UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ

***CAMPUS* LUIZ MENEGHEL**

ELITON KATSUHIRO NOUCHI

RACIOCÍNIO BASEADO EM CASOS APLICADO À CORREÇÃO DE QUESTÕES SUBJETIVAS

Bandeirantes

2013

ELITON KATSUHIRO NOUCHI

RACIOCÍNIO BASEADO EM CASOS APLICADO À CORREÇÃO DE QUESTÕES SUBJETIVAS

Monografia apresentada à Universidade Estadual do Norte do Paraná – *campus* Luiz Meneghel – como requisito parcial para aprovação na disciplina Projeto Final II do Curso de Sistemas de Informação.

Orientador: Prof. Me Glauco Carlos Silva

Bandeirantes

2013

ELITON KATSUHIRO NOUCHI

RACIOCÍNIO BASEADO EM CASOS APLICADO À CORREÇÃO DE QUESTÕES SUBJETIVAS

Monografia apresentada à Universidade Estadual do Norte do Paraná – *campus* Luiz Meneghel – como requisito parcial para aprovação na disciplina Projeto Final II do Curso de Sistemas de Informação.

COMISSÃO EXAMINADORA

_____
Prof. Me. Glauco Carlos Silva
UENP – *Campus* Luiz Meneghel

_____
Prof. Dr. André Luis Menolli
UENP – *Campus* Luiz Meneghel

_____
Prof. Me.Bruno Miguel Nogueira de Souza
UENP – *Campus* Luiz Meneghel

Bandeirantes, 18 de novembro de 2013

AGRADECIMENTOS

Primeiramente, quero agradecer a Deus, por ter me dado forças para desenvolver meu trabalho.

Em segundo, quero agradecer minha família que me deram força, coragem e apoio nos momentos de dificuldades, iluminando meus pensamentos, me levando a buscar mais conhecimentos.

Agradeço também a todos os professores que me acompanharam durante a graduação, em especial a Professora Me. Roberta Ekuni de Souza e ao meu Orientador Me. Glauco Carlos Silva.

A todos que de alguma forma me apoiaram e contribuíram para meu crescimento como pessoa.

RESUMO

Este trabalho apresenta a introdução à alguns conceitos sobre recuperação de casos, reutilização de conhecimento, revisão e aprendizado, em Sistemas de Raciocínio Baseado em Casos (RBC), mostra também, tarefas de Mineração de Texto para o pré-processamento, extração de conhecimento e pós-processamento de textos. Utilizando esses conceitos foi desenvolvido um sistema RBC que corrige questões subjetivas com menos interação humana. Para poder realizar a correção das respostas o sistema “limpa” as respostas a serem corrigidas utilizando de técnicas de tokenização, remoção de stopwords e stemming. Feito isso, o sistema recupera as respostas mais similares à nova resposta aplicando o calculo das Similaridades Locais e Globais. Obtendo as respostas mais similares, elas são validadas aplicando-as na técnica de Validação cruzada.

Palavras-chaves: Mineração de textos, recuperação de informação, pré-processamento, extração de conhecimento, pós-processamento, classificação, raciocínio baseado em casos.

ABSTRACT

This paper presents an introduction to some concepts of recovery cases, knowledge reuse, revision and learning in Systems Based Reasoning (CBR), it also shows tasks of text mining for pre-processing, knowledge extraction and post-processing texts. Using these concepts a RBC system that corrects subjective questions with less human interaction was developed. To perform the correction of the answers, the system “clean” the answers to be corrected using techniques of tokenization, removing stopwords and stemming. Then, the system retrieves the most similar answers to the new answer applying the calculation of Local and Global Similarities. Getting the most similar answers, they are validated applying them in the cross-validation technique.

Keywords: Text Mining, Information Retrieval, pre-processing, knowledge extraction, post-processing, classification, case-base reasoning.

LISTA DE FIGURAS

SUMÁRIO

1 Introdução

1.1 Contextualização

Com o aumento da utilização de sistemas na rotina diária das pessoas sendo no trabalho, na escola ou em casa, houve também um grande crescimento da agilidade com que se geram dados e documentos textuais digitalizados. No entanto, a capacidade do próprio ser humano em processar as informações desses documentos se manteve no mesmo ritmo. Uma das áreas que teve grande crescimento na utilização de softwares foi a educação, que teve investimentos em softwares educacionais nas escolas e na criação de cursos online. Porém, a maioria desses casos utiliza somente questões objetivas nas suas avaliações (RODRIGUES, 2006). A preferência na utilização de questões objetivas se da pela razão que as respostas têm formato estruturado e a correção delas é mais automatizada do que em questões subjetivas, que as respostas são sem estrutura, ou seja, textos (SOMMER; SOMMER, 2002).

A nota dada às respostas das questões objetivas pode somente assumir dois valores, certo ou errado, que correspondem a 100% ou 0% de acerto da questão. Já nas questões subjetivas o valor das notas pode variar entre valores na faixa de 0% a 100% do valor da questão, dependendo também do critério de correção adotado pelo especialista.

As questões subjetivas, também chamadas de questões discursivas, têm suas respostas em formato textual, portanto é possível a utilização da mineração de textos para extrair algum conhecimento. Nahm e Mooney (2002) definem mineração de textos como o processo de encontrar padrões, modelos, direções, tendências e regras, úteis ou interessantes de documentos textuais. A mineração de textos é mais fácil de ser aplicada em questões que são respondidas virtualmente, mas podem ser utilizadas em respostas manuscritas contanto que sejam transformadas em um documento digital.

De acordo com Fernandes (2003), Raciocínio Baseado em Casos é uma técnica que busca a solução de um novo caso por meio da recuperação e adaptação de soluções passadas similares, dentro de um mesmo domínio do problema. Para auxiliar o sistema RBC no processamento de documentos textuais pode-se aplicar a mineração de textos.

1.2 Formulação e Escopo do Problema

Na avaliação do aprendizado dos alunos o método mais utilizado ainda é a prova, dentro dessas provas podem ter vários tipos de exercícios para medir o conhecimento dos alunos e cada tipo de questão avalia melhor alguns aspectos da aprendizagem do aluno (NÜNEZ, 2013).

Comparando a correção de questões subjetivas com a de questões objetivas, as objetivas os alunos preenchem um gabarito e depois uma pessoa sem conhecimento nenhum sobre o assunto das questões, consegue corrigi-las somente tendo as alternativas corretas em mãos e rapidamente corrige diversas questões. Ainda pode se utilizar um leitor de gabarito que identificam quais alternativas foram marcadas e com auxilio de algum sistema que tenha as respostas corretas faz a comparação e já é obtida a nota da avaliação.

A necessidade de um especialista e da grande demora em analisar as inúmeras respostas de questões subjetivas são, em vários casos, os fatores que fazem optarem por utilizar questões objetivas nas avaliações (SOMMER; SOMMER, 2002). Para tentar tornar a correção das questões subjetivas mais fácil e diminuir a demora na correção, o sistema proposto utilizará uma base de casos, ou seja, uma base com respostas já corrigidas por um especialista, e por meio dessa base irá encontrar os mais similares que serão sugeridos ao usuário como correção para da nova resposta. Depois de corrigida a nova resposta será armazenada na base de casos para ajudar na correção de outros casos.

1.3 Justificativas

Uma das desvantagens da utilização de questões subjetivas citada por Mattar (1994) é que elas são mais onerosas e mais demoradas para serem analisadas que outros tipos de questões.

Nünez (2013) apresenta algumas vantagens das questões subjetivas em relação às objetivas, como a melhor avaliação da capacidade de analisar, organizar, sintetizar o conhecimento, aplicá-lo e avaliá-lo. Para aproveitar as vantagens e não deixar de usufrui-las somente por serem mais demoradas e caras na correção, foi proposto o desenvolvimento do sistema RBC.

1.4 Objetivos

Nas seguintes subseções serão apresentados os objetivos geral e específico que esperam ser atingidos no término do trabalho.

1.4.1 Objetivo Geral

O objetivo do trabalho é facilitar a correção de questões subjetivas implementando um Sistema de Raciocínio Baseado em Casos (RBC) que dado um caso novo, utiliza a mineração de textos para recuperar casos similares de uma base de respostas previamente corrigidas por um especialista (casos) e mostra sugestões de correção do novo caso para o usuário escolher a que ele considere mais similar, depois de encontrado o que mais se enquadra o sistema armazena o caso novo corrigido na base para melhorar o seu desempenho em casos futuros.

1.4.2 Objetivos Específicos

- Seleção da base de respostas a uma questão subjetiva com sua correção;
- Realizar o pré-processamento na base de respostas;
- Calcular a Similaridade Local e Similaridade Global para recuperar os casos similares;
- Armazenar a nova resposta corrigida;
- Aplicar um método de validação para avaliar os resultados obtidos;
- Avaliar junto ao especialista os resultados obtidos;

1.5 Organização do Trabalho

O trabalho será organizado da seguinte forma. Na seção 2 é apresentada a fundamentação teórica necessária para o trabalho. Na seção 3 é exibido o desenvolvimento da solução proposta. Na seção 4 está a conclusão que foi possível obter com o desenvolvimento do projeto e as propostas para trabalhos futuros que podem melhorar o projeto.

2 Fundamentação teórica

2.1 Raciocínio Baseado em Casos

Raciocínio baseado em casos (RBC) é uma técnica de inteligência artificial baseada no comportamento humano para resolver problemas.Um sistema RBC resolve um problema novo a partir de soluções de problemas similares em que as soluções foram dadas por um especialista ou foram bem sucedidas.

Von Wangenheim e Von Wangenheim (2003) define RBC como um enfoque para solução de problemas e aprendizado utilizando experiências passadas, podendo ser necessário adaptar as experiências passadas para adequar a solução ao novo caso.

