenização*, eliminação de termos irrelevantes, normalização morfológica, identificação de sinônimos e criação de índices.

O produto final deste trabalho será a demonstração da utilização de técnicas de RI para classificar documentos, cujos dados foram extraídos manualmente dos sites ESPN[4] e CNN[5] 2, totalizando 50 notícias de Esportes e 50 de Políticas, submetendo-os às técnicas de pré-processamento resultando num vetor de palavras onde aplicaremos um classificador probabilístico (JOHN; LANGLEY, 1995). Como resultado final obteve-se acurácia superior a 90%, resultado condizente com a literatura.

O restante deste trabalho divide-se nas seguintes seções:

A seção 1 é constituído pela revisão da literatura, obtido por meio de pesquisa científica com intuito de fornecer uma base de conhecimento para realizar os procedimentos propostos no objetivo deste trabalho. Nele é abordado de maneira sucinta a execução dos processos responsáveis pela realização da categorização de documentos, ou seja, o pré-processamento e seus processos, além de descrever resumidamente as ferramentas utilizadas.

Na seção 2 serão abordadas as etapas necessárias para a execução deste trabalho, citando todos os processos necessários para obtenção dos resultados que serão discutidos na próxima seção.

Na seção 3 será feita uma análise dos resultados obtidos. Serão expostos e discutidos os dados obtidos por meio da execução da categorização de documentos supervisionada.

Por fim, a seção 4 contém a conclusão da análise realizada na seção anterior, nele está descrita a opinião dos autores, baseada nos resultados obtidos, sobre a importância do pré-processamento na categorização de documentos, além de uma sugestão para uma futura continuação deste trabalho.

1 Revisão da Literatura

Esta seção tem o intuito de fornecer uma base de conhecimento para realizar os procedimentos propostos no objetivo deste trabalho. Nele é abordado de maneira sucinta a execução dos processos responsáveis pela realização da categorização de documentos, ou seja, o pré-processamento e seus processos, além de descrever resumidamente as ferramentas utilizadas.

1.1 Categorização de Documentos

Lopes (2004) explica que o objetivo da categorização, em mineração de texto, é de identificar tópicos relevantes em um documento e assim então vincular a uma ou mais

[4] http://espn.go.com/.
[5] 2 http://edition.cnn.com/.

categorias predefinidas. As regras para enquadrar estes documentos e as categorias são determinadas de forma manual por especialistas. Essas regras, ou conjunto de regras, especificam características no qual os documentos devem satisfazer para adequar-se a determinada categoria.

Para que a categorização de documentos seja feita de maneira eficaz faz-se necessário que haja uma preparação do texto a ser categorizado, porém, a implementação e a execução de um processo de categorização de documentos não é uma tarefa fácil, as ferramentas de mineração de textos estão em processo de edificação e, ainda, carecem de um elevado conhecimento técnico para a sua utilização, além de estarem atreladas à linguagens em que os documentos estão escritos. Dado isto Junior (2007) destaca que o processo de categorização de textos organiza-se em cinco etapas encadeadas na seguinte ordem:

- Coleta de dados;
- Pré-processamento:
    * Tokenização;
    * Análise léxica;
    * Remoção de termos irrelevantes (*Stopwords*);
    * Normalização morfológica (*stemming*);
    * Identificação de sinônimos;
- Indexação;
- Categorização;
- Análise dos Resultados.

Em uma breve análise podemos observar que a categorização de documentos é realizada por meio da coleta de dados, cujo objetivo é a formação da coleção de documentos, em seguida é realizado o pré-processamento tendo por finalidade estruturar os documentos realizando diversas operações sobre o texto, e logo após a etapa de indexação é realizada, criando índices afim de estruturar e garantir rapidez e agilidade na recuperação dos documentos e seus termos.

Em seguida é realizada a categorização em si, obtida por meio de métricas que determinam o peso de um termo dentro do documento. E por fim são realizadas as análises responsáveis pela avaliação e interpretação de todo conhecimento obtido pelo processo.

1.2 Coleta

A coleta é responsável pela aquisição dos elementos sob os quais se apoiam o restante do trabalho e, segundo Schiessl (2007), é a primeira etapa a ser realizada na categorização de

documentos. Junior (2007) descreve que esta etapa envolve a seleção de textos que irão compor o *corpus* e cita que a origem dos documentos pode ser adquirida das mais variadas fontes, desde pasta de arquivos em discos rígidos à tabelas de diversos tipos de banco de dados e da internet.

Para realização da coleta podem ser aplicadas diversas técnicas, e diferenciando-se principalmente pelo grau de automatização e também deve-se observar o formato e a língua em que o documento foi escrito pois diversos aplicativos de agrupamentos de textos estão atrelados ao idioma do documento.

1.3 Pré-Processamento

Para Han, Kamber e Pei (2006) o pré-processamento tem a finalidade de melhorar a qualidade dos dados, aperfeiçoando a precisão e a eficiência dos processos de mineração subsequentes. Sendo assim, o pré-processamento constitui-se da aplicação de várias técnicas para captação, organização, tratamento e a preparação dos dados. É uma etapa que possui fundamental relevância na categorização de documentos pois compreende desde a correção de dados errados até o ajuste da formatação dos dados para os algoritmos de mineração de dados que serão utilizados.

Segundo Silva (2004) o pré-processamento consiste na execução das seguintes etapas:

- Tokenização;
- Análise léxica;
- Normalização morfológica;
- Remoção e termos irrelevantes;
- Identificação de sinônimos.

Silva (2004), ainda complementa que estas etapas são executadas com intuito de reduzir a quantidade de termos, visando uma melhor representatividade dos documentos no desempenho do sistema.

Os itens a seguir irão descrever de forma rápida e objetiva as etapas que formam o pré-processamento.

1.3.1 Tokenização

Segundo Gomes (2009) o primeiro passo para o pré-processamento de texto escrito é a tokenização ou atomização. Junior (2007) complementa que o processo de tokenização tem como finalidade extrair termos a partir de um texto livre. Esses termos recebem o nome de

Token e podem ser formados por palavras simples ou compostas, ou sequencias de "n" caracteres. Podemos citar como exemplo:

- Datas numéricas "26/04/2015";
- Números com casas decimas "1000,00";
- Siglas como PM, "Policia Militar";
- Abreviações como: Ex. "Exemplo".

Gomes (2009) explica que o processo de tokenização é auxiliado pelo fato de palavras serem separadas por espaços ou sinais de pontuação; e complementa que este processo pode ser bastante complexo para um computador pelo fato de alguns delimitadores serem utilizados em mais de uma função.

Podemos citar como exemplo o "ponto" que é usado em abreviações, o travessão que pode ser usado em contas e em citações e a virgula que pode ser usada em informações numéricas.

Neste caso, BARCALA (2002) apud Conceição (2013) menciona que a tokenização pode utilizar dois dicionários, um de abreviaturas e outro de acrônimos, além de uma coleção de regras com o propósito de identificar estas datas, números, siglas e abreviações.

Gomes (2009) complementa que a Tokenização baseada em delimitadores não se aplica a algumas línguas, como árabe, chinês ou japonês, pois nestas línguas não se usam sequer espaços entre os caracteres, uma vez que um caractere pode significar toda uma ideia.

1.3.2 Análise Léxica

Esta fase é responsável pela execução de um analisador léxico com a função de identificar as palavras presentes no texto, ignorando símbolos, caracteres de controle de arquivos ou formatação. A tabela 1 mostra um exemplo do funcionamento de um analisador léxico.

- Aplicação do case *folding* (maiúscula/minúscula);
- Correção de múltiplos espaços e tabulações por um único espaço;

Tabela 1 – Aplicação de um analisador léxico

Identificação de termos válidos	
Documento original	Documento normalizado
... œæß Na maioria das vezes os documentos retornados pelas ferramentas de recuperação da informações ,(ou Mineração de dados), envolvem um contexto mais amplo, fazendo com que o usuário tenha que garimpar, ou seja, especificar ou filtrar estes documentos < o que demanda tempo e conhecimento > a fim de obter a informação que ele realmente necessita... (Texto desenvolvido no LATEX 2ε)	na maioria das vezes os documentos retornados pelas ferramentas de recuperação de informações ou mineração de dados envolvem um contexto mais amplo fazendo com que o usuário tenha que garimpar ou seja especificar ou filtrar estes documentos o que demanda tempo e conhecimento a fim de obter a informação que ele realmente necessita texto desenvolvido no látex.

Fonte: Sistemas inteligentes: fundamentos e aplicações (EBECKEN et al., 2003).

- Padronização de datas e números;
- Remoção de hífens.

Pode-se fazer a utilização tendo de um dicionário para a validação das sequências de caracteres e correção de possíveis erros ortográficos, quanto de um thesaurus ou dicionário de sinônimos para auxiliar na normalização da palavra.

1.3.3 Remoção de termos irrelevantes (*Stopwords*)

Para Ebecken et al. (2003) um dos primeiros passos na preparação de dados, é a questão do que pode não ser levado em conta, mediante o processamento de dados, ou seja, palavras de baixa representatividade, uma vez que nada acrescentam, podendo ser preposições, artigos e advérbios, tal conjunto é chamado de Stop *Words*. Suas remoções dos índices gerados em sistemas de RI normalmente visam:

- Diminuir o tamanho do índice;
- Tornar mais rápidas as consultas às frases que envolvam *Stopwords*;
- Melhorar a qualidade dos resultados.

De acordo com Passini (2012) existe uma relação entre a frequência das palavras e sua importância para o entendimento do contexto das informações. Roncero (2010), alega que é possível reduzir até 50% o tamanho de um documento removendo termos considerados irrelevantes ao contexto, tais como: pronomes, artigos, preposições e interjeições, reduzindo significativamente a quantidade de termos e diminuindo o custo computacional das próximas etapas do pré-processamento.

