CENTRO UNIVERSITÁRIO UNIVATES
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE ENGENHARIA DA COMPUTAÇÃO

DANIEL HARTMANN

IDEIA – SISTEMA DE INTELIGÊNCIA COLETIVA

Lajeado
2011

DANIEL HARTMANN

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

IDEIA – SISTEMA DE INTELIGÊNCIA COLETIVA

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Tecnológicas do Centro Universitário UNIVATES, como parte dos requisitos para a obtenção do título de bacharel em Engenharia da Computação.
Área de concentração: Inteligência Artificial

ORIENTADOR: Fabrício Pretto

Lajeado

2011

DANIEL HARTMANN

IDEIA – SISTEMA DE INTELIGÊNCIA COLETIVA

Este trabalho foi julgado adequado para a obtenção do título de bacharel em Engenharia da Computação do CETEC e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____

Prof. Fabrício Pretto, UNIVATES

Mestre pela PUCRS – Porto Alegre, Brasil

Banca Examinadora:

Prof. Marcelo de Gomensoro Malheiros, UNIVATES

Mestre pela UNICAMP – Campinas, Brasil

Prof. Evandro Franzen, UNIVATES

Mestre pela UFRGS – Porto Alegre, Brasil

Coordenador do curso de Engenharia da Computação :_____

Prof. Marcelo de Gomensoro Malheiros

Lajeado, julho de 2011.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

"De sozinho a nada
— um passo, um espaço de lapso,
um lapso nas arestas, uma coisa
de nada. " Paulo Leminski

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

RESUMO

A interação entre pessoas por meio de tecnologias como a Internet, o telefone celular e outros dispositivos eletrônicos tem aumentado gradativamente nos últimos tempos. Isso ocorre principalmente através da Internet, seja por meio de portais de relacionamento como Twitter ou Facebook, ou através de sites de compras *online*, nos quais as pessoas têm adotado o meio eletrônico como ferramenta para realização de suas tarefas profissionais, educacionais e de lazer. De posse da informação gerada pela interação dos usuários com os serviços citados, a Internet passou a oferecer e implantar serviços que possibilitam a análise de comportamentos de forma a conhecer melhor seus usuários para informá-los sobre conteúdos que lhes sejam relevantes. Exemplos disso são sistemas de busca que melhoram seus resultados conforme sua utilização e geram recomendações de produtos similares em lojas *online*. Esses recursos pertencem à área de Inteligência Coletiva, a qual consiste na criação de uma inteligência baseada no coletivo, ou seja, uma inteligência formada por pequenas ações individuais somadas. Dessa forma, não é necessário que os sistemas implementem cálculos complexos para verificar a preferência de seus usuários, ao invés disso é feita uma análise sobre suas ações afim de prever seus interesses e suas futuras interações. O presente trabalho apresenta uma descrição dos conceitos de Inteligência Coletiva, sua aplicabilidade, bem como o desenvolvimento de um portal *web* que faz uso dessa tecnologia para demonstrar como esses conceitos podem ser aplicados.

Palavras-chaves: Internet, Inteligência Coletiva, Extração de Palavras-chave, Sistema de Recomendação.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

ABSTRACT

The interaction between people through technologies such as the Internet, the cellphone and another electronic devices have been growing gradually in the last years. That occurs mainly through the Internet, whether by social networking portals like Twitter or Facebook, or through online shop sites, in which people have adopted the electronic method as tool to perform their professional, educational and leisure tasks. With the information generated through user interaction with the cited services, the Internet began offering and deploying services which enable behavior analysis in order to know better their users to inform them about relevant content. Examples for this are search systems which improve their results according to their use and generate similar product recommendation at online stores. This features belong to the Collective Intelligence area, which consists on intelligence creation based upon the collective, in other words, intelligence made by summed small individual actions. This way, it is not necessary that the systems implements complex calculation to verify their users preferences and their future interactions. This work shows a description of Collective Intelligence concepts, its applicability, as well a development of a web portal which makes use of this technology to show how these concepts may be applied.

Keywords: Internet, Collective Intelligence, Keyword Extraction, Recommendation System.