Sistemas RBCs tentam resolver problemas que eram resolvidos melhor por humanos do que maquinas, para isso é necessário que os sistemas tenham conhecimento do que os especialistas têm sobre o problema a ser resolvido. Esse conhecimento deve ser representado na base de casos, então quanto mais conhecimento, ou seja, quanto mais casos houver na base, melhor é o desempenho na resolução dos problemas (REZENDE, 2005).

2.1.1 Ciclo do RBC

O ciclo mostra os processos envolvidos na resolução de problemas utilizando Raciocínio baseado em casos. O modelo mais aceito é apresentado por Aamodt e Plaza (1994) e pode ser descrito em quatro processos:

1. **Recuperar** o caso mais similar ou conjunto de casos mais similares.
2. **Reutilizar** a informação e conhecimento do caso recuperado para resolver o problema.
3. **Revisar** a solução proposta.
4. **Reter** o novo caso solucionado ou parte dele para utilização futura.

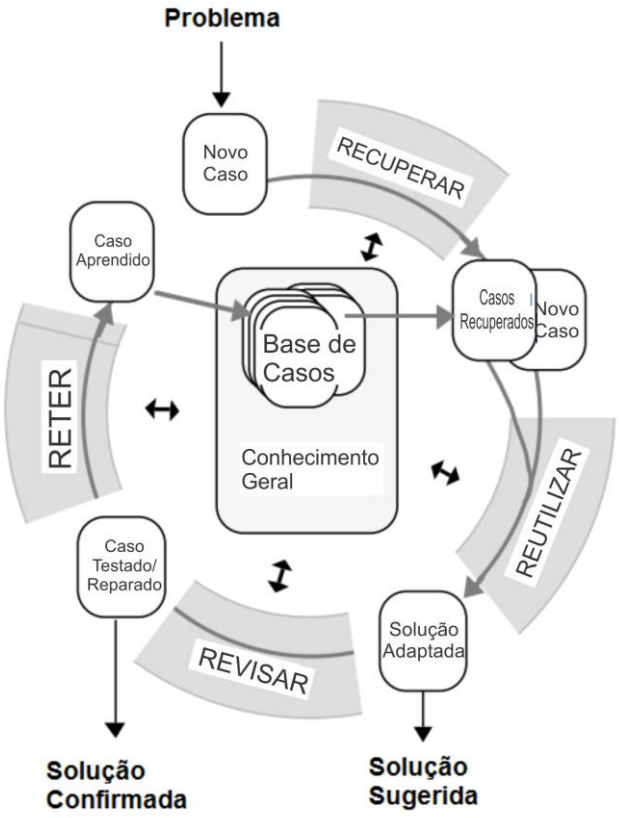


Figura 2.1 - Ciclo Raciocínio Baseado em Casos (AAMODT; PLAZA, 1994).

2.1.1.1 Recuperar

O objetivo do processo de recuperação de casos é encontrar um caso ou um conjunto de casos da base de casos que tenham uma solução útil para o novo problema apresentado (VON WANGENHEIM; VON WANGENHEIM, 2003). O processo de recuperação é responsável por descobrir os casos similares ao novo caso (problema). A busca pelos casos similares é realizada comparando cada caso na base com o problema utilizando de uma medida de similaridade. Von Wangenheim e Von Wangenheim (2003) descrevem medida de similaridade como a maneira que será calculada a similaridade entre o caso novo a um caso da base, aplicando para todos os casos armazenados na base.

2.1.1.2 Reutilizar

De acordo com Von Wangenheim e Von Wangenheim (2003) a reutilização é basicamente a adaptação da solução do caso recuperado para o novo problema,

utilizando de técnicas que tentam resolver desafios como: descobrir os aspectos da situação que devem ser adaptados, como realizar e controlar esse processo. De modo geral, a adaptação envolve a identificação das diferenças entre o caso problema e o recuperado, verificando que características podem ser adaptadas para o caso problema.

Não é estritamente necessário que todos os casos sejam adaptados, somente em situações em que o caso recuperado não satisfaz totalmente os requisitos implícitos no novo problema. A solução do melhor caso recuperado pode ser reutilizado diretamente, copiando a solução para o problema. A reutilização por cópia é frequentemente utilizada em tarefas de classificação, que podem se basear em raciocínios complexos, mas na maioria das vezes a solução é simples e cada classe pode ser representada varias vezes na base e se o caso recuperado for suficientemente similar é provável que seja uma solução adequada (VON WANGENHEIM; VON WANGENHEIM, 2003). Porem, se necessário, a solução deve ser analisada e moldada às necessidades do problema. A adaptação é muito utilizada, por exemplo, em sistemas de decisão medica, onde um sintoma diferente pode levar ao diagnostico de uma doença com tratamento diferente (MÁNTARAS et al., 2005).

2.1.1.3 Revisar

Para que um sistema RBC possa aprender com suas experiências é necessário interpretar o que estava certo ou errado nas soluções. Sem essa validação o sistema poderia repetir as diferenças e os erros cometidos na solução recuperada e não ter um aprendizado correto.

Uma das principais características do RBC é a habilidade de aprender com suas experiências. Sem um feedback o sistema pode resolver mais rápido os problemas, mas reutilizará os erros e não irá melhorar seu conhecimento (KOLODNER, 1992).

2.1.1.4 Reter

A retenção é o processo em que as experiências obtidas ao longo da resolução do problema são armazenadas para uso em problemas similares futuros. Com isso, o sistema RBC é constantemente melhorado. A meta da fase de retenção é tornar o

sistema mais poderoso conforme o passar do tempo e da sua utilização(VON WANGENHEIM; VON WANGENHEIM, 2003).

Segundo Aamodt e Plaza (1994), reter é o processo que incorpora o conhecimento útil do novo caso resolvido para o conhecimento já existente. A aprendizagem pode ser do sucesso ou falha da solução proposta dependendo do resultado da revisão, em casos de falha, os casos são armazenados para que evitar que aquelas soluções sejam reutilizadas em usos futuros.

Von Wangenheim e Von Wangenheim (2003) dividem a maneira como são realizadas a retenção em métodos de força bruta, que armazenam todos os casos solucionados com sucesso na base de casos, ou por métodos inteligentes, que fazem uma filtragem e avaliação do conhecimento dos novos casos.

2.1.2 Recuperação de casos

A recuperação de casos é o centro do processo de um sistema RBC, etapa em que realmente ocorre a tarefa de descobrimento de padrões em documentos textuais.

A recuperação de casos consiste na fase em que são realizadas as tarefas de similaridade para recuperar os casos úteis ao objetivo proposto (MENDES, 2009).

Nesta etapa, são aplicados os algoritmos para a identificação da similaridade global, como o Nearest Neighbor, Naive Bayes, Árvore de decisão entre outros, cada um desses algoritmos tem uma melhor atuação dependendo do tipo de tarefa a ser realizada. A similaridade global é uma medida que determina a similaridade entre o caso de entrada e o caso da base, levando em consideração todos os índices (VON WANGENHEIM; VON WANGENHEIM, 2003).

Para deixar o calculo da similaridade global mais sensitivo é necessário obter o valor da similaridade de todos os índices individualmente, definido como similaridade local. Sem o calculo individual os valores da similaridade de um atributo só podem assumir 0 ou 1, ou seja, 0 se forem diferentes e 1 se forem iguais (VON WANGENHEIM; VON WANGENHEIM, 2003).

Os métodos de análise são separados em diferentes grupos como: classificação, regressão, clustering e análise de associações.

A classificação busca características que sejam semelhantes com as de documentos que já foram classificados.

A classificação na mineração de textos é uma tarefa que identifica os tópicos principais de um documento e sua associação baseando-se em um algoritmo pré-definido, construído a partir de um conjunto de treinamento definido por pessoas experientes no assunto (BARION; LAGO, 2008).

2.2 Mineração de textos

De acordo com Feldman (1997), mineração de textos é uma extração não trivial de informações, não explícitas, de grandes bases textuais, previamente desconhecidas e potencialmente úteis. Ela utiliza várias técnicas de mineração de dados, porém se torna um processo mais complexo por tratar com dados não estruturados.

2.2.1 Pré-processamento

O processamento de documentos textuais envolve vários desafios. Um desafio é o formato do texto que é sem nenhuma estrutura ou pouca estruturação, sendo, o fator que mais o difere da mineração de dados. O tamanho dos documentos é outro desafio, eles facilmente têm milhares de termos ou palavras, e muitas dessas palavras tem o mesmo significado ou não são significantes para o resultado da recuperação. O tratamento desses e outros desafios devem ser realizados na etapa de Pré-processamento. Esta etapa abrange a tarefa de seleção das bases de textos e o tratamento dos textos para filtrar e selecionar o conteúdo que melhor represente o texto, eliminando toda a informação contida nos documentos que não seja de interesse ao objetivo da mineração.

Feldman e Sanger (2007) determinam tarefas dessa etapa como sendo, todas as rotinas, processos e métodos necessários para preparar os dados para a etapa de recuperação de casos.

2.2.2 Recuperação de informação

Uma das técnicas que é base para a mineração de textos foi a recuperação de informação (RI). A recuperação de informação realiza a indexação e busca documentos por meio de frases ou palavras pré-definidos. Surgiu com objetivo de organizar e representar informação de forma que torne fácil e objetivo o seu acesso.

Um sistema de recuperação de informação deve analisar um conjunto de documentos e ordena-los de acordo com a importância para o usuário.

De acordo com Salton e Mcgill (1983), a recuperação de informação utiliza de técnicas que aumentam a velocidade com que se armazena e acessa os dados. Estas técnicas incluem a atribuição de identificadores que representem o conteúdo dos documentos.