Além dessas, segundo Wives (2002) apud Passarin (2005), outras palavras que são tidas como irrelevantes, são aquelas que aparecem com frequência na coleção de documentos. Desta forma, são consideradas incapazes de discriminar os mesmos, tornando-se desnecessária a permanência destas na estrutura de índices.

Conforme Ebecken et al. (2003), no processo de eliminação de *Stopwords* são analisados um documento e uma lista de *Stopwords*, o que resulta assim em uma eliminação de palavras consideradas desnecessárias no texto. Na tabela 2 podemos analisar como é dado esse processo.

Tabela 2 – Remoção dos *Stopwords*

Remoção de *Stopwords*	
Documento normal	Documento sem *Stopwords*
Primeiro encontro dos estudantes de Segurança da Informação da Faculdade de Tecnologia de Ourinhos. Uma iniciativa pioneira dos formandos de 2015 no intuito de promover a troca de experiências dos alunos	Primeiro encontro estudantes Segurança Informação Faculdade Tecnologia Ourinhos. Iniciativa pioneira formandos 2015 intuito promover troca experiências alunos formados entraram mercado trabalho. Assunto

formados com os que entraram no mercado de trabalho. Todos assuntos tratados no decorrer do evento estarão a disposição dos nossos alunos na biblioteca da instituição por meio de ata.	tratado decorrer evento estarão disposição alunos biblioteca instituição meio ata.

Fonte: Sistemas inteligentes: fundamentos e aplicações (EBECKEN et al., 2003).

1.3.4 Normalização Linguística (*Stemming*)

Roncero (2010) explica que o *Stemming* é uma técnica que busca reduzir variâncias em um termo, ou seja, consiste numa normalização linguística onde as formas variantes de um termo são reduzidas a uma forma comum.

Abrange em identificar os radicais de uma palavra reduzindo assim a quantidade de termos, permitindo transformar estas em elementos mais simples, emoutras palavras, o *stemming* lida com a remoção de prefixos ou sufixos de um termo, como também a eliminação dos plurais de determinadas palavras. Ou até mesmo na transformação de um verbo para sua forma no infinitivo. Além da diminuição quanto aos termos, a técnica possibilita que o usuário não se preocupe com a forma ortográfica na qual a palavra foi escrita.

A tabela 3 mostra um exemplo de aplicação do *stemming* na palavra considerar.

Tabela 3 – Aplicação do *Stemming*

Aplicação do *Stemming*	
Quatro Palavra	Uma Palavra (seu radical)
Considerar Considerado Consideração Considerações	Consider

Fonte: Sistemas inteligentes: fundamentos e aplicações (EBECKEN et al., 2003).

Entretanto, existe uma dificuldade na técnica de *stemming* em relação a outros idiomas existentes, por ser uma técnica que trabalha com variações na estrutura da palavra levando-se em conta que cada língua deve ser considerada, foram desenvolvidos ou adaptados vários algoritmos de *stemming* dentre os quais podemos citar os algoritmos representados na tabela 4, onde podemos observar a predominância do algoritmo de Porter em vários idiomas.

Lopes (2004) menciona que os algoritmos de *stemming* correntes são precários no uso de informação para determinar o sentido correto de cada palavra, o que por sua vez não ajuda muito, onde a maioria destas podem vir a apresentar um único significado, o que no geral não compensam nos ganhos obtidos pelo processo de *stemming*. Acerca destes, existem outros tipos de erros que devem ser observados durante a execução do *stemming*, associado a dois grupos:

- Over*stemming*: remoção não apenas do sufixo, como também uma parte do radical;

Tabela 4 - Algoritmos de radicalização (*Stemming*)

Algoritmos de radicalização (*Stemming*)		
Língua	Algoritmo	Autoria
Inglês	Porter KStem Paice/Husk Porter 2 Dawson	Porter Krovetz Paice e Husk Porter Dawson
Português	Porter - Português Orengo Pega*stemming*	Porter Orengo Gonzales
Alemão	Porter - Alemão Porter - Alemão Variação	Porter Porter
Amárico (Etíope)	Alemayehu-Willett	Alemayehu e Willett
Búlgaro	BulStem	Nakov
Dinamarquês	Porter - Dinamarquês	Porter
Esloveno	Popovic-Willet	Popovic e Willet
Espanhol	Porter - Espanhol	Porter
Finlandês	Porter - Finlandês	Porter
Francês	Porter - Francês	Porter
Holandês	Porter - Holandês Kraaj-Pohlmann	Porter Kraaj e Pholmann
Italiano	Porter - Italiano	Porter
Latim	Schinke et al.	Schinke et al.
Norueguês	Porter - Norueguês Carlberger et al.	Porter Carlberger et al.
Russo	Porter - Russo	Porter
Sueco	Porter - Sueco	Porter
Turco	Ekemekçiogli et al.	Ekemekçiogli et al.

Fonte: Uma revisão dos algoritmos de radicalização em língua portuguesa (VIERA; VIRGIL, 2006)

- Under*stemming*: não remoção do sufixo, ou quando feito se é reduzido apenas uma parte deste, causando falha nas palavras, e na recuperação de documentos que seriam pertinentes.

Segundo Viera e Virgil (2006) a língua portuguesa apresenta diversas razões para complicar o processo de radicalização, principalmente no que se refere à morfologia, citando:

- o número de exceções, devido ao uso comum de sufixos como terminações de palavras;
- a irregularidade na conjugação dos verbos;
- mudanças no radical morfológico, como no caso de emissão e emitir;
- uso frequente de termos estrangeiros;
- o uso de nomes próprios, que não deveriam ser radicalizados.

Os melhores algoritmos para a realização do *stemming* na língua portuguesa são o algoritmo de Porter (adaptado) e o algoritmo de Orengo. A seguir uma breve descrição de como funcionam estes algoritmos.

Algoritmo de Porter

De acordo com Ebecken et al. (2003), o processo de *stemming* de Porter avalia que a remoção dos sufixos seja mais importante que a dos prefixos, uma vez que esse método remove 60 sufixos diferentes, promovendo uma transformação no *stem*. Segundo Viera e Virgil (2006) o algoritmo de Porter adaptado à língua portuguesa é composto de cinco etapas, sendo elas:

1. Remoção dos sufixos;
2. Remoção dos sufixos verbais, caso a regra 1 não tenha realizado nenhuma alteração na palavra;
3. remoção do sufixo i, se precedido de c no final da palavra;
4. Remoção dos sufixos residuais os, i, o, á, í, ó regra esta executada caso as regras 1 e 2 não tenham alterado a palavra;
5. Remoção dos sufixos e, é, ê, tratamento da cedilha e tratamento das silabas gue, gué, guê .

Para que a realização destas etapas seja feita de forma satisfatória primeiramente são tratadas as vogais nasalizadas ã e õ que após a realização do processo são retornadas a sua forma normal.

Algoritmo de Orengo

De acordo com Viera e Virgil (2006) o algoritmo de Orengo é baseado em regras de remoção de sufixos, entretanto algumas regras apresentam exceções com tratamento baseado em dicionário de termos, que ocasionam uma diminuição na ocorrência de *overstemming*. O algoritimo de Orengo é executado obedecendo as seguintes etapas:

1. Remoção do plural

   Remove-se o final -s indicativo de plural de palavras que não se constituem em exceções à regra, realizando modificações, quando necessário. As exceções a regra se dão por palavras terminadas em s que não constituem plural como exemplo podemos citar a palavra gis

2. Redução do feminino

   Remove-se o final -a de palavras femininas com base nos sufixos mais comuns, as palavras são transformadas para o gênero masculino, como exemplo podemos citar: solteira para solteiro.

3. Redução adverbial.

Remove-se o final -mente de palavras que não se constituem em exceção.

4. Redução do aumentativo/diminutivo

Removem-se os indicadores de aumentativo e diminutivo mais comuns, transformando as palavras para o modo superlativo ou normal.

5. Redução nominal

Removem-se 61 sufixos possíveis para substantivos e adjetivos. Alguns autores, em sua implementação do algoritmo, expandiram o número de sufixos para 84 eliminado as 6 e 7 caso a palavra sofra alteração neste passo.

6. Redução verbal

Reduzem-se as formas verbais aos seus radicais. Esta etapa reflete a complexidade da língua portuguesa, pois os verbos regulares têm mais de 50 formas, sendo cada uma delas com um sufixo diferente. Caso a palavra (verbo) tenha sido alterada nesta etapa deverá ser executado a 8, pulando assim a próxima etapa.

7. Remoção de vogais

Removem-se as vogais a, e e o das palavras que não foram tratadas pelos dois passos anteriores. Desta forma palavras como, por exemplo, garoto que não tenha sido alterada em nenhuma etapa anterior tenha seu radical extraído.

8. Remoção de acentos

Removem-se os sinais diacríticos das palavras, ou seja, este processo remove todas as letras acentuadas por seus equivalentes sem acentuação para que as formas da palavra sejam reduzidas ao mesmo *stem*.