SUMÁRIO

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

LISTA DE FIGURAS

LISTA DE TABELAS

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

LISTA DE ABREVIATURAS

AJAX: Asynchronous JavaScript and XML

IC: Inteligência Coletiva

JSP: JavaServer Pages

NLTK: Natural Language Toolkit

OCR: Optical Character Recognition

PHP: PHP Hypertext Preprocessor

RSLP: Removedor de Sufixos da Língua Portuguesa

1 INTRODUÇÃO

A Internet como é utilizada hoje já não tem mais o mesmo comportamento e a mesma escassez de funcionalidades como era quando foi difundida fora dos círculos acadêmicos, no início dos anos 90. Naquela época, utilizavam-se basicamente páginas estáticas sem qualquer interação com o usuário, exceto o ato do clique do *mouse* para redirecionamento a outras páginas, também estáticas.

Mais recentemente, a partir do ano 2000, a interação e a geração de conteúdos pelos usuários em sistemas *web* já se tornaram corriqueiras e passaram a ser consideradas requisitos para um sistema ou um portal se tornar competitivo.

Sendo assim, qualquer serviço desenvolvido para a *web* que não possuísse recursos de interação ou que não pudesse ser utilizado para disponibilização de um certo tipo de conteúdo – como vídeos, fotos ou textos – não faria tanto sucesso entre os usuários, quando comparado a um serviço com os recursos citados.

Esses recursos foram desenvolvidos devido à evolução das ferramentas utilizadas na criação de *sites*. Surgiram funcionalidades como a requisição assíncrona de dados – AJAX (Asynchronous JavaScript and XML), a qual cria uma conexão transparente entre a linguagem interpretada no navegador, geralmente linguagem JavaScript, e a linguagem interpretada no servidor, como as linguagens PHP (PHP Hypertext Preprocessor), JSP (JavaServer Pages) e Python.

Por trás dos serviços mais avançados estão conceitos de aprendizado de máquina, sistemas de recomendação e produção coletiva. Assuntos estes que compõem a área conhecida como Inteligência Coletiva.

Mesmo que muitas vezes não percebidos pelos usuários, esses recursos tendem a facilitar a navegação e incentivar a utilização dos serviços. Para os *sites* de lojas, por exemplo, funcionalidades como recomendação de produtos são indispensáveis, dado que já são comuns nesse tipo de sistema.

Além de lojas virtuais que praticam comércio eletrônico, também podem ser citados portais de busca. Por exemplo, o Google, o portal mais acessado segundo dados da empresa de estatísticas sobre acessos a *websites*, a Alexa, implementa recursos de aprendizagem de máquina desde sua primeira versão. Portanto torna-se impraticável, baseando-se nas tecnologias existentes, outro sistema de busca na Internet se manter competitivo sem o uso de Inteligência Coletiva (ALEXA, 2010).

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

1.1 Objetivos

Esse trabalho tem por objetivo explicar os métodos de aplicação de Inteligência Coletiva em sistemas *web*, bem como demonstrar, através de um sistema computacional, o uso de tais conceitos, servindo, dessa maneira, como uma referência introdutiva para estudantes de graduação que desejarem pesquisar na área de desenvolvimento de sistemas de IC (Inteligência Coletiva).

Para tanto, será desenvolvido um sistema *web* com recursos de extração automática de palavras-chave a partir de um texto informado pelo usuário e recomendações de textos similares. Assim, pretende-se apresentar possibilidades de aplicação dos conceitos de IC e também explicar alguns algoritmos e suas utilidades.

1.2 Estrutura do trabalho

O presente trabalho está estruturado da seguinte forma: no Capítulo 2 é explicado o conceito de Inteligência Coletiva, bem como técnicas de implementação. No Capítulo 3 são descritos sistemas existentes que utilizam algoritmos de Inteligência Coletiva. No Capítulo 4 é descrito o desenvolvimento do sistema exemplo. Finalmente no Capítulo 5 são apresentadas as considerações finais.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

2 INTELIGÊNCIA COLETIVA

Também conhecida como Web 2.0, essa nova maneira de utilizar a Internet, que prioriza a interação dos usuários em sites e sistemas, teve repercussão em grandes meios de comunicação, como por exemplo na revista Time, a qual, em 2006, em sua tradicional edição que elege a personalidade do ano, exibiu em sua capa a palavra "You", referindo-se à possibilidade de qualquer pessoa, facilmente, gerar conteúdo e disponibilizá-lo publicamente, podendo dar-lhe visibilidade de forma abrangente, ou seja, a qualquer usuário de Internet do mundo (GROSSMAN, 2006).