2.2.3 Extração de conhecimento

Para Feldman e Sanger (2007) a extração de conhecimento, consiste em vários mecanismos para descobrimento de padrões dentro de uma determinada coleção de documentos ou um subconjunto de uma coleção de documentos. Como objetivo, a extração de conhecimento tenta encontrar termos em um documento textual para estrutura-los em um formato em que as maquinas possam processar (HACK et al., 2013).

A extração de conhecimento é o centro do processo de mineração de textos, etapa que realmente ocorre a tarefa de descobrimento de padrões em documentos textuais. Nesta etapa são aplicados os algoritmos para a identificação de padrões, como o Nearest Neighbor, Naive Bayes, Arvore de decisão, entre outros (FELDMAN; SANGER, 2007).

2.2.3.1 Naive Bayes

O Naive Bayes (naive = ingênuo, do inglês) é um algoritmo que calcula a probabilidade de que um caso novo pertença a cada uma das classes. Ele considera que o valor de um termo de uma determinada classe é independente (YANG; LIU, 1999).

Mccallum e Nigam (1998) descrevem o algoritmo como o mais simples dos classificadores, pelo fato do algoritmo assumir que todos os atributos dos casos são independentes dos outros do mesmo contexto da classe.

Isso o torna menos eficiente do que aproximações exponencialmente complexas não baseadas em Naive Bayes porque ele não usa a combinação de palavras para predição (YANG; LIU, 1999).

Para uma melhor compreensão do algoritmo Naive Bayes, primeiro é necessário entender o Teorema de Bayes. Este teorema se baseia na probabilidade conjunta de dois atributos ocorrerem, considerando os dois atributos como sendo *X* e *Y*, podemos assumir como *P(X)* sendo a probabilidade do atributo *X* ocorrer, *P(Y)* como a probabilidade o atributo Y ocorrer e *P(X,Y)* representa a probabilidade de X e *Y* ocorrerem.

P(X,Y) é o resultado da multiplicação entre *P(X)* com *P(Y)*, isso se os atributos forem independentes, em que a presença de um não afete a probabilidade do outro ocorrer. Agora se *X* e *Y* forem dependentes um do outro, é necessário que a probabilidade condicional de *Y* dado *X* seja calculada com a seguinte equação:

$$\boldsymbol{P(Y|X) = P(X,Y)/P(X)} \quad \textbf{(2.1)}$$

A probabilidade condicional $P(Y|X)$ representa a probabilidade de Y ocorrer dado que X já tenha ocorrido. Sendo que $P(X,Y) = P(Y,X)$ então temos que $P(X|Y) * P(Y) = P(Y|X) * P(X)$, então por meio dessa equação é dada o Teorema de Bayes:

$$\boldsymbol{P(Y|X) = P(X|Y) * P(Y)/P(X)} \quad \textbf{(2.2)}$$

A partir desse teorema que o algoritmo de classificação Naive Bayes se baseia, Yang e Liu (1999) afirma que na categorização de documentos esse classificador usa da probabilidade condicional das palavras e das classes para poder estimar a probabilidade de uma classe dado um documento.

O classificador Naive Bayes é considerado “ingênuo” porque ele assume que os atributos são todos independentes e não considera a probabilidade condicional de um atributo dada uma classe, dos outros atributos da mesma classe.

Sendo assim, Berton (2011) apresenta a equação do algoritmo Naive Bayes como:

$$\boldsymbol{P(X|Y) = \frac{P(Y)\prod_{i=1}^{N} P(X_i|Y)}{P(X)}} \quad \textbf{(2.3)}$$

2.2.3.2 Nearest Neighbor

O Nearest Neighbor é uma técnica relativamente simples, que parte da ideia de uma visão geométrica de dois pontos em um espaço multidimensional, ou seja, os casos são representados como pontos em um espaço geométrico em que a similaridade é definida por meio da distância entre os pontos, os quais são ordenados e a classe do caso mais próximo é atribuída ao novo caso. De acordo com Von Wangenheim e Von Wangenheim (2003), realizando a representação dos casos em um modelo geométrico, a distância espacial entre os casos representa a similaridade entre eles.

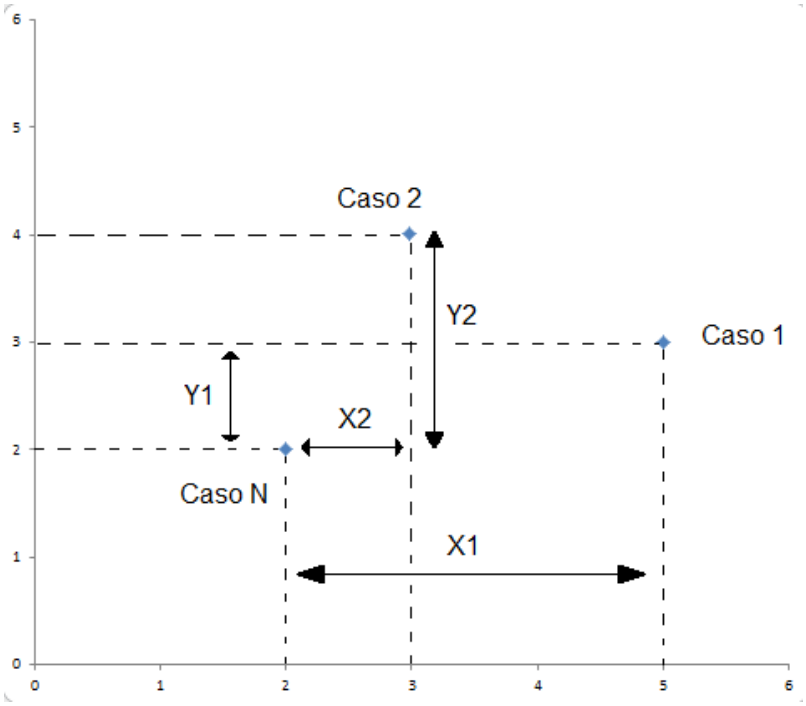


Figura 2.2 - Distância Nearest Neighbor (VON WANGENHEIM; VON WANGENHEIM, 2003).

A figura 2.2 demonstra um modelo geométrico bidimensional, em que cada eixo representa um atributo dos casos.

Considere que haja apenas 3 casos, o caso N que é o novo caso, e os casos 1 e 2 que representam casos da base, cada caso contem dois atributos, representados pelos eixos no gráfico. Como pode ser visto no gráfico, a distância X1 do caso N para o caso 1 é de 3 unidades e a distância Y1 é 1 unidade, por sua vez,

a distância X2 entre o caso N para o caso 2 é 1 unidade e a distância Y2 é de 2 unidades.

Considerando os valores do gráfico, então se pode calcular a distância Nearest Neighbor.

A distância D1 entre o Caso N para o Caso 1 é: D1 = X1 + Y1 = 3 + 1 = 4

A distância D2 entre o Caso N para o Caso 2 é: D2 = X2 + Y2 = 1 + 2 = 3

O resultado que obteve a menor distância é considerado o vizinho mais próximo do caso N, que neste exemplo foi o Caso 2.

O algoritmo Nearest Neighbor tem algumas variações como o Nearest Neighbor ponderado, o normalizado e o K-Nearest Neighbor.

O Nearest Neighbor ponderado considera que cada atributo tenha um peso que significa a importância que ele tem na determinação da similaridade (VON WANGENHEIM; VON WANGENHEIM, 2003).

Assumindo que o peso Wx do atributo X seja 1 e o peso Wy do atributo Y seja 3, podemos aplicar o Nearest Neighbor ponderado no exemplo da figura 2.2.

A fórmula do algoritmo Nearest Neighbor ponderado é apresentado por Koslosky (1999) da seguinte maneira:

$$\boldsymbol{Dist(N,F) = \sum_{i=1}^{n} f(N_i, F_i) \times w_i} \quad \textbf{(2.4)}$$

Sendo que:

- *N* é o novo caso
- *F* são os casos contidos na base de casos
- *n* é o numero de atributos
- *w* representa o peso do atributo
- *f* consiste na função de similaridade local

Sendo assim, podemos calcular o Nearest Neighbor ponderado:

A distância Dp1 entre o Caso N e o Caso 1 é dada por:

$$Dp1 = (X1 * Wx) + (Y1 * Wy) = (3 * 1) + (1 * 3) = 6$$

A distância Dp2 do Caso N para o Caso 2 é dada por:

$$Dp2 = (X2 * Wx) + (Y2 * Wy) = (1 * 1) + (2 * 3) = 7$$

Aplicando o Nearest Neighbor ponderado o vizinho mais próximo do Caso N é o Caso 2, resultado diferente do exemplo anterior que foi aplicado ao Nearest Neighbor básico, devido a essas diferenças de importância dos atributos na determinação da similaridade.

Outra variação é o Nearest Neighbor normalizado que é basicamente um Nearest Neighbor aplicado uma normalização realizada pela divisão da similaridade pelo somatório dos pesos dos atributos.

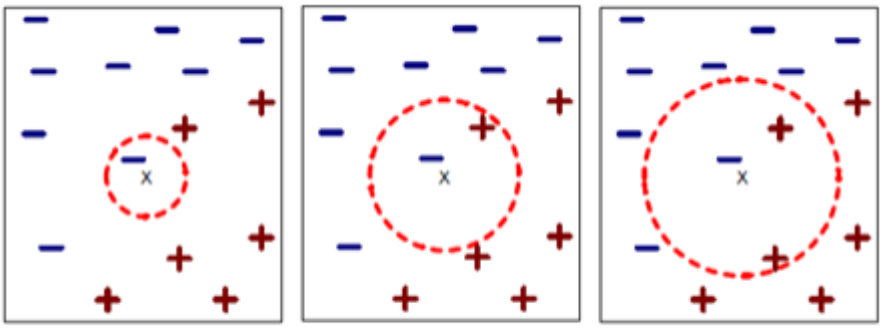
A formula dada por Von Wangenheim e Von Wangenheim (2003) é apresentada a seguir:

$$\boldsymbol{Sim(N,F)} = \frac{\sum_{i=1}^{n} f(N_i,F_i)\times w_i}{\sum_{i=1}^{n} w_i} \qquad \textbf{(2.5)}$$

A versão K-Nearest Neighbor difere do Nearest Neighbor porque ele seleciona uma quantidade K de vizinhos mais próximos e classifica o novo caso com base na classe desses vizinhos.