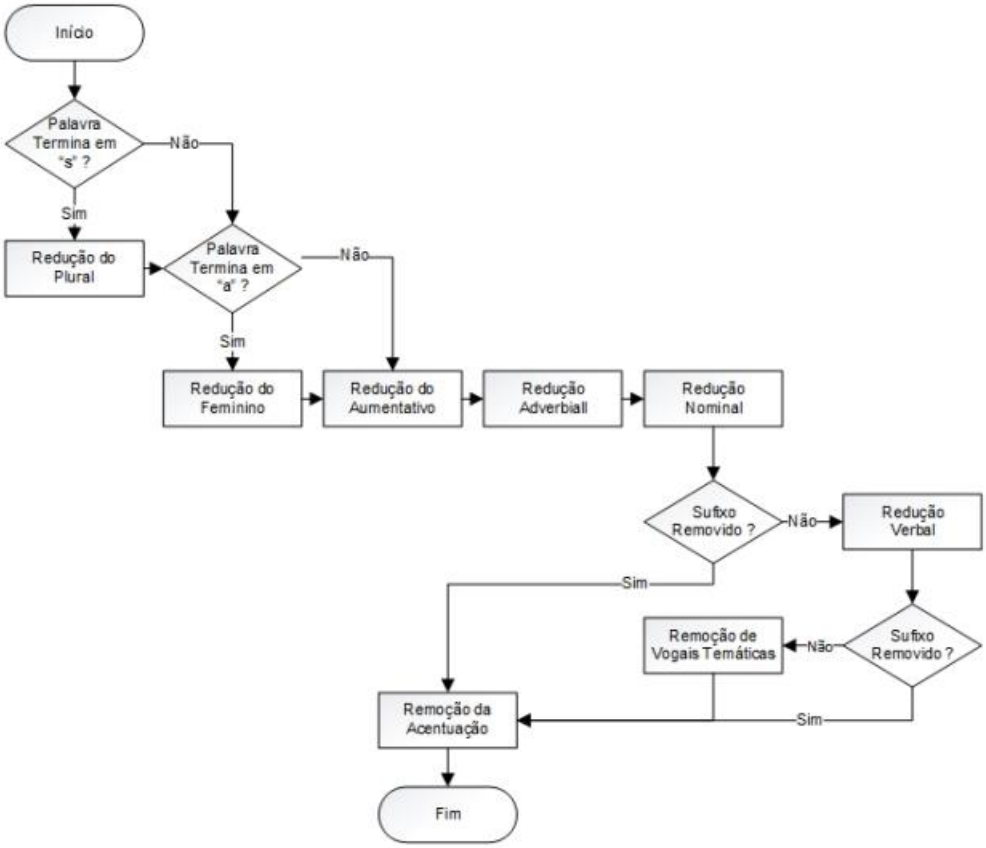
Na figura 1 podemos observar como a sequência de passos do algoritmo é realizada. Note que algumas etapas só serão executadas se determinada característica é encontrada.

Viera e Virgil (2006) cita ainda que o algoritmo de Orengo possui uma pequena vantagem em relação ao algoritmo de Porter reduzindo o vocábulo em 51% enquanto o algoritmo de Porter reduz apenas 44%.

1.4 Indexação

Soares (2008) menciona que a Indexação e normalização tendem a facilitar a identificação de similitude de significado entre as palavras, considerando-se também as variedades morfológicas e situações de sinonímia. Tal processo se resulta na geração de um índice, que se dá no processo de indexação. Para tanto, convém explicar que, indexar significa identificar peculiaridades de uns documentos colocando-as numa estrutura de índice.

Figura 1 - Sequência de passos para o removedor de sufixos1



Fonte: Uma revisão dos algoritmos de radicalização em língua portuguesa (SILVA, 2004)

Conforme diz Martins (2009), localizar as informações usando indexação tende a facilitar vida dos usuários que muitas vezes usam termos específicos de sua área, porém se for utilizado de uma forma diferente a qual foi indexado não haverá eficácia na busca, uma vez que os problemas em relação ao vocabulário serão maiores, em outras palavras permite a eficiência da busca de documentos em textos sem precisar examiná-lo por inteiro.

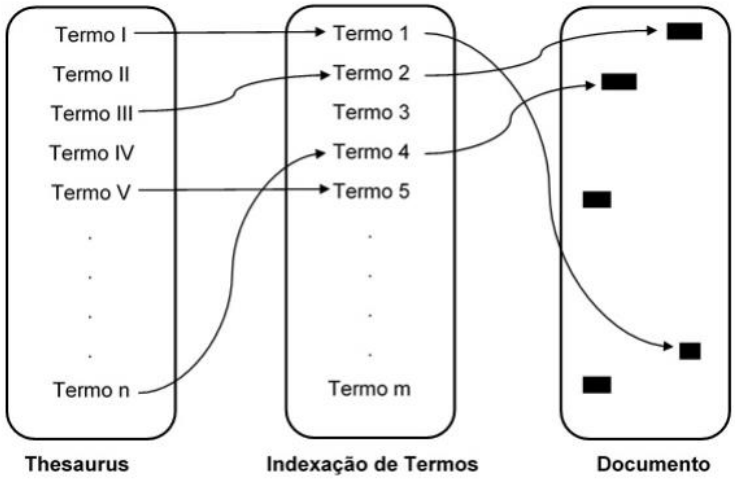
Existem alguns tipos de indexação; há a indexação temática, indexação do texto completo, que são as mais comuns, e há também a indexação tradicional, a indexação por *tags*, a indexação por listas invertidas e a indexação semântica latente. Dentre todas, vamos dar base em três que são as mais usadas.

A Indexação do Texto Completo age automaticamente em várias ferramentas que analisam textos quando documentos são carregados explica Lopes (2004); sobre as informações localizadas dentro de um texto, podemos facilitá-las pelo índice que geralmente guardam as informações, como operadores booleanos constituídos por *and, or*, *not*, e os operadores de proximidade; sendo eles, *near e within*. Assim sendo, a utilização dos operadores age de forma

rápida e eficaz diante de um texto, uma vez que não é necessária a busca de uma palavra no texto todo, podendo utilizar apenas o índice.

Segundo Lopes (2004), a Indexação Temática depende do uso do dicionário. Um conjunto de termos que pode ser definido por um vocabulário usando relacionamentos chamados de Thesaurus, este fornece hierarquicamente uma estrutura na qual se usa ferramentas de *text mining* onde se encontram rapidamente termos específicos como mostrado na figura 2.

Figura 2 - Utilização do Thesaurus na indexação temática



Fonte: Lopes (2004)

A indexação por *tags* age na seleção de algumas partes do texto automaticamente fazendo assim parte do índice. No uso desta indexação são empregados o uso de gramática parsers e expressões regulares para reconhecimento e definição das *tags*. Lopes (2004) diz que as palavras-chave usadas em certa base são extraídas destas *tags*.

Manning (2007) apud Junior (2007) menciona que a fase de indexação é a responsável pela criação dos chamados índices que nada mais são que uma estrutura de dados capazes de permitir que uma consulta seja realizada sem a necessidade de analisar toda uma base de dados.

Junior (2007) explica que os índices são utilizados para aprimorar a velocidade e o desempenho da busca de um documento relevante em relação a um determinado termo buscado, e os compara a um sumário de um livro exemplificando que este é composto de uma lista detalhada, com a informação da localização no texto, dos principais tópicos abordados por este. Destaca-se ainda que a etapa de pré-processamento influencia diretamente o processo de indexação, uma vez que todo o conteúdo a ser indexado, ou não, necessariamente foi obtido por meio desta etapa.

1.5 Avaliação da recuperação

Para avaliar sistemas de RI, é necessário medir o quão bem o sistema atende as necessidades do usuário. Sem uma avaliação, não temos como saber se um sistema de RI está tendo o desempenho desejado e nem podemos comparar a qualidade de sua recuperação em relação a outros sistemas (MOHRI; ROSTAMIZADEH; TALWALKAR, 2012).

Segundo Baeza-Yates et al. (2012) a avaliação da recuperação é um processo sistemático no qual se associa uma métrica quantitativa aos resultados produzidos por um sistema de RI em resposta a um conjunto de consultas de usuários. Essa métrica deve ser diretamente associada à relevância dos resultados para os usuários.

O processo de associar uma métrica numérica é, até hoje, comumente adotado por ser simples, podendo ser repetido diversas vezes a custos relativamente baixos. A repetibilidade permite estudar lotes de consultas cada vez maiores e seus resultados em espaços de tempo relativamente curtos, possibilitando descobrir o que não está funcionando na função de ranqueamento.

Note que nesta seção discutiremos a avaliação da recuperação levando-se em conta apenas a avaliação de qualidade em termos de resultado e não de desempenho de processamento.

1.6 Ferramentas Utilizadas

Para realização dos experimentos contidos neste trabalho foi necessário a utilização de algumas ferramentas, sendo estas o Statistica12 responsável por executar a redução e limpeza do texto contidas na Extração de Características, com o pré-processamento (tokenização, *Stopwords*, *stemming* e indexação) e o *Weka*, responsável pela classificação do texto por meio da execução dos algoritmos classificadores Naive Bayes, SMO e J48. A seguir uma breve descrição destas ferramentas.

Statistica12

STATISTICA[6] é um pacote de estatísticas e análise de software desenvolvido pela *StatSoft* em 1991. Com o passar dos anos o software vem recebendo atualizações e melhorias em suas versões, chegando em 2013 em sua versão mais atual o *Staistica12* que está disponível em diversos idiomas.

[6] http://www.statsoft.com/Products/STATISTICA/Data-Miner.

O software inclui uma matriz de análise de dados, gerenciamento de dados, visualização de dados e procedimentos de mineração de dados; bem como uma variedade de modelos de previsão, *clustering*, classificação e técnicas exploratórias.

O Statistica12 pode ser adquirido diretamente no site da *StatSoft* com uma licença *trial* para degustação. A sua instalação depende de um cadastro e validação online.

Weka

WEKA[7], é um software livre de código aberto, com licença GNU, desenvolvido na linguagem JavaTM em 1997 na Universidade de Waikato (Nova Zelândia).

Este software consiste em uma coleção de algoritmos de aprendizado de máquina para minerar dados, contendo ferramentas para pré-processamento de dados, classificação, regressão, clusterização, regras de associação e visualização, além de servir como base para o desenvolvimento de novos sistemas de aprendizagem.

Ainda que concebido para trabalhar com dados estruturados, o aplicativo conta com um filtro capaz de efetuar a tokenização e remoção de *Stopwords* de documentos textuais. Além de executar o cálculo de relevância dos termos *tokenizados* de acordo com as métricas TF-IDF e capaz de realizar a operação Case Folding, clusterização e classificação de documentos entre outros.

O *WEKA* pode ser adquirido diretamente no site da universidade de Waikato e para executá-lo é necessário obter um *Java Runtime Environment* (JRE) instalado em seu computador. A instalação do *WEKA* é rápida e simples.

1.7 Aprendizado de Máquina

O aprendizado de máquina segundo McCarthy é uma área ampla da Inteligência Artificial preocupada com o projeto e o desenvolvimento de algoritmos que aprendem padrões presentes nos dados fornecidos como entrada.

Simon 1983 apud Conduta e Magrin (2010), define aprendizado como qualquer mudança num sistema que melhore o seu desempenho na segunda vez que ele repetir a mesma tarefa, ou outra tarefa da mesma população.

Wang, Ma e Zhou (2009) apud Bernardes (2010), define o Aprendizado de Máquina (AM) como o estudo da utilização de computadores para simular atividades humanas de

[7] http://www.cs.waikato.ac.nz/ml/weka/.

aprendizagem e desenvolver métodos auto incrementais de obtenção de novos conhecimentos e novas habilidades e identificação de conhecimento já existente.

De acordo com Xue e Zhu (2009) apud Souza (2014), AM é o estudo de como o computador pode realizar e/ou simular o comportamento de aprendizagem do ser humano. O objetivo é obter novos conhecimentos e novas habilidades e organizar a estrutura do conhecimento, que pode ajudar num processo progressivo de aprendizagem. Seguindo esta linha, modelos de computação e de entendimento são criados baseados em pesquisas nas áreas humanas de psicologia e ciência cognitiva.

Os estudos sobre aprendizado de máquina dividem-se em três grupos básicos: aprendizagem supervisionada, não-supervisionada e por reforço, as quais serão abordadas com mais detalhes a seguir:

Segundo Bigus (1996) apud Conduta e Magrin (2010, p. 5), o aprendizado supervisionado é utilizado quando, em um banco de dados, se tem tanto as perguntas como as respostas. Usado para a realização de treinamento de redes neurais na obtenção de classificação, funções de aproximação ou modelagem e previsões baseadas no tempo. Fornece a resposta "certa" durante o treinamento.

No aprendizado não-supervisionado segundo Conduta e Magrin (2010), existe a dúvida sobre a saída esperada, desta forma, se utilizam métodos probabilísticos para simular uma experiência não vivida. Para realizar tais procedimentos, é amplamente difundida a utilização da aprendizagem bayesiana ou redes bayesianas.

O aprendizado por reforço é baseado em dados de um ambiente completamente observável. Sua meta é aprender o quanto a política é boa, ou seja, descobrir a sua utilidade.

1.8 Aprendizado supervisionado

No aprendizado supervisionado, o objetivo é induzir conceitos a partir de exemplos que estão pré-classificados, ou seja, exemplos que estão rotulados com uma classe conhecida. Se as classes possuírem valores discretos, o problema é categorizado como classificação. Caso as classes possuam valores contínuos, o problema é categorizado como regressão.

Segundo Bigus (1996) apud Conduta e Magrin (2010), O aprendizado supervisionado é utilizado quando, em um banco de dados, se tem tanto as perguntas como as respostas. Usado para a realização de treinamento de redes neurais na obtenção de classificação, funções de aproximação ou modelagem e previsões baseadas no tempo.

1.9 Algoritmos de Classificação

Na literatura são encontrados diversos algoritmos classificadores que podem ser empregados na classificação de texto, no entanto descreveremos, orma sucinta, apenas os algoritmos Naive Bayes, J48 e SMO utilizados em nossos experimentos.

De Acordo com Rodrigues (2009), Naive Bayes é um classificador ingênuo probabilístico, fundamentado no teorema de Bayes a fim de definir a classe de maior probabilidade para cada instância a ser classificada.

Segundo Oguri (2006) o classificador Naive Bayes modelo multinomial é provavelmente o classificador mais utilizado em aprendizado de máquina.

Neste modelo é assumido que cada documento é representado por um vetor de atributos inteiros caracterizando o número de vezes que cada característica ocorre no documento.

De acordo com Brilhadori et al. (2013) o algoritmo J48, proposto por Quilian em 1993, é um indutor top down de árvores de classificação baseado no algoritmo C4.5. A seleção da melhor partição dos nós e o critério de parada são baseados na entropia de Shannon, como é usual em parte da família de indução de árvores de classificação.

O SMO (Sequential Minimal Optimization), conforme Gevert et al. (2009), surgiu da necessidade de implementação de um algoritmo SVM de maneira rápida, simples e capaz de tratar conjuntos de dados mais extensos.

Além disso, possui a capacidade de tratar um conjunto de dados esparsos, que possuem um número substancial de elementos com valor zero.

De acordo com Gevert et al. (2009) o SMO foi proposto por Plat em 1998, é um algoritmo de aprendizado de máquina que utiliza duas variáveis em cada iteração onde se propõe a resolver o problema de programação quadrática do SVM sem o armazenamento de matrizes extras e sem o uso de métodos de solução numérica, ou seja, apresenta uma solução analítica.

1.10 Matriz Confusão

Silva (2005) descreve a matriz confusão como sendo uma técnica empregada para analisar o desempenho de sistemas classificadores enquanto Anacleto (2009) explica que a Matriz Confusão é uma tabela para visualização dos resultados, onde cada linha da matriz representa as instâncias reais de uma classe, enquanto cada coluna da matriz representa as instâncias previstas de uma classe.

Um dos benefícios da matriz de confusão se dá pela facilidade de análises, principalmente se o sistema contém apenas duas classes. No caso de sistemas que contemplam mais do que

duas classes, estas podem ser reduzidas a duas. Desta forma, podemos considerar que a matriz de confusão é uma tabela com duas linhas e duas colunas que regista o número de Verdadeiro Negativo (VN), Falso Positivo (FP), Falso Negativo (FN) e Verdadeiro Positivo (VP).

Carvalho et al. (2011) apresenta na tabela 5 um exemplo de matriz de confusão para um classificador baseado em duas classes distintas onde VP, FN, FP e VN são assim definidas:

- VP corresponde ao número de objetos da classe positiva classificando-os corretamente;
- FN corresponde ao número de objetos pertencentes à classe positiva que foram incorretamente atribuídos à classe negativa;
- FP corresponde ao número de objetos cuja classe verdadeira é negativa, mas que foram classificados incorretamente como pertencentes à classe positiva;
- VN corresponde ao número de objetos da classe negativa classificados corretamente.

Tabela 5 – Matriz Confusão

.

Matriz Confusão		
	Previsão Positiva	Previsão Negativa
Caso Positivo	VP	FN
Caso Negativo	FP	VN

Fonte: Carvalho et al. (2011)

Há quatro combinações possíveis, estando as combinações corretas na diagonal principal da matriz e as combinações incorretas na diagonal secundária. Sendo assim os valores de VP e VN correspondem as respostas corretas e os valores de FP e FN correspondem as respostas incorretas.

No processo de classificação pode ocorrer de haver muitos casos classificados como negativo (incorreto) poucos positivos (corretos). Nestes casos pode-se utilizar os parâmetros VP, FN, FP e VN para calcular precisão (*precision*), revocação (*recall*), medida F (F-*Measure*), acurácia (*acuracy*), razão verdadeiro positivo (TP Rate), razão falso positivo (FP Rate), razão verdadeiro negativo (TN Rate) e posição ROC (ROC area) associados ao classificador conforme descrito em Olson e Delen (2008).

Precisão

A precisão (*precision*) é a proporção de casos positivos que foram corretamente identificados pelo classificador. Esta medida é calculada por meio da equação:

$$Precisão = \frac{VP}{VP + FP}$$

Revocação

A Revocação (Recall) é a proporção de casos corretamente classificados como positivo e a quantidade de casos que deveriam ter sido classificados como positivos. Esta medida é calculada por meio da equação:

$$\text{Revocação} = \frac{\text{VP}}{\text{VP} + \text{FN}}$$

Medida F

A Medida F (*Fl-Measure*) é definida como a medida harmônica entre os Valores Revocação e Precisão. Esta medida é calculada por meio da equação:

$$\text{F1} - Measure = \frac{2 * recall * precision}{Recall + precision}$$

Acurácia

A acurácia é o acerto do sistema considerando a proporção de instâncias corretamente classificadas no total dos registros. Esta medida é calculada por meio da equação:

$$\text{Acurácia} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{VN} + \text{FP} + \text{FN}}$$

Razão Falso Negativo (RFN)

Consiste na proporção dos casos positivos que foram classificados incorretamente como negativos, isto é, o indivíduo ser evento (Y=1) dado que o modelo classificou o indivíduo como não evento ($Y\hat{} = 1$). Esta medida é calculada por meio da equação:

$$\text{RFN} = \frac{\text{FN}}{\text{FN} + \text{VP}}$$

Razão verdadeiro negativo (RVN)

Consiste na proporção dos casos negativos que foram classificados corretamente tal, ou seja, o indivíduo ser não evento (Y=0) dado que o modelo o classificou como não evento ($Y\hat{} = 0$). Esta medida é calculada por meio da equação:

$$\text{RVN} = \frac{\text{VN}}{\text{VN} + \text{FP}}$$

Razão Falso Positivo (RFP)