Dado um novo caso o algoritmo procura os K vizinhos mais próximos por meio da base de casos e de acordo com as classes dos vizinhos mais próximos é atribuída uma classe ao novo caso.

Segundo Berton (2011) o novo caso é classificado de acordo com a classe dos seus vizinhos mais próximos, sendo que a classe do novo caso é determinada pela classe que mais ocorre entre os K vizinhos mais próximos, e em caso de empate será atribuída a classe aleatoriamente entre as classe empatadas.



(a) 1- vizinho mais próximo (b) 2 - vizinho mais próximo (c) 3 - vizinho mais próximo

Figura 2.3 - Exemplo de K-vizinhos mais próximos (TAN; STEINBACH; KUMAR, 2006).

A figura 2.3 mostra exemplos de K assumindo valor 1, 2 e 3. Com o Valor de K=1 o novo caso, representado na figura por X, assume a mesma classe dos casos representados pelos traços azuis. Nesse exemplo, se K for atribuído com o valor 2, os vizinhos mais próximos serão um de cada classe, obtendo-se um empate, então pode ser definido tanto uma quanto a outra. Já com o K obtendo valor 3, a classe de X se torna a mesma classe representada pelas cruzes vermelhas, pois, 2 dos 3 vizinhos são classificadas com essa classe.

A determinação do valor de K deve ser cuidadosamente analisada, se o valor de K for muito pequeno o algoritmo de similaridade tem mais chances de considerar como mais similar um caso da base que tenha dados corrompidos ou incertos, como também o valor de K pode ser muito grande e abranger vários casos, sendo assim com uma maior probabilidade de ser atribuída uma classe ao novo caso de um vizinho distante, ou seja, que não é muito similar (BERTON, 2011).

2.2.3.3 Árvores de decisão

As árvores de decisão tem uma estrutura hierárquica, essa representação traduz uma progressão da extração de conhecimento no sentido de desempenhar uma tarefa, nesse caso, a classificação (RODRIGUES, 2004).

A estrutura da árvore de decisão é formada por raiz, nós de decisão e nós folhas. Cada nó de decisão da árvore implementa uma regra de decisão que divide os exemplos em duas ou mais partições e o nó folha é considerado o terminal que é baseado na condição de parada que determina a classe (KOTHARI; DONG, 2000).

Um nó de decisão realiza um teste, para cada um dos resultados obtidos por esse teste existe uma aresta que leva para uma subárvore que tem as mesmas características de uma árvore.

Uma árvore de decisão pode ser representada como um conjunto de regras. Um exemplo de uma árvore simples que ajuda a decidir se viaja ou não, pode ser lida como:

SE Tem dinheiro = Sim **ENTÃO**
    SE Tem que trabalhar = Sim **ENTÃO**
        Classe = Não Viajar

SENÃO Tem que trabalhar = Não

SE Tem que Estudar = Sim **ENTÃO**

Classe = Não Viajar

SENÃO Tem que Estudar = Não

Classe = Viajar

FIM SE

FIM SE

SENÂO Tem dinheiro = Não

Classe = Não Viajar

FIM SE

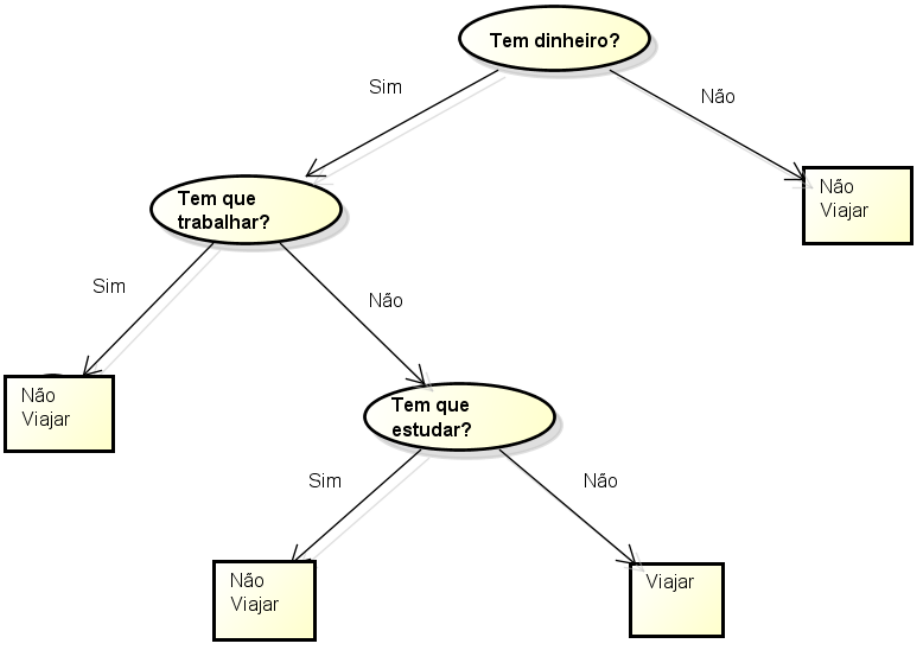


Figura 2.4 - Exemplo de uma árvore de decisão.

A figura 2.4 é a representação da árvore de decisão do exemplo dado acima. Cada eclipse representa um nó de decisão que realiza um teste para definir o caminho que seguirá na árvore, e cada retângulo representa uma classe, ou seja, a decisão se viaja ou não viaja.

Rezende (2005) afirma que são necessários pelo menos cinco passos para a construção de uma árvore de decisão:

1. Sendo *T* um conjunto de treinamento tendo um ou mais casos e considerando que as classes sejam representadas por $\{C_1, C_2, C_3 \ldots\ldots, C_k\}$. Se todos os casos

do conjunto *T* sejam da mesma classe C_j, a árvore de decisão terá somente um nó folha representando a classe C_j.

2. Nas situações que o conjunto *T* não conter exemplos, a árvore será um nó folha e a classe desse nó folha deve ser determinada utilizando informações além do conjunto *T*.
3. Quando o conjunto *T* conter casos de várias classes, basicamente o conjunto *T* deve ser divido em subconjuntos que sejam ou pareçam ser da mesma classe. Para escolher qual dos casos do conjunto *T* representará a árvore é realizado um teste utilizando um indutor, que identificará o nó e cada resultado possível desse nó representa uma aresta.
4. São aplicados os passos 1,2 e 3 para cada subconjunto de treinamento.
5. Realizar a poda da árvore de decisão para aumentar a habilidade de generalização da árvore.

Os algoritmos de indução, mencionado no passo 3, procuram o melhor atributo para se utilizar em um determinado nó da árvore, a escolha desse atributo vai de acordo com qual realiza a melhor divisão do conjunto de treinamento, um dos algoritmos é o ganho máximo (BORGES, 2013).

O ganho máximo determina o nó da árvore como sendo o atributo com o maior ganho de informação, assim resultará em subárvores menores (REZENDE, 2005).

Borges (2013) explica que ganho de informação mede a redução da entropia na divisão entre subconjuntos. Para conseguir medir essa redução é necessário que sejam comparados o grau de entropia do nó pai antes da divisão com o grau de entropia do nó filho depois que foi realizada a divisão. Sendo assim o ganho de informação pode ser dado pela seguinte equação:

$$\boldsymbol{Ganho = entropia(pai) - \sum_{j=1}^{n} \frac{N(v_j)}{N} * entropia(v_j)} \quad \textbf{(2.6)}$$

- Sendo, N o número total de casos do nó pai.
- n o número de atributos.
- $N(v_j)$ é o número de casos ligados ao nó v_j.

Segundo Halmenschlager (2002), a entropia pode ser medida por meio a seguinte equação:

$$\boldsymbol{Entropia(P) = -\sum_{j=1}^{k} p_j * \log_2(p_j)} \quad \textbf{(2.7)}$$

- Sendo p_j a probabilidade relativa da classe C_j no conjunto *T*.
- K representa o número de classes.

Com essas etapas é possível concluir a construção de uma árvore de decisão, porem há a possibilidade de um classificador induzido se ajustar de mais ao conjunto de treinamento, se tornando muito específico. Esse superajuste é chamado de *overfitting* (REZENDE, 2005).

Uma árvore com *overfitting* se especializa demais no conjunto de treinamento, em testes de validação com a utilização de casos da base de treinamento serão obtidos resultados melhores, porem falsos, do que se aplicado os testes de validação em um conjunto separado somente para testes, que simulam melhor a realidade.

Com o intuito de solucionar ou diminuir os problemas de *overfitting* em árvores de decisão alguns algoritmos aplicam uma técnica chamada poda.

De acordo com Rezende (2005), a poda é uma técnica que visa reduzir a complexidade da árvore de decisão, mas ao mesmo tempo melhorar o seu desempenho. Essa técnica normalmente separa um conjunto de treinamentos e outro conjunto para realizar a poda.

Esse é um dos tipos poda, chamado de pós-poda, há também a pré-poda que é realizada enquanto a árvore de decisão esta sendo induzida.

2.2.4 Pós-processamento

Realizada a recuperação dos casos vem a etapa de pós-processamento que melhora a compreensão e faz uma seleção do conhecimento descoberto para passar ao usuário os resultados de maneira mais clara, para realizar sua avaliação.