A razão falso positivo (FP *Rate*) consiste na proporção de casos classificados erradamente como positivos. Esta medida é calculada por meio da equação:

$$\mathrm{RFN} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{VN}}$$

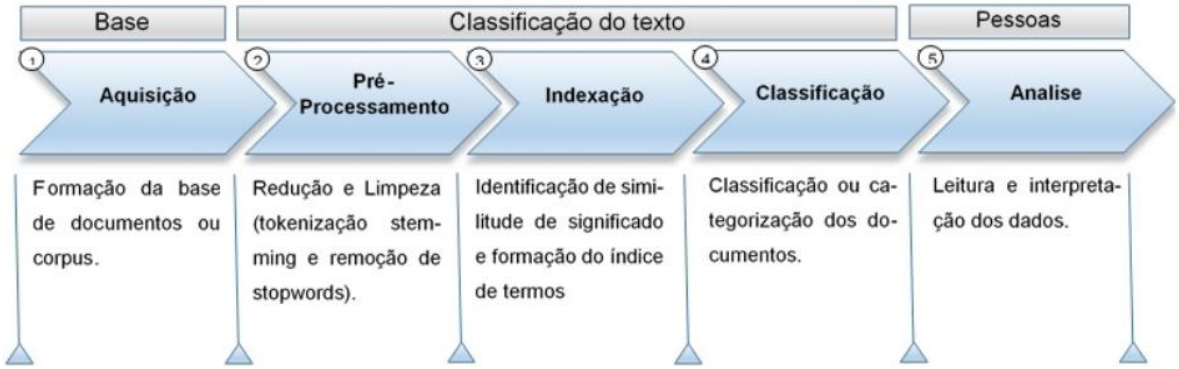
Área ROC

Segundo Camargo (2010) a área ROC representa a sensibilidade (calculada em revocação) e o complemento da especificidade (calculada em RVN) em um gráfico para sistema de classificação binário, cujo limiar de distinção entre as duas classes é variável. A área ROC apresenta a relação custo (especificidade) x benefício (sensibilidade) dos modelos à medida que o limiar é alterado.

2 Materiais e Métodos

A realização dos experimentos contidos neste trabalho obedece a uma ordem cronológica descrita na figura 3, determinada por meio de pesquisa cientifica em livros, dissertações, teses e artigos.

Figura 3 – Descrição das etapas do experimento



Fonte: Autores

Para este experimento foram utilizados como fonte de dados os sites da ESPN e CNN de domínio público totalizando um conjunto de 200 documentos com um vocabulário de 22280 termos distintos. A redução e limpeza, contidas na Extração de Características, com o pré-processamento reduziu o vocabulário para quantidade de 15426 termos formando o *corpus*. O detalhamento do conjunto de documentos por categoria pode ser observado na tabela 6.

Tabela 6 – Informações do *Corpus*

Informações da composição do *corpus* por categoria				
Quantidade de:	Notícia	Basquete	Beisebol	Futebol
Documentos	50	50	50	50
Termos no *corpus*	66125	53963	71143	81720
Termos distintos	6632	4624	5672	5952
Termos distintos processados	4415	3044	3799	4168
Termos Distintos removidos	2217	1580	1873	1784

Fonte: Os autores.

Para realizar extração de características foi utilizada a ferramenta Statistica12[8] que executou o pré-processamento e a indexação, resultando no índice de termos utilizado na classificação dos documentos por meio da ferramenta *Weka*[9], com os algoritmos classificadores Naive Bayes, J48 e SMO.

Para avaliar a eficiência de nosso experimento, utilizamos 4 equações estatísticas, muito conhecidas, encontradas em Olson e Delen (2008) conforme exibidas na Tabela 7.

Tabela 7 – Equações utilizadas

Nome Equação	
Nome	Equação
Precisão	$Precisão = \frac{VP}{VP + FP}$
Acurácia	$Acurácia = \frac{VP + VN}{VP + VN + FP + FN}$
Razão Falsos Negativos	$RFN = \frac{FN}{FN + VP}$
Razão Verdadeiros Negativos	$RVN = \frac{VN}{VN + FP}$

Fonte: Olson e Delen (2008)

Onde precisão significa o índice de acerto na classificação notícias em suas devidas categorias. A acurácia é o acerto do sistema considerando a proporção de instâncias corretamente classificadas no total dos registros. Razão de falso negativo consiste na proporção dos casos positivos que foram classificados incorretamente como negativos enquanto a razão de verdadeiro negativo consiste na proporção dos casos negativos que foram classificados corretamente tal.

A análise dos resultados encontra-se na seção 3.

[8] http://www.statsoft.com/Products/STATISTICA/Data-Miner.
[9] http://www.cs.waikato.ac.nz/ml/weka/.

3 Resultados

Com o objetivo de preparar os documentos para realização de uma das etapas dos experimentos se fez necessário a redução e limpeza do texto por meio da remoção de *Stopwords*, descrito na seção 1.3.3, e do *stemming*, descrito na seção 1.3.4, contidos no pré-processamento. Estas etapas têm por finalidade deixar os documentos mais concisos para realização dos próximos passos.

Como resultado deste processo, realizado por meio do Statistica12, obtemos uma redução significativa de até 97,8% do vocabulário do *corpus* em relação ao total de termos, conforme podemos observar na tabela 8.

Tabela 8 – Proporção de remoção de termos

Proporção da remoção de termos em relação ao *corpus*			
Notícia	Basquete	Beisebol	Futebol
Termos distintos	89,97%	91,43%	92,03%
Termos distintos processados	93,32%	94,36%	94,66%
Termos distintos removidos	96,65%	97,07%	97,37%

Fonte: Os autores.

Para dar sequência nos experimentos foram realizados, ainda por meio do Statistica12, o restante das etapas do pré-processamento, ou seja, a tokenização descrito na seção 1.3.1, e a indexação descrito na seção 1.4, que finalizaram a preparação do *corpus* para realização da classificação de documentos executada por meio do *Weka*.

Como resultado da classificação de documentos contemplando os três métodos de classificação (SMO, Naive Bayes e J48) aplicados sob dois *corpus* (pré-processado e não pré-processado) obteve-se a matriz confusão que pode ser observadas nas tabelas 9,10,13, 14, 17 e 18.

Das matrizes calculou-se a acurácia, precisão conforme descrito na seçao 1.10, constantes nas tabelas 11, 12, 15, 16, 19 e 20.

Pode-se verificar que exceto para Naive Bayes, os demais métodos de classificação apresentaram ganho de acurácia no *corpus* pré-processado em relação ao *corpus* não pré-processado conforme podemos constatar nas figuras 7 e 8.

Mesmo com perda de acurácia e precisão em Naive Bayes, pode-se constatar que com relação à performance, todos os métodos apresentaram ganho significativo, conforme podemos observar na figura 9 que contém o gráfico de performance.

3.1 Avaliação com CNB

Atendendo às premissas do objetivo do trabalho o classificador Naive Bayes (CNB) foi utilizado na classificação do *corpus* não pré-processado e pré-processado e como resultado obteve-se a matriz confusão para ambos *corpus* descritos na tabelas 9 e 10 respectivamente.

Tabela 9 – Matriz Confusão - CNB não pré-processado

CNB não pré-processado				
	Beisebol	Basquete	Notícia	Futebol
Beisebol	48	1	1	0
Basquete	1	47	1	1
Notícia	3	2	44	1
Futebol	2	4	2	42

Fonte: Os autores

Tabela 10 – Matriz Confusão - CNB pré-processado

CNB pré-processado				
	Beisebol	Basquete	Notícia	Futebol
Beisebol	48	0	1	1
Basquete	2	46	1	1
Notícia	2	1	44	3
Futebol	3	2	5	40

Fonte: Os autores

Com base nos dados fornecidos pelas matrizes confusão obteve-se as tabelas 11 e 12 de precisão por classe, onde foi possível determinar a precisão e a acurácia, além de identificar qual categoria o algoritmo obteve melhor desempenho.

Tabela 11 – Precisão por classe com CNB não pré-processado

Detalhamento da precisão por classe do CNB não pré-processado						
	RVP	RFP	Precisão	Revocação	Medida F	Área ROC
Beisebol	0,96	0,04	0,889	0,96	0,923	0,976
Basquete	0,94	0,047	0,87	0,94	0,904	0,977
Notícia	0,88	0,027	0,917	0,88	0,898	0,971
Futebol	0,84	0,013	0,955	0,84	0,894	0,967
Média	0,905	0,032	0,908	0,905	0,905	0,973

Fonte: Os autores.

Analisando os resultados expostos na tabela 11 para classificação não pré-processada, podemos observar que o algoritmo NB obteve um ótimo desempenho na classificação dos documentos das categorias beisebol e basquete com taxa de acerto acima de 90% no entanto o houve uma perda 12% de desempenho na classificação dos documentos referente a futebol onde apenas 84% foram classificados corretamente.

O Classificador NB apresenta comportamento parecido na classificação não pré-processada, descrita na tabela 12 com taxas de acerto de 94% nas categorias Basquete e Beisebol enquanto apresenta queda de 12% na categoria Futebol.

Podemos observar na tabela 12 que houve uma queda de 2% na precisão de todas as categorias (comparando o *corpus* não pré-processado e pré-processado), exceto Basquete que se manteve nos mesmos 94% tando na classificação não pré-processado quanto na pré-processada. Em suma houve uma queda de desempenho da classificação não pré-processada para a pré processada que pode ser facilmente identificada na figura 4, onde é possível verificar uma há um aumento de 1,5% em relação a acurácia de 89% no *corpus* pré-processado para a acurácia de 90,5% para o *corpus* não pré-processado.

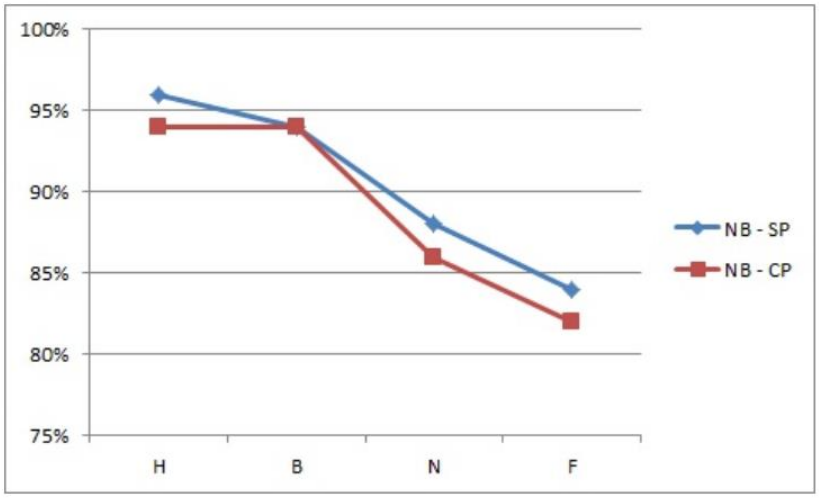
Outro dado importante a ser ressaltado é o que compreende a performance do algoritmo CNB na realização da classificação dos documentos. Com tempo de execução de 0.11 segundos para o *corpus* não pré-processado e de 0.02 segundos para o *corpus* pré-processado. Sendo assim o pré-processamento proporcionou um ganho significativo na performance reduzindo em 81,8% o tempo de execução do CNB.

Tabela 12 – Precisão por classe com CNB pré-processado

Detalhamento da precisão por classe do CNB pré-processado						
	RVP	RFP	Precisão	Revocação	Medida F	Área ROC
Beisebol	0,94	0,04	0,887	0,94	0,913	0,986
Basquete	0,94	0,067	0,825	0,94	0,879	0,977
Notícia	0,86	0,027	0,915	0,86	0,887	0,98
Futebol	0,82	0,013	0,953	0,82	0,882	0,96
Média	0,88	0,037	0,895	0,89	0,89	0,976

Fonte: Os autores.

Figura 4 – Gráfico de Precisão do CNB



Fonte: Os autores.

3.2 Avaliação com J48

O algoritmo J48 também foi utilizado na realização de experimentos propostos neste trabalho e por meio de seus resultados foi possível estabelecer uma base de comparação com os demais algoritmos utilizados neste trabalho.

Os resultados da sua utilização foram descritos por meio de matriz confusão nas tabelas 13 e 14 que foram empregadas para determinar a acurácia e precisão descritos nas tabelas 15 e 16.

Tabela 13 – Matriz Confusão - CJ48 não pré-processado

CJ48 não pré-processado				
	Beisebol	Basquete	Notícia	Futebol
Beisebol	48	1	1	0
Basquete	1	47	1	1
Notícia	3	2	44	1
Futebol	2	4	2	42

Fonte: Os autores.

Tabela 14 – Matriz Confusão - CJ48 pré-processado

CJ48 não pré-processado				
	Beisebol	Basquete	Notícia	Futebol
Beisebol	49	0	1	0
Basquete	0	48	1	1
Notícia	1	1	44	4
Futebol	2	2	2	44

Fonte: Os autores.

A partir destes resultados descritos nas tabelas 15 e 16 foi possível determinar que CJ48 teve acurácia de 89% no *corpus* não pré-processado e de 92,5% no *corpus* pré-processado, obtendo assim um ganho de 3,5% no desempenho em relação ao *corpus* não pré-processado.

Quando analisamos a precisão por categoria, descritos nas tabelas 15 e 16, observamos que o CJ48 obteve, na categoria futebol, um ganho de precisão de 8%, enquanto as demais categorias obtiveram ganho de apenas 2% exceto na categoria notícia, que assim como aconteceu com a categoria basquete no CNB, se manteve estável com precisão de 88%.

Na figura 5 é possível obter uma visualização da precisão por categoria, nela é evidenciado o ganho de precisão obtido na classificação do *corpus* pré-processado.

A performance do algoritmo CJ48 na realização da classificação dos documentos também pode ser verificada e como resultado obteve-se um tempo de execução de 1.1 segundos para o

corpus não pré-processado e de 0.41 segundos para o *corpus* pré-processado. Com uma redução de 62,7% no tempo de execução o algoritmo CJ48 teve uma leve desvantagem na performance em relação ao algoritmo CNB.

Tabela 15 – Precisão por classe com CJ48 não pré-processado

Detalhamento da precisão por classe do CJ48 não pré-processado						
	RVP	RFP	Precisão	Revocação	Medida F	Área ROC
Beisebol	0,96	0,047	0,873	0,96	0,914	0,974
Basquete	0,92	0,02	0,939	0,92	0,929	0,966
Notícia	0,88	0,047	0,863	0,88	0,871	0,935
Futebol	0,8	0,033	0,889	0,8	0,842	0,901
Média	0,89	0,037	0,891	0,89	0,889	0,944

Fonte: Os autores.

Tabela 16 – Precisão por classe com CJ48 pré-processado

Detalhamento da precisão por classe do CJ48 não pré-processado						
	RVP	RFP	Precisão	Revocação	Medida F	Área ROC
Beisebol	0,98	0,02	0,942	0,98	0,961	0,983
Basquete	0,96	0,02	0,941	0,96	0,95	0,963
Notícia	0,88	0,027	0,917	0,88	0,898	0,935
Futebol	0,88	0,033	0,898	0,88	0,889	0,901
Média	0,925	0,025	0,925	0,925	0,925	0,946

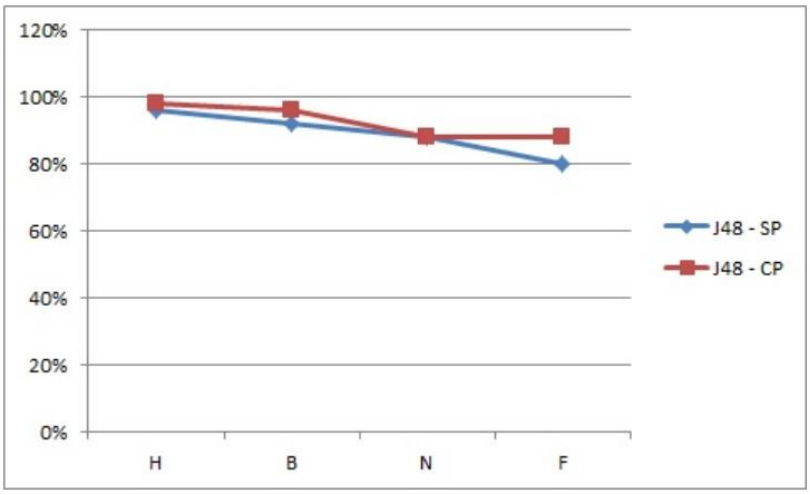
Fonte: Os autores.

3.3 Avaliação com CSMO

O último algoritmo classificador testado foi o SMO que obteve como resultado as matrizes confusão retratadas nas tabelas 17 e 18 que deram origem as tabelas 19 e 20, utilizadas para determinar a acurácia e a precisão da classificação nos dois *corpus*.

Ao analisar os resultados expostos nas tabelas 19 e 20 podemos observar que o CSMO obteve acurácia de 83% no *corpus* não pré processado e 86,5% no *corpus* pré-processado, ou seja o pré processamento proporcionou um aumento de 3,5% no desempenho do CSMO em relação ao *corpus* não pré-processado.

Figura 5 – Gráfico de Precisão do CJ48



Fonte: Os autores.

Tabela 17 - Matriz Confusão - CSMO não pré-processado

CSMO não pré-processado				
	Beisebol	Basquete	Notícia	Futebol
Beisebol	35	6	1	8
Basquete	3	41	1	5
Notícia	1	0	48	1
Futebol	1	4	3	42

Fonte: Os autores.

Com a análise também foi possível observar, que em relação aos algoritmos analisados anteriormente o CSMO foi o que obteve os piores resultados na classificação da categoria beisebol com taxas de precisão abaixo dos 80% contrastando categoria notícia com precisão de 96% em conformidade com os demais algoritmos.

Além de obter o pior desempenho na realização da classificação dos documentos o CSMO também obteve a pior performance com um tempo de execução de 1.68 segundos para o *corpus* não pré-processado e de 0.69 segundos para o *corpus* pré-processado. Com uma redução de 58,9% no tempo de execução o CSMO foi o algoritmo que obteve menor ganho em relação ao *corpus* não pré-processado.

Tabela 18 - Matriz Confusão - CSMO pré-processado

CSMO pré-processado				
	Beisebol	Basquete	Notícia	Futebol
Beisebol	38	5	0	7
Basquete	3	44	1	2
Notícia	1	1	48	0
Futebol	0	3	4	43

Fonte: Os autores.

Tabela 19 - Precisão por classe com CSMO não pré-processado

Detalhamento da precisão por classe do CSMO não pré-processado						
	RVP	RFP	Precisão	Revocação	Medida F	Área ROC
Beisebol	0,7	0,033	0,875	0,7	0,778	0,868
Basquete	0,82	0,067	0,804	0,82	0,812	0,916
Notícia	0,96	0,033	0,906	0,96	0,932	0,98
Futebol	0,84	0,093	0,75	0,84	0,792	0,88
Média	0,83	0,057	0,834	0,83	0,829	0,911

Fonte: Os autores.

Na figura 6 podemos observar melhor o a precisão obtida na classificação dos dois *corpus*. Nota-se que com pré-processamento obteve-se um leve ganho de desempenho em relação ao *corpus* pré-processado, exceto na categoria notícias que precisão idêntica em ambos os *corpus*.