Segundo Feldman e Sanger (2007), o pós-processamento aplica métodos que filtram informação redundante e agrupa dados que tem quase os mesmos significados, para uma representação completa, compreensiva e ordenada.

Na etapa de pós-processamento também esta incluída a validação do sistema em que são determinadas métricas para mensurar a precisão e avaliar os resultados obtidos.

2.2.4.1 Validação Cruzada

A Validação Cruzada é uma técnica de avaliação da precisão ou o risco de um sistema, essa validação consiste em dividir a base de respostas em dois subconjuntos, a base de treinamento e a base de testes (Ling; ÖZSU, 2009).

Segundo Arlot e Celisse (2010) a principal ideia da validação cruzada é dividir os dados, uma ou varias vezes para estimar o risco do algoritmo, parte dos dados são usados para treinamento e o restante é usado para estimar o risco do algoritmo.

Adiante serão citadas três variações da validação cruzada em relação ao modo como o conjunto de dados é divido e aplicado.

2.2.4.1.1 Hold-Out

Esse método consiste em dividir o conjunto total da base em dois subconjuntos independentes um para treinamento e outro para testes. Normalmente são separados em uma proporção de 2/3 para treinamento e 1/3 para teste (ARLOT; CELISSE, 2010).

Um conjunto independente de teste ajuda a evitar o over-fitting. Para isso os dados disponíveis são divididos em duas partes sem sobreposição. O conjunto de teste é deixado de fora e não é usado durante o treinamento do algoritmo (LING; ÖZSU, 2009).

2.2.4.1.2 K-fold

O *K-fold* realiza a divisão do conjunto total de dados em *k* subconjuntos do mesmo tamanho, um deles é deixado para teste e o restante para treinamento, então é calculada a precisão do modelo. Esse processo se repete *k* vezes alternando o subconjunto de teste (BENGIO; GRANDVALET, 2004).

Para Ling e Özsu (2009) na validação *k-fold* os dados primeiramente são particionados em *k* segmentos iguais. Então *k* iterações de treinamento e validação são realizadas e cada iteração com um segmento diferente para validação.

O consumo de recursos computacionais da validação *k-fold* é bem menor que a validação *leave-one-out*, apresentada na seção a seguir, pois ocorrem *k* validações, se *k* for igual ao número total de amostras -1, então ele se torna igual a validação *leave-one-out* (ARLOT; CELISSE, 2010).

2.2.4.1.3 Leave-one-out

Leave-one-out é um caso parecido com o K-fold, sendo *k* o número total de documentos, são realizados o calculo de precisão *k* vezes, um para cada documento (LING; ÖZSU, 2009).

Kearns e Ron (1997) descrevem o processo do leave-one-out como, executar o algoritmo de aprendizagem *k* vezes, cada vez removendo uma das *k* amostras de treinamento, e testando os resultados na amostra que foi deletada.

2.2.4.2 Bootstrap

O validador Bootstrap, dada uma base de documentos com tamanho *n* uma amostra do Bootstrap é criada por meio de amostragens de *n* instâncias uniformemente (com reamostragem) da base (KOHAVI, 1995).

Por as amostras de treinamento serem criadas com reamostragem a probabilidade de uma amostra ser selecionada é igual para todas.

2.3 Questões subjetivas

Grillo e Gessinger (2010) definem questões subjetivas como sendo questões em que o aluno é requerido a organizar a resposta, a expressar suas ideias, apresentando informações e mostrando o que ele considera mais importante em relação à questão.

A maior diferença entre as questões subjetivas e objetivas é que as questões subjetivas são as que os alunos escrevem/digitam a informação em vez de somente marcar uma opção de uma lista de alternativas, como nas questões objetivas.

No quadro 2.1 são destacadas algumas vantagens e desvantagens na utilização de questões subjetivas e objetivas:

Vantagens	Desvantagens
Questões Objetivas	
• Pode fornecer avaliação direta de muitas habilidades, incluindo capacidade para discriminar, entender conceitos e princípios, julgar possíveis cursos de ação, inferir, raciocinar, completar declarações, interpretar dados e aplicar informação; • Eficiência na administração; • Não requer que os alunos escrevam ou deem respostas elaboradas; • A análise de distratores fornece informação diagnóstica; • Vasta amostragem do domínio do conteúdo; • Pode ser objetivamente registrado.	• Avaliações indiretas limitadas de algumas habilidades, incluindo a capacidade para relembrar, explicar por si mesmo, dar exemplos, expressar ideias, organizar ideias ou construir alguma coisa; • Opções fixas limitam a expressão de ideias ou soluções originais; • Tende a se basear num conhecimento artificialmente estruturado e fechado para a interpretação; • Compreensão de leitura pode interferir quando a leitura não é o objetivo a ser alcançado; • Suscetível à adivinhação.
Questões Subjetivas	
• Apropriado quando o objetivo a ser alcançado requer uma resposta escrita; • Itens e respostas de uma só palavra são relativamente fáceis de construir e se registrar objetivamente; • Itens de respostas curtas (requerendo mais do que uma única palavra) podem avaliar altos níveis de conhecimento e habilidades; • Permite uma série de respostas e soluções originais, fornecendo vasta reflexão sobre as variações no aprendizado; • Ampliar respostas pode avaliar processos complexos: sintetização, organização e ordenação; • Dificuldade para adivinhar corretamente, reduzindo o erro por acaso nos escores.	• A possibilidade de múltiplas respostas lógicas cria problemas de registro e pode reduzir a confiabilidade, requerendo muito tempo de testagem para alcançar uma fidedignidade moderada; • Itens no formato de resposta com uma só palavra são menos apropriados para avaliar habilidades cognitivas elevadas; • Dificuldade em escrever itens que limitem as respostas sem confundir os alunos; • Poucos itens podem ser administrados, resultando na limitação do alcance do conteúdo dado o mesmo tempo de teste; • Habilidade de escrever pode interferir quando a escrita não é o objetivo a ser alcançado.

Quadro 2.1 Vantagens e desvantagens de questões subjetivas e objetivas (TINDAL; HALADYNA, 2009).

Uma das limitações citadas por Santrock (2009) é que “a correção e atribuição de notas consome tempo”. Em grande escala essa limitação se torna ainda mais visível e desmotivadora.

3 Desenvolvimento

A construção do sistema foi dividida em cinco etapas: Construção da base de casos, pré-processamento, recuperação de casos, retenção e validação, detalhadas nas seções a seguir.

3.1 Construção da base de casos

Um caso consiste em uma experiência real no qual o problema já foi resolvido por um especialista (TELLES et al., 2006). Por meio dessa definição é possível compreender melhor como é importante a elaboração de uma base de casos confiável para um sistema RBC.

Fernandes (2003) considera os seres humanos como grandes solucionadores de problemas e que com a experiência adquirida na resolução de dos problemas a performance deles melhora cada vez mais. Igualmente são os sistemas RBCs, quanto mais casos houver na base de casos melhor será seu desempenho.

Os casos, que nesse projeto são as respostas subjetivas já corrigidas, foram adquiridos por meio de avaliações aplicadas por um especialista da área de psicologia que as corrigiu e as classificou com uma nota no valor de 0,00 a 1,25. Todas as respostas foram corrigidas ortograficamente evitando que os erros de ortografia se tornassem “ruídos” nos dados da base. A base inicial contem 93 respostas de uma mesma questão, respondida por alunos diferentes. As notas foram categorizadas por valores entre “A”, “B”, “C”, “D” e “E”, sendo “A” a melhor nota e “E” a pior nota. Para fazer essa categorização das notas foram definidas faixas de valores, respostas com nota entre:

- 0,00 a 0,25 assumem a categoria “E”;
- 0,26 a 0,50 assumem a categoria “D”;
- 0,51 a 0,75 assumem a categoria “C”;
- 0,76 a 0,99 assumem a categoria “B”;
- 1,00 a 1,25 assumem a categoria “A”.

Após a obtenção das respostas já corrigidas, elas passaram por um processo de indexação para facilitar a comparação da similaridade entre a resposta à ser corrigida e as respostas da base de casos. Segundo Von Wangenheim e Von Wangenheim (2003) índices são todos os termos que a informação é relevante para estabelecer a similaridade entre o caso de entrada e os casos da base.

A indexação da base de casos inicial foi aplicada realizada submetendo às respostas subjetivas as seguintes tarefas de pré-processamento: a tokenização, remoção de *stopwords*, *stemming* e representação numérica. O resultado das tarefas de pré-processamento foi uma lista de termos com suas respectivas ocorrências nas respostas, cada termo dessa lista é um índice que será utilizado na comparação da similaridade. Os índices e suas ocorrências foram armazenados em uma base de dados do Mysql, juntamente com a resposta original e sua nota.

3.2 Pré-processamento

Com a base de casos já construída é necessário que a nova resposta inserida pelo usuário passe por todas as tarefas de pré-processamento que foram utilizadas para construção da base de casos inicial, exceto a remoção de *stopwords* que não é necessária porque os índices já foram definidos então os termos que não estiverem entre a lista de índices serão descartados.

3.2.1 Tokenização

De acordo com Alencar (2013) sendo o documento pontuado corretamente, basta dividi-lo quando tiver ocorrências de espaços em branco e caracteres de pontuação. Para realizar a tokenização da nova resposta foram utilizados como separadores o espaço em branco e os caracteres: vírgula, ponto final, ponto e vírgula, parênteses, dois pontos e aspas duplas. A seguir é apresentado um exemplo de tokenização:

Entrada: O processamento de documentos textuais envolve vários desafios. Um desafio é o formato do texto que é sem nenhuma estrutura ou pouca estruturação, sendo ainda, o fator que mais o difere da mineração de dados.