3.4 Comparação dos classificadores

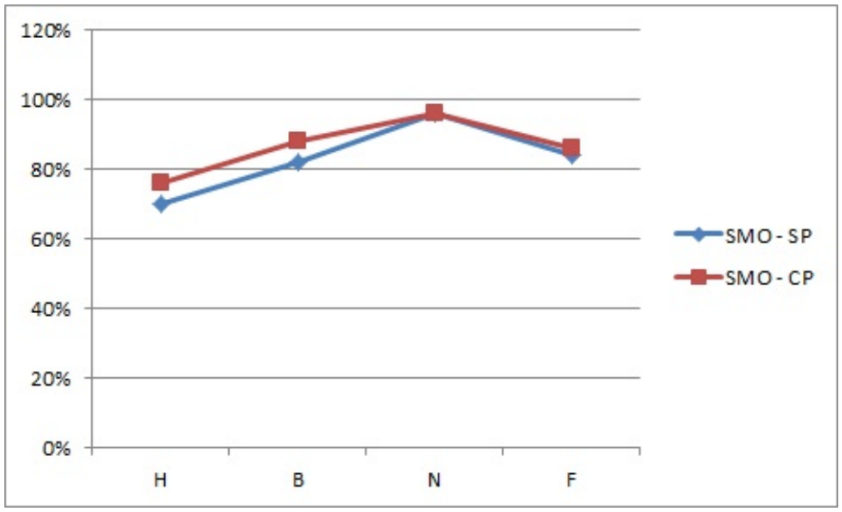
Para efetuar a comparação dos algoritmos classificadores NB, J48 e SMO foram utilizadas dentre as métricas citadas na seção 1.10 a acurácia que corresponde ao acerto do sistema considerando a proporção de instâncias corretamente classificadas no total dos registros, a precisão que é a proporção de casos positivos que foram corretamente identificados pelo classificador e o tempo de processamento que corresponde ao tempo gasto pelos algoritmos para realização da classificação dos experimentos.

Tabela 20 - Precisão por classe com CSMO pré-processado

Detalhamento da precisão por classe do CSMO pré-processado						
	RVP	RFP	Precisão	Revocação	Medida F	Área ROC
Beisebol	0,76	0,27	0,905	0,76	0,826	0,908
Basquete	0,88	0,06	0,83	0,88	0,854	0,927
Notícia	0,96	0,033	0,906	0,96	0,932	0,979
Futebol	0,86	0,06	0,827	0,86	0,843	0,91
Média	0,865	0,045	0,867	0,865	0,864	0,931

Fonte: Os autores.

Figura 6 – Gráfico de Precisão do CSMO



Fonte: Os autores.

A acurácia foi o primeiro item a ser analisado, os resultados obtidos foram descritos na tabela 21.

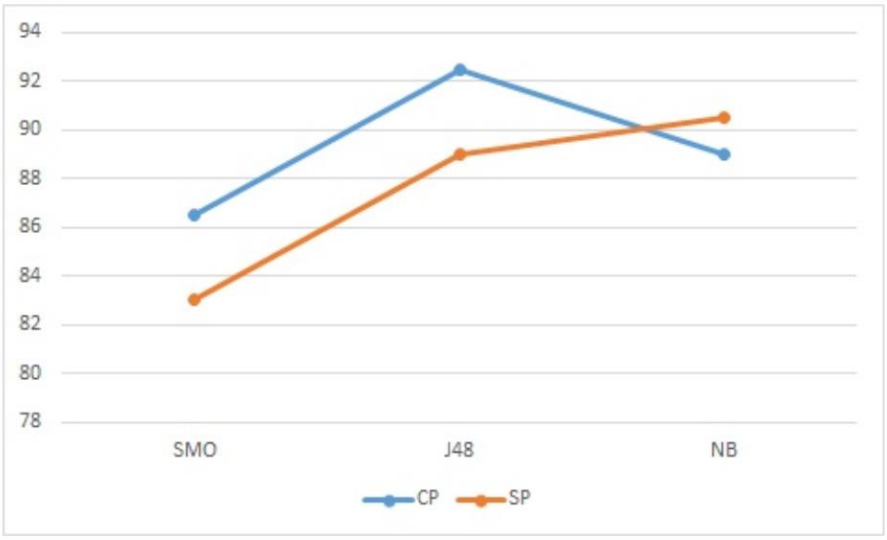
A partir das informações contidas na tabela 21 é possível determinar que todos os algoritmos testados obtiveram um resultado satisfatório, acima de 80% em ambos os experimentos. No entanto dois algoritmos se destacaram, sendo eles, o NB obtendo melhor acurácia na classificação do *corpus* não pré processado com uma taxa de 90,5% de acerto, e o J48 que obteve melhor acurácia no *corpus* pré-processado com uma taxa de 92,5% de acerto. Na figura 7 é possível obter uma melhor visualização da comparação da acurácia dos algoritmos.

Tabela 21 – Comparação da acurácia dos algoritmos

Comparação da acurácia dos algoritmos			
Algoritmo	Não Pré-Processado	Pré-Processado	Média
CNB	83%	86,5 %	84,75 %
CJ48	89%	92,5 %	90,75 %
CSMO	90,5 %	89%	89,75 %
Média	87,5 %	89,3 %	88,42 %

Fonte: Os autores

Figura 7 – Gráfico de acurácia dos classificadores



Fonte: Os autores.

Destaca-se também que com exceção ao algoritmo NB que teve uma queda de acurácia de 1,5%, todos os outros obtiveram um ganho de acurácia de 3,5% em comparação do *corpus* não pré-processado para o pré-processado o que deixa evidente a importância do pré-processamento no ganho de acurácia e por consequência também no desempenho como constatado na figura 9.

Outra variável importante possível de analisar com os resultados dos experimentos, foi a taxa de precisão que nada mais é que o índice de acerto na classificação de documentos em suas devidas categorias. A tabela 22 contém uma comparação da precisão por categoria dos resultados obtidos na classificação de ambos os *corpus* de documentos contemplando os três algoritmos propostos.

Tabela 22 – Comparação da precisão dos classificadores

Comparação da precisão dos classificadores						
	NBSP	NBCP	J48SP	J48CP	SMOSP	SMOCP
Beisebol	96%	94%	96%	98%	70%	76%
Basquete	94%	94%	92%	96%	82%	88%
Notícia	88%	86%	88%	88%	96%	96%
Futebol	84%	82%	80%	88%	84%	86%
Média	90%	88%	89%	92%	83%	86%

Fonte: Os autores.

Analisando a tabela 22 podemos observar que o algoritmo J48 obteve melhor desempenho na classificação dos Esportes; Beisebol e Basquete, com uma média de precisão atingindo 95% de acerto nos dois *corpus* analisados, 2% acima da taxa obtida com NB, enquanto o SMO se destacou positivamente na classificação das notícias com precisão de 96% em ambos os *corpus*,

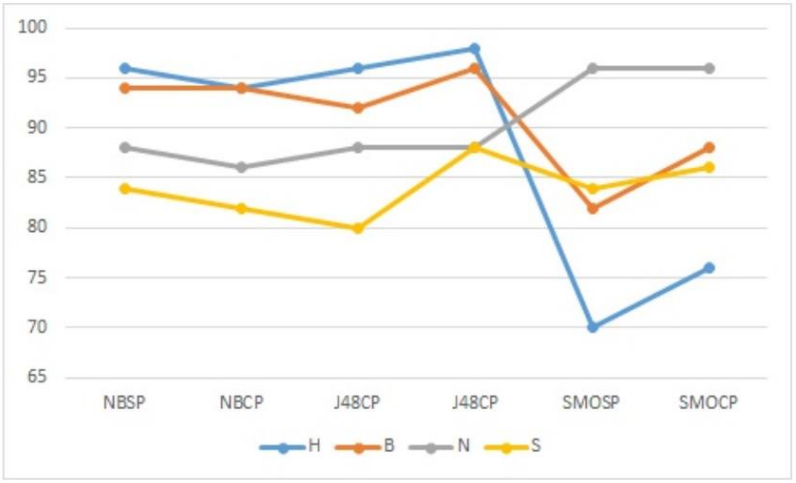
e negativamente obtendo a menor precisão dentre todas as categorias analisadas com apenas 70% de precisão no *corpus* não pré-processado.

A figura 8 nos fornece uma melhor visualização da comparação dos classificadores. Nela podemos observar que com exceção ao NB, todos os outros obtiveram um ganho de desempenho do *corpus* não pré-processado em comparação ao pré-processado.

Como resultado da execução dos algoritmos classificadores obteve-se também o tempo de execução que cada um levou para realizar a classificação. O tempo gasto por cada algoritmo está descrito na tabela 23.

Com esta informação foi possível avaliar qual destes algoritmos obteve melhor performance, uma fez que todos contemplaram o mesmo hardware durante a execução de seus processos. Sendo assim, analisando o gráfico de performance contido na figura 9, podemos concluir que o algoritmo NB apresentou melhor performance durante a realização dos experimentos em ambos os *corpus*, no entanto não apresentou o mesmo em relação aos demais algoritmos testados, como comprovado anteriormente.

Figura 8 – Gráfico da comparação precisão dos classificadores



Fonte: Os autores.

Tabela 23 – Tempo gasto na classificação dos experimentos

Tempo de Processamento(segundos)			
Algoritmo	Não Pré-Processado	Pré-Processado	Média
CNB	0,11	0,02	0,07
CJ48	1,1	0,41	0,76
CSMO	1,68	0,69	1,19
Média	0,96	0,37	0,67

Fonte: Os alunos.