Saída: O processamento de documentos textuais envolve vários desafios Um desafio é o formato do texto que é sem nenhuma estrutura ou pouca estruturação sendo ainda o fator que mais o difere da mineração de dados

Cada segmentação destacada representa uma unidade distinta e significativa, chamada *token*.

3.2.2 Stemming

O objetivo de encontrar os *stems* dos termos é obter o elemento único que representa várias palavras relacionadas, ou seja, uma semântica única. Para isso é necessário encontrar o radical das palavras removendo as variações de indicação de plural, flexões verbais ou variantes que são sintaticamente similares (VARGAS, 2012). Por exemplo, as palavras *Caminhar, Caminhamos, Caminhou* e *Caminhando*, podem ser representadas pelo mesmo *stem Caminh.*

O algoritmo *Snowball*, procura o sufixo das palavras dividindo o final delas em regiões por meio de regras preestabelecidas. No algoritmo são definidas três regiões: R1, R2, RV.

Segundo Porter (2013), R1 é definida como a região depois da primeira não-vogal seguida por uma vogal, ou assume valor nulo caso não haja a não-vogal. A região R2 segue a mesma regra da R1, exceto por aplica-la dentro da região R1 ao invés da palavra inteira. A região RV representa a parte em que, se a segunda letra for uma consoante RV é região depois da próxima vogal, ou se as duas primeiras letras forem vogais então RV representa a região depois da próxima consoante, ou ainda se as primeiras letras forem uma consoante seguida por uma vogal é considerada como RV a região depois da terceira letra.

Com as regiões definidas, por meio de uma lista de sufixos já estabelecida no algoritmo os sufixos são removidos, ou em alguns casos são substituídos. Primeiramente são removidos os sufixos mais comuns, seguido pelo dos verbos e então é feita uma retirada dos sufixos residuais.

3.2.3 Representação numérica

Os documentos devem ser representados de forma que os algoritmos para recuperação de casos possam agir, no entanto, a maioria deles atua em dados estruturados e numéricos (ALENCAR, 2013). Então para poder calcular a relevância (Tf-Idf) de cada termo das novas respostas, foi necessária a transformação dos termos por meio de um processo de contagem para descobrir a quantidade de vezes em que eles aparecem em cada resposta.

Existem varias formas de se calcular o Tf-Idf, mas nesse projeto será utilizado a formulação proposta por Bell, Moffat e Witten (1999):

$$\boldsymbol{tf - idf_{t,d}} = (\mathbf{1} + \mathbf{log}(\boldsymbol{tf_{t,d}}))\ \mathbf{x}\ \mathbf{log}\left(\mathbf{1} + \frac{|\boldsymbol{D^{(c)}}|}{|\boldsymbol{D_t^{(c)}}|}\right) \quad \textbf{(3.1)}$$

Em que *t* é definido como termo, para cada documento (resposta) *d* que pertence ao conjunto de documentos *c*. O $tf_{t,d}$ é a quantidade de ocorrências de um termo *t* que pertence ao documento *d*. $D^{(c)}$ é o numero de documento de um conjunto *c* e $D_t^{(c)}$ é a quantidade de documentos em que o termo *t* ocorre pelo menos uma vez.

O Tf-Idf é calculado para todos os termos das respostas e o sistema recalcula sempre que um novo caso é adicionado à base, pois a parte *idf* da formula necessita da quantidade de documentos em que um termo *t* ocorre pelo menos uma vez, que pode ser alterado sempre que adicionado um novo documento a base.

3.3 Recuperação de casos

A recuperação de casos é a etapa em que a nova resposta é comparada com os casos armazenados na base de casos em busca da resposta já corrigida que seja mais semelhante com a nova resposta.

A semelhança entre as respostas é encontrada por meio da medida de similaridade, aplicando-se o cálculo de similaridade aos pares de nova resposta com cada caso da base de casos. A similaridade é calculada neste projeto é por meio da Similaridade Local e Similaridade Global, essas medidas serão mais detalhadas nas seções 3.3.1 e 3.3.2. Posteriormente, com a similaridade calculada, os valores são ordenados e as notas dos mais similares são indicadas como possíveis classificações da nova resposta.

3.3.1 Similaridade Local

O cálculo da similaridade local é uma medida de similaridade aplicada analisando um par de termos individualmente, sem levar em consideração o documento como um todo. A similaridade local utilizada nesse projeto é apresentada por Von Wangenheim e Von Wangenheim (2003) pela seguinte equação:

$$\boldsymbol{V(v_i, v_j)} = \mathbf{1} - \frac{|\boldsymbol{v_i - v_j}|}{\boldsymbol{v_{max} - v_{min}}} \quad \textbf{(3.2)}$$

- Em que V é a similaridade local entre dois termos v_i, v_j;
- v_{max} é o maior valor que aquele termo pode assumir nos documentos;
- v_{min} o menor valor que o termo pode assumir.

Essa medida de similaridade assume a quantidade de ocorrências do termo como o principal fator de comparação, sendo que, quanto mais próximos forem o valor das ocorrências mais similares serão o termo da resposta de entrada com o termo do caso. A figura 3.1 mostra o exemplo de uma função linear representando a similaridade local.

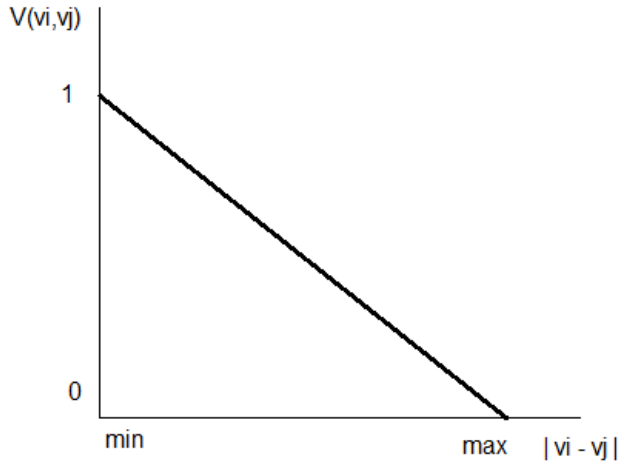


Figura 3.1 - Exemplo função linear (VON WANGENHEIM; VON WANGENHEIM, 2003).

Com a similaridade local calculada temos então a similaridade individual de cada termo, mas ainda é necessário calcular a similaridade entre respostas não somente entre seus termos e isso será realizado por meio da similaridade global.

3.3.2 Similaridade Global

Na aplicação da similaridade local sobre os termos dos documentos se obteve a similaridade dos mesmos separadamente. Já a similaridade global tem uma abordagem mais ampla, abrangendo os termos do documento integralmente.

Mesmo sendo conceitos diferentes, a similaridade local faz parte do calculo da similaridade global que para encontrar a similaridade calcula a média ponderada das similaridades locais. Von Wangenheim e Von Wangenheim (2003) apresentam a equação de similaridade global como:

$$\boldsymbol{sim(Q_i, C_i) = \frac{\sum_{i=1}^{n} f(Q_i, C_i) \times w_i}{\sum_{i=1}^{n} w_i}} \quad \textbf{(3.3)}$$

A similaridade local entre Q_i, C_i é representada por $f(Q_i, C_i)$, em que Q_i é o valor do termo da resposta de entrada e C_i é o valor do mesmo termo de um caso armazenado na base. Essa similaridade é multiplicada pelo peso w_i e o resultado é dividido pelo soma de todos os pesos. O peso dos termos neste projeto foi estimado por meio do tf-idf apresentado anteriormente.

Esse cálculo é baseado no algoritmo de classificação *Nearest Neighbor* e resulta na média da similaridade local de todos os termos da resposta, levando em consideração o peso que representa a relevância dos termos para determinada resposta e sua respectiva nota.

O resultado do calculo da similaridade será um valor na faixa de 0 até 1, onde 1 é a similaridade total e 0 os casos não tem similaridade nenhuma. As similaridades então são ordenadas em ordem decrescente deixando nas primeiras posições os casos mais similares.

3.4 Retenção

Para a realização do aprendizado ou retenção do novo caso à base é necessário que seja realizada uma avaliação da correção que pode ser automática ou com participação do usuário (MENDES, 2009). Neste trabalho, a avaliação da correção será feita pelo usuário. O sistema exibirá ao usuário cinco sugestões de correção para a resposta inserida, o qual ele poderá visualizar as respostas e a nota para analisar se alguma delas condiz com a nova resposta. A figura 3.2 mostra a tela onde o usuário irá visualizar as sugestões de correção:

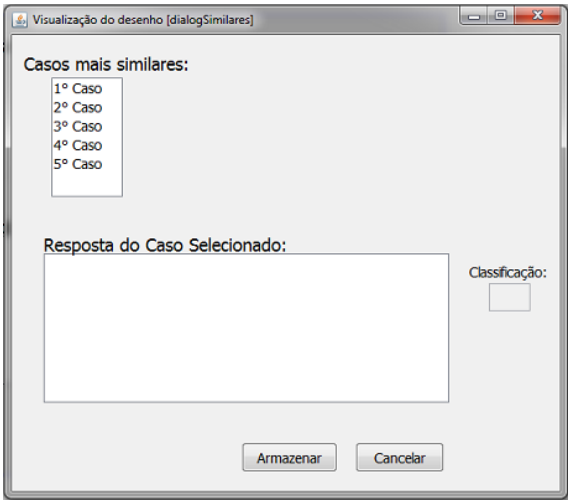


Figura 3.2 – Tela de visualização dos casos similares

Ao usuário clicar em um dos casos da lista dos casos mais similares, a resposta e a classificação do respectivo caso serão exibidos, então ao analisar que algum dos casos é bem semelhante a resposta nova, o usuário pode reter o novo caso na base com a mesma solução, clicando no botão Armazenar.

Ao acionar o botão Armazenar, o sistema identifica a nota indicada no campo Classificação e guarda na base de casos a nota juntamente com a resposta original inserida pelo usuário e as ocorrências dos termos nela contidos.

A retenção ou aprendizado do novo caso é um processo que deve ser realizado mesmo que nenhuma das sugestões de correção seja aceita pelo usuário, para o sistema aprender com seus erros (AAMODT; PLAZA, 1994). Se nenhuma das indicações para a solução for aceita, o usuário pode sugerir uma nota para ser armazenada.

3.5 Validação

O método para validação aplicado no sistema foi o Validação Cruzada em sua versão Hold-out. A base de casos foi divida em dois subconjuntos, 2/3 da base total para o treinamento do algoritmo e 1/3 para a utilização nos testes.

Cada caso do subconjunto de teste foi aplicado para a correção no sistema, que utilizou somente do subconjunto de treinamento para realizar as correções. Em todos os testes, cada sugestão de nota dada pelo sistema foi armazenada e atribuída como uma sugestão certa ou errada, de acordo com a nota correta

disponibilizada pelo especialista. A figura 3.3 mostra um gráfico com a quantidade de acertos e erros que o sistema obteve:

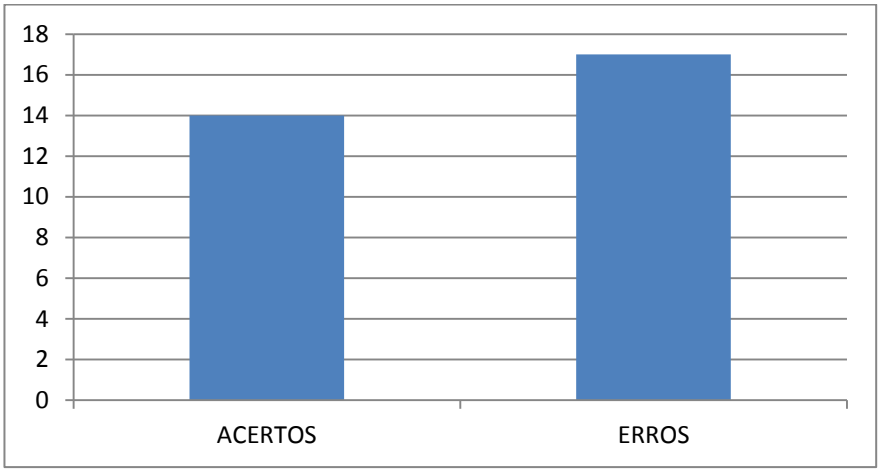


Figura 3.3 – Acertos e erros do sistema com o caso mais similar

Na figura 3.3, são considerados acertos os testes em que o primeiro caso similar sugerido pelo sistema tem a mesma categoria dada pelo especialista e os erros quando assume qualquer outro valor. Por meio desses resultados obtidos pela validação cruzada, pode-se calcular que o sistema obteve uma precisão de 45,16%, nesse cenário em que é somente utilizado o caso mais similar recuperado. Sendo *P* a precisão esta é calculada pela seguinte equação:

$$\boldsymbol{P = \frac{casos\ relevantes\ recuperados}{total\ de\ casos\ recuperados}} \qquad \textbf{(3.4)}$$

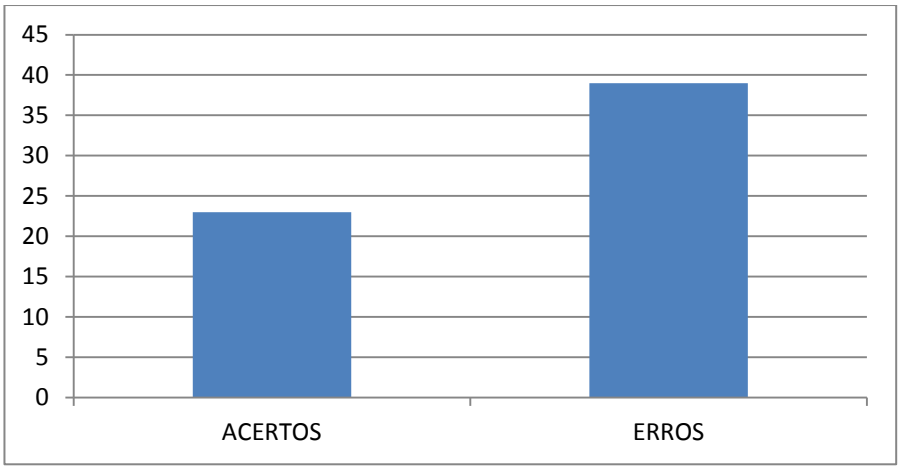


Figura 3.4 – Acerto e erros do sistema entre os 2 casos mais similares

Na Figura 3.4 são apresentados os acertos e erros considerando os dois casos mais similares recuperados. A precisão nessa situação é de 37,09%, houve uma diminuição de 8,07% em comparação com a precisão em que foi levado em consideração somente o caso mais similar.

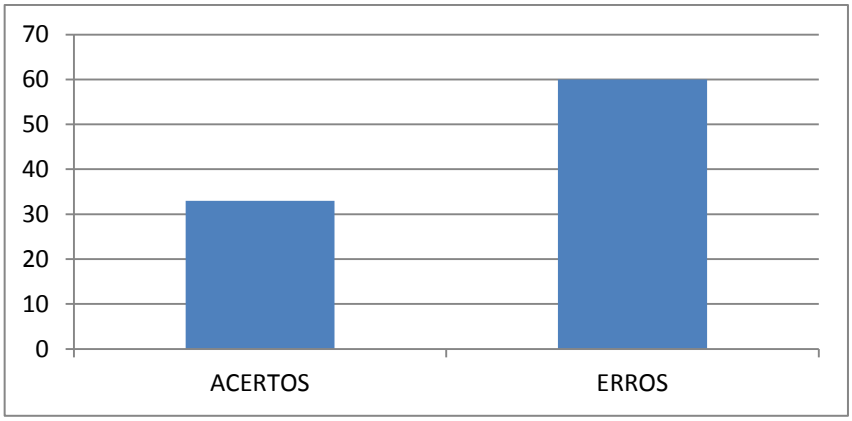


Figura 3.5 - Acerto e erros do sistema entre os 3 casos mais similares

A figura 3.5 apresenta os acertos e erros do sistema considerando os 3 casos mais similares, a precisão apurada neste teste é de 35,48%. Com isso é possível confirmar a tendência de que o aumento no número de casos recuperados diminui a precisão do sistema. Porem, apresenta mais casos ao usuário, que pode identificar que o caso mais similar, não é realmente o que melhor representa a resposta que ele inseriu.

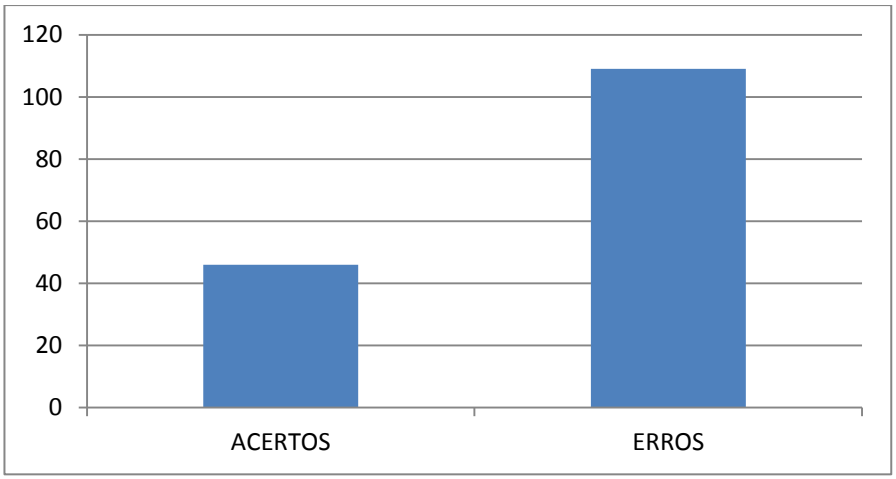


Figura 3.6 - Acerto e erros do sistema entre os 5 casos mais similares

Na figura 3.6 fica ainda mais nítida a diminuição da precisão com o aumento de casos recuperados, a precisão adquirida nesta situação é de 29,67%. Um dos motivos pelo qual o sistema tem essa deficiência é porque a base de casos é pequena, sendo que são no total 93 casos, mas 62 casos fazem parte do conjunto de casos de treinamento e 31 casos são o conjunto de casos para teste. Ainda, a base tem uma especificidade de que 31,18% das notas das respostas são “A”, tendo assim uma tendência a corrigir melhor respostas que tem a nota “A”, do que respostas que tem outras notas.

Nestas estimativas de precisão não são levadas em consideração a avaliação do usuário sobre os casos sugeridos. Sendo selecionado entre os 3 casos mais similares, se considerarmos que o usuário analise e escolha uma sugestão com a mesma nota dada pelo especialista temos que 87,09% dos testes tem pelo menos um caso sugerido corretamente.

4 Conclusão e Trabalhos Futuros

As questões subjetivas têm suas vantagens, porém, também há as desvantagens, como a demora e o custo para a correção, pois estão em formato textual e muitas vezes tem grande extensão. O trabalho tem como principal objetivo aumentar a viabilização da aplicação de questões subjetivas, mitigando essas desvantagens.

Para isso, foi proposto, um sistema RBC que por meio da semelhança da resposta inserida para correção com as respostas já corrigidas previamente por um especialista. O sistema foi integrado com técnicas de mineração de textos que possibilita ao sistema eliminar ruídos da base de casos e os novos casos inseridos. O sistema ainda conta com um mecanismo de retenção do conhecimento adquirido em cada uma das correções feita por ele.