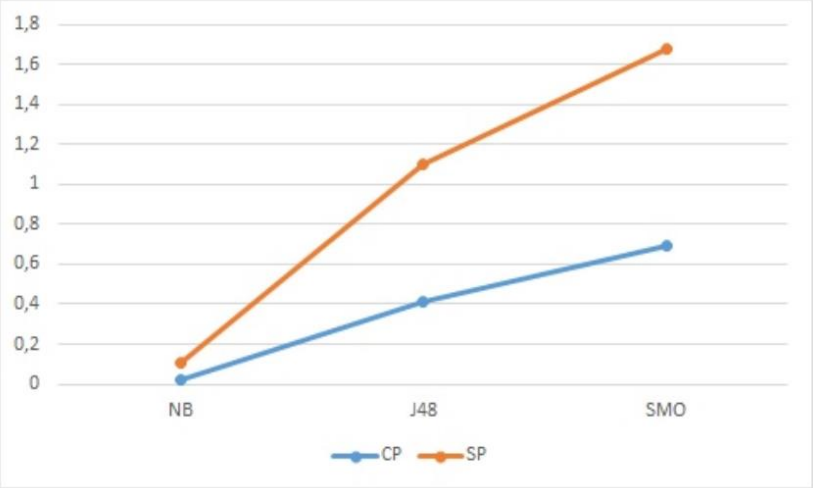
Na figura 9 é possível observar também que todos os algoritmos apresentaram melhor performance quando aplicados no *corpus* pré-processado. Com redução de 81,1%, o CNB foi

o algoritmo que obteve maior redução no tempo de processamento, enquanto o CJ48 obteve redução de 62,7% e o CSMO 58,9% de redução no tempo de processamento.

Ao comparar a classificação realizada no *corpus* pré-processado entre algoritmos, observamos que o CNB obteve um ganho de performance de 97,1% em relação ao CSMO e 95,12% em relação ao CJ48 que por sua vez obteve ganho de 40,58% em relação ao CSMO.

Na comparação com o *corpus* não pré-processado o CNB obteve ganho de performance de 98,81% em relação ao CSMO e 98,1% em relação ao CJ48 que por sua vez obteve ganho de 75,6% em relação ao classificador SMO não pré-processado.

Figura 9 - Gráfico da performance dos classificadores



Fonte: Os autores.

Quando comparamos o experimento não pré-processado observamos que o CNB obteve ganho de 93% em relação ao algoritmo CSMO e 90% em relação ao CJ48 que por sua vez obteve ganho de 34,52% em relação ao CSMO.

4 Conclusão

Concluímos com base nos resultados descritos na seção 3 que o melhor algoritmo classificador, dentre os analisados, para dados não pré-processados é o Naive Bayes, enquanto o algoritmo J48 foi eleito o melhor na classificação de dados pré-processados.

Quanto a performance, em relação ao tempo de processamento (execução), foi possível concluir que Naive Bayes é o melhor algoritmo classificador tanto em dados pré-processados quanto não processados.

Em relação ao pré-processamento foi possível concluir que este contribui para o ganho de performance e desempenho dos algoritmos, melhorando a taxa de acurácia e reduzindo o tempo de processamento.

Mediante as etapas de *stemming* e remoção de stop words, contidas no pré-processamento, ambas são as que mais influenciam diretamente no resultado final do experimento. Portanto a escolha do algoritmo *stemming* e da lista de *Stopwords* deve ser feita com atenção, uma vez que estes são responsáveis pela otimização da coleção de documentos.

Concluímos também que a classificação de documentos é uma excelente "ferramenta" na organização de grandes quantidades de dados.

Para trabalhos futuros sugerimos a realização de outros estudos comparativos contemplando outros algoritmos de *stemming* (comparando os algoritmos de *Stemming* para língua portuguesa e inglesa), novas listas de *Stopwords*, novos algoritmos classificadores e base de dados em diversos idiomas.

Referências

ANACLETO, A. C. da S. **Aplicação de técnicas de data mining em extração de elementos de documentos comerciais**. 107 p. Dissertação (Análise de Dados e Sistemas de Apoio à Decisão) - Universidade do Porto, Porto - PT, 2009.

BAEZA-YATES, R. et al. **Recuperação de Informação:** Conceitos e Tecnologia das Máquinas de Busca. 2. ed. [S.l.]: Porto Alegre: Bookman, 2012. 614 p.

BERNARDES, J. A. B. **Algoritmo de aprendizado de máquina e representação de incerteza em sistemas baseados em conhecimento sob a ótica de funções de pertinência aproximada**. 108 p. Monografia (Monografia) — Universidade Federal de Lavras, Lavras - MG, 2010.

BRILHADORI, M. et al. **Estudo comparativo entre algoritmos de árvores de classificação e máquinas de vetores suporte, baseados em ensembles de classificadores**. Universidade de São Paulo, p. 109, 2013.

CAMARGO, S. da S. **Um modelo neural de aprimoramento progressivo para redução de dimensionalidade**. 107 p. Tese (Doutorado em Ciência da Computação) — Universidade Federal do Rio Grande do Sul, Porto Alegre -RS, 2010. Citado na página 21.

CARVALHO, A. de et al. **Inteligência artificial: uma abordagem de aprendizado de máquina**. 1. ed. [S.l.]: São Paulo: Grupo Gen LTC, 2011. 394 p. Citado 2 vezes nas páginas 18 e 19.

CONCEIÇÃO, A. W. **Um sistema voltado ao armazenamento e recuperação de conteúdo textual de dierentes contextos**. 61 p. Monografia (Monografia) - Universidade Federal de Santa Catarina, Araranguá - SC, 2013.

CONDUTA, B. C.; MAGRIN, D. H. **Aprendizagem de Máquina**. 19 p. Tese (Doutorado) - Universidade Estadual de Campinas, Campinas - SP, 2010.

EBECKEN, N. F. et al. **Mineração de textos. Sistemas inteligentes:** fundamentos e aplicações. São Carlos: Manole, p. 337–370, 2003.

GEVERT, V. G. et al. Análise de crédito bancário utilizando o algoritmo sequential minimal optimisation. **XLI Simpósio Brasileiro de Pesquisa Operacional**, p. 2242–2253, 2009.

GOMES, R. M. **Desambiguação de Sentido de Palavras Dirigida por Técnicas de Agrupamento sob o Enfoque da Mineração de Textos**. 119 p. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Pontifica Católica, Rio de Janeiro - RJ, 2009.

HAN, J.; KAMBER, M.; PEI, J. **Data mining, southeast asia edition**: Concepts and techniques. 2. ed. [S.l.]: Morgan kaufmann, 2006.

JOHN, G. H.; LANGLEY, P. Estimating continuous distributions in Bayesian classifiers. In: MORGAN KAUFMANN PUBLISHERS INC. **Proceedings of the Eleventh conference on Uncertainty in artificial intelligence**. [S.l.], 1995. p. 338–345.

JUNIOR, J. R. C. **Desenvolvimento de uma Metodologia para Mineração de Textos**. 96 p. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Pontifica Católica, Rio de Janeiro - RJ, 2007.

LOPES, M. C. S. **Mineração de Dados Textuais Utilizando Técnicas de Clustering para o Idioma Português**. 180 p. Tese (Doutorado em Engenharia Civil) - Universidade Federal do Rio de Janeiro, Rio de Janeiro - RJ, 2004.

MARTINS, S. G. **O processo de indexação e sua relação com a linguística: uma revisão literária**. 41 p. Monografia (Monografia) — Universidade Federal do Rio Grande do Norte, Natal - RN, 2009.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. **Foundations of machine learning**. [S.l.]: MIT press, 2012.

OGURI, P. **Aprendizado de máquina para o problema de Sentiment Classification**. 54 p. Dissertação (Mestrado em Informática) - Universidade Pontifica Católica, Rio de Janeiro - RJ, 2006.

OLSON, D. L.; DELEN, D. **Advanced data mining techniques**. [S.l.]: Springer Science & Business Media, 2008.

PASSARIN, D. **Text Mining no Aperfeiçoamento de Consultas e Definição de Contextos de uma Central de Notícias Baseada em RSS**. 60 p. Monografia (Monografia) - Centro Universitário Luterano de Palmas, Palmas - TO, 2005.

PASSINI, M. L. C. **Mineração de Textos para Organização de Documentos em Centrais de Atendimento**. 105 p. Dissertação (Mestrado em Engenharia Civil) - Universidade Federal do Rio de Janeiro, Rio de Janeiro - RJ, 2012.

RODRIGUES, J. P. **Sistemas inteligentes híbridos pra classificação de texto**. 110 p. Dissertação (Mestrado) - Universidade Federal de Pernambuco, Recife - PE, 2009.

RONCERO valeriana G. **Classificação semi-supervisionada de textos em ambientes distribuídos**. 107 p. Tese (Doutorado em Engenharia Civil) - Universidade Federal do Rio de Janeiro, Rio de Janeiro - RJ, 2010.

SCHIESSL, J. M. **Descoberta de Conhecimento em Texto aplicada a um sistema de atendimento ao consumidor**. 106 p. Dissertação (Mestrado em Ciência da Informação) - Universidade de Brasília, Brasília - DF, 2007.

SILVA, C. F. da. **Curso de Informações Linguísticas na etapa de pré-processamento em Mineração de Textos**. 109 p. Dissertação (Mestrado em Computação Aplicada) - Universidade do Vale do Rio dos Sinos, São Leopoldo- RS, 2004.

SILVA, R. M. **Redes neurais artificiais aplicadas à detecção de intrusão em redes TCP/IP**. 144 p. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Pontifica Católica, Rio de Janeiro - RJ, 2005.

SOARES, F. A. **Mineração de textos na Coleta Inteligente de Dados na Web**. 120 p. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Pontifica Católica, Rio de Janeiro - RJ, 2008.

SOUZA, J. L. de. **Aplicando Técnicas de Aprendizado de Máquina em Planejamento**. 101 p. Dissertação (Mestrado em Ciência da computação) - Universidade Federal de Uberlândia, Uberlândia - MG, 2014.

TAN, A.-H. et al. Text mining: The state of the art and the challenges. In: **Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases**. [S.l.: s.n.], 1999. v. 8, p. 65–70.

VIERA, A. F. G.; VIRGIL, J. Uma revisão dos algoritmos de radicalização em língua portuguesa. **Information Research**, v. 12, n. 3, p. 8, 2006.