Conforme os resultados apresentados na validação, os acertos na correção de questões subjetivas utilizando o sistema RBC foram de 45,16%, considerando somente o primeiro caso sugerido pelo sistema, buscando a classificação correta entre os cinco casos sugeridos pelo sistema, obteve-se uma taxa de acerto de 29,67%, esses dois casos sem a avaliação do usuário. Um resultado abaixo do ideal, porem, se analisar as três sugestões dadas pelo sistema, conseguiu-se observar que 87,09% dos testes realizados continham pelo menos um caso sugerido que era satisfatoriamente similar com a resposta de entrada.

Portanto, o sistema necessita que o usuário avalie as sugestões dadas pelo sistema dizendo se são úteis ou não, e ajudar o sistema a aprender, atribuindo uma nota em situações em que o sistema não encontre nenhum caso satisfatório.

Alguns trabalhos futuros que são propostos e que poderão tornar o sistema mais preciso e o processo mais automático são apresentados adiante:

- Implantar um corretor ortográfico para evitar que palavras com o mesmo significado sejam consideradas como se fossem distintas. Assim deixando o processo de stemming mais eficiente, pela padronização do modo de escrita(CONGRESSO DA SBC, 2006).
- Analisar semanticamente as respostas, ou seja, analisar o significado das palavras no contexto apresentado, que podem ser mais complexos pela combinação de palavras (MORAIS; AMBRÓSIO, 2007).

- Realizar análises sintáticas, para conhecer a estrutura do conjunto de palavras e como elas podem se unir para produzir sentenças, em que algumas palavras podem alterar o significado de outras quando utilizadas em conjunto (MORAIS; AMBRÓSIO, 2007).

Referências

AAMODT, Agnar; PLAZA, Enric. Case-based reasoning: foundational issues, methodological variations, and system approaches. **Ai Communications,**London, p. 39-59. 01 mar. 1994.

ALENCAR, Aretha Barbosa. **Visualização da evolução temporal de coleções de artigos científicos.** 2013. 157 f. Tese (Doutorado) - Icmc Usp, São Carlos, 2013.

CONGRESSO DA SBC, 26., 2006, Campo Grande. **Etapas do Processo de Mineração de Textos:** Uma abordagem aplicada a textos em Português do Brasil. Campo Grande: Workshop de Computação e Aplicações, 2006.

ARLOT, Sylvain; CELISSE, Alain. **A survey of cross-validation procedures for model selection.** Lille: Irma, 2010.

BARION, Eliana Cristina Nogueira; LAGO, Decio. Mineração de textos: text mining. **Revista de Ciências Exatas e Tecnologia**, Valinhos, n. , p.123-140, 08 dez. 2008.

BELL, Timothy C.; MOFFAT, Alistair; WITTEN, Ian H.. **Managing gigabytes: compressing and indexing documents and images.**San Francisco: Morgan Kaufmann, 1999.

BENGIO, Yoshua; GRANDVALET, Yves. No Unbiased Estimator of the Variance of K-Fold Cross-Validation. **Journal Of Machine Learning Research.** Montreal, p. 1089-1105. set. 2004.

BERTON, Lilian. **Caracterização de classes e detecção de outliers em redes complexas.** 2011. 844 f. Dissertação (Mestrado) - Curso de Ciências de Computação e Matemática Computacional, Usp, São Carlos, 2011.

BORGES, Fábbio Anderson Silva. **Extração de características combinadas com árvore de decisão para detecção e classificação dos distúrbios de qualidade de energia elétrica.** 2013. 118 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Usp, São Carlos, 2013.

FELDMAN, Ronen; HIRSH, Haym. Exploiting Background Information in Knowledge Discovery from Text. **Journal Of Intelligent Information Systems**, v. 9, n. 1, p.83-97, 01 jul. 1997.

FELDMAN, Ronen; SANGER, James. **The Text Mining Handbook:**Advanced Approaches in Analyzing Unstructured Data. New York: Cambridge University Press, 2007.

FERNANDES, Anita Maria Da Rocha. **Inteligencia Artificial:** Noções Gerais. Florianópolis: Visual Books, 2003.

GRILLO, Marlene Correro; GESSINGER, Rosana Maria. **Por que falar ainda em avaliação?** Porto Alegre: Universitária Pucrs, 2010.

HACK, Augusto Fredigo et al. **Text minig.** Florianópolis: Ufsc, 2013.

HALMENSCHLAGER, Carine. **Um algoritmo para indução de árvores e regras de decisão.** 2002. 112 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2002.

KAO, Anne; POTEET, Stephen R.. **Natural language processing and text mining.** London: Springer, 2007.

KEARNS, Michael; RON, Dana. **Algorithmic Stability and Sanity-Check Bounds for Leave-One-Out Cross-Validation.** New York: Acm, 1997.

KOHAVI, Ron. **A Study of CrossValidation and Bootstrap for Accuracy Estimation and Model Selection.** Stanford: Ijcai, 1995.

KOLODNER, Janet L. **An introduction to case-based reasoning.**Atlanta: Georgia Institute Of Technology, 1992.

KOSLOSKY, Marco Antônio Neiva. **Aprendizagem baseada em casos:** um ambiente para ensino de lógica de programação. 1999. 2 v. Dissertação (Mestrado) - Curso de Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 1999.

KOTHARI, Ravi; DONG, Ming. **Decision trees for classification:** a review and some new results. Cincinnati: Word Scientific, 2000.

MÁNTARAS, Ramon López de et al. **Retrieval, reuse, revision, and retention in case-based reasoning.** London: Cambridge University Press, 2005.

MATSUBARA, Edson Takashi.**O algoritmo de aprendizado semi-supervisionado CO-TRAINING e sua aplicação na rotulação de documentos.** 2004. 83 f. Dissertação (Mestrado) - Icmc Usp, São Carlos, 2004.

MATTAR, Fauze Najib. **Pesquisa de Marketing.** São Paulo: Elsevier Inc, 1994.

MCCALLUM, Andrew; NIGAM, Kamal. **A Comparison of Event Models for Naive Bayes Text Classification.** Pittsburgh: Aaai Press, 1998.

MENDES, Marcio Almeida.**Sistematização da Assistência de Enfermagem usando Raciocínio Baseado em Casos implementado em JAVA.** 2009. 69 f. Dissertação (Mestrado) - Icmc Usp, São Paulo, 2009.

MORAIS, Edison Andrade Martins; AMBRÓSIO, Ana Paula L..**Mineração de textos.** Goiânia: Universidade Federal de Goiás, 2007.

MURTHY, Kavi Narayana. **Advances in automatic text categorization.** Hyderabad: University Of Hyderabad, 2003.

NAHM, Un Yong; MOONEY, Raymond J.. **Text mining with information extration.** Austin: Department Of Computer Sciences, 2002.

NÜNEZ, Isauro Beltran. **INSTRUMENTOS DE AVALIAÇÃO:** PROVAS DE PERGUNTAS DISCURSIVAS. Disponível em: <www.sistemas.ufrn.br>. Acesso em: 19 jun. 2013.

PORTER, Martin F.. **Portuguese stemming algorithm.** Disponível em: <http://snowball.tartarus.org>. Acesso em: 14 out. 2013.

REFAEILZADEH, Payam; TANG, Lei; LIU, Huan. Cross-validation. In: LIU, Ling; ÖZSU, M. Tamer. **Encyclopedia of Database Systems.** Tempe: Springer Us, 2009. p. 532-538.

RODRIGUES, Luciana Silveira. **O USO DE SOFTWARE EDUCACIONAL NO ENSINO FUNDAMENTAL DE MATEMÁTICA E A APRENDIZAGEM DO SISTEMA DE NUMERAÇÃO DECIMAL POR ALUNOS DE 3ª SÉRIE.** 2006. 175 f. Dissertação (Mestrado) - Curso de Educação, Universidade CatÓlica Dom Bosco, Campo Grande, 2006.

RODRIGUES, Marco António Dos Santos. **Árvores de classificação.**Açores: Universidade de Açores, 2004.

SALTON, Gerard; MCGILL, Michael J.. **Introduction to modern information retrieval.** 2. ed. Michigan: Mcgraw-hill, 1983. 448 p.

SANTROCK, John W.. **Psicologia Educacional.** São Paulo: Mcgraw Hill Brasil, 2009.

SELLTIZ, Claire et al. **Métodos de pesquisa das relações sociais.**São Paulo: Hender, 1965.

SILVA, Valdinei Freire da et al. **RECONHECIMENTO DE ESCRITA BASEADO EM REDES NEURAIS ARTIFICIAIS UTILIZANDO B-SPLINES E TDF.** Bauru: Escola Politécnica da Universidade de São Paulo, 2003.

SOMMER, Robert; SOMMER, Barbara Baker. **A practical guide to behavioral research:** tools and techniques. New York: Oxford University Press, 2002.

TAN, Pang-ning; STEINBACH, Michael; KUMAR, Vipin. **Introduction to data mining.** Minnesota: Addison-wesley, 2006.

TELLES, Viviane Carra et al. **Sistema de Raciocínio Baseado em Casos para Recomendação de Programa Alimentar.** Pelotas: Resi, 2006.

TINDAL, Gerald; HALADYNA, Thomas M.. **Large-scale Assessment Programs for All Students:** Validity, Technical Adequacy, and Implementation. New Jersey: Psychology Press, 2009.

VARGAS, Rosa Nathalie Portugal.**Identificação da cobertura espacial de documentos usando mineração de textos.** 2012. 140 f. Dissertação (Mestrado) - Icmc Usp, São Carlos, 2012.

VON WANGENHEIM, Christiane Gresse; VON WANGENHEIM, Aldo.**Raciocínio baseado em casos.** Barueri: Manole, 2003.

YANG, Yiming; LIU, Xin. **A re-examination of text categorization methods.** Pittsburgh: Acm Press, 1999.