Em um âmbito coletivo, essa possibilidade de geração de conteúdo proporcionou a criação e disseminação de sistemas como a Wikipédia, uma enciclopédia criada e gerenciada por usuários com o objetivo de compartilhamento de informação (WIKIPEDIA, 2011).

Não bastasse a possibilidade de criação coletiva que a Web 2.0 propõe, há também mecanismos que capacitam o sistema a se modificar conforme as ações dos usuários e o conhecimento gerado por eles, a fim de tornar a navegação mais ágil e eficiente. Ou seja, uma aplicação pode ser capaz de aprender e se adaptar com o conteúdo gerado pelas pessoas. Por exemplo, um portal de buscas pode verificar que, ao pesquisar por um determinado termo, alguns resultados nunca ou raramente são acessados. De posse dessa informação, pode então remover ou modificar a ordem do retorno de futuras buscas sobre o mesmo assunto, priorizando aqueles resultados que são mais acessados. Essa técnica é chamada de aprendizagem de máquina (ALAG, 2009).

Há ainda a possibilidade de utilizar as ações dos usuários como filtros das informações que eles estão acessando. Pode-se, portanto, analisar as interações e preferências de cada indivíduo para informá-lo sobre conteúdos que possam lhe ser relevantes (ALAG, 2009).

Essas três abordagens, geração de conteúdo em conjunto, aprendizagem de máquina e criação de filtros conforme ações dos usuários, somadas a outros fenômenos de massa presentes na Internet, como o compartilhamento de conteúdo, constituem a Inteligência Coletiva.

2.1 Antropologia

O filósofo francês Pierre Lévy foi quem, em 1994, introduziu o termo Inteligência Coletiva em seu livro "A Inteligência Coletiva: por uma antropologia do ciberespaço". Segundo ele, o termo define uma inteligência distribuída, incessantemente valorizada, coordenada em tempo real, que resulta em uma mobilização efetiva das competências.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

Dessa forma, cada indivíduo é considerado uma parte importante da inteligência. O conjunto das individualidades expressas ao mesmo tempo compõe um ambiente inteligente, mutável e, acima de tudo, evolutivo.

Conforme citado por LÉVY (1994), a Inteligência Coletiva é a possibilidade de integração das pessoas através da Internet, ou seja, a Internet através do coletivismo pode ser considerada uma nova forma de sociedade, e é através dela que as pessoas podem aprender, discutir, ensinar e tomar decisões de maneira mais rápida e eficiente.

Sua eficiência se dá pelo fato das pessoas como indivíduos possuírem um conhecimento limitado. Porém, ao serem consideradas como coletivos, o conhecimento individual somado tende a se tornar completo.

O maior exemplo que prova que a individualidade somada pode formar um conteúdo rico e consistente é a Wikipédia. A enciclopédia *online* já ultrapassa 17 milhões de artigos em mais de 270 línguas, com mais de 91 mil contribuidores ativos (WIKIPEDIA, 2011). Nela, qualquer usuário pode se cadastrar gratuitamente e adicionar, editar e/ou remover conteúdo dos artigos existentes, ou então criar novos artigos. Logicamente essa liberdade acaba permitindo que pessoas má intencionadas apaguem artigos ou criem informações falsas, exigindo do sistema uma série de regras e recursos para conter essas ações.

Outro fator que deve ser levado em consideração ao analisar-se a Wikipédia é a descentralização das decisões sobre as edições. Ou seja, não há uma pessoa que tem o poder de decidir o que pode e o que não pode ser feito, mas há grupos de pessoas que analisam os conteúdos e tomam decisões coletivamente.

Muito citada por LÉVY (1994), a descentralização do poder é um fator importante na Inteligência Coletiva. Segundo ele, somente dessa forma o indivíduo é valorizado e, portanto, incentivado à participação nos grupos, seja criando, seja tomando decisões.

2.2 Benefícios

Os benefícios da Inteligência Coletiva podem ser considerados imensuráveis, pois é difícil limitar uma tecnologia que utiliza comportamentos aleatórios de seres humanos para aprender e, consequentemente, se auto-desenvolver. Multiplicando esses comportamentos das pessoas com as formas de se desenvolver e de se pensar o sistema, é possível obter inúmeros recursos e, portanto, inúmeros benefícios diferentes.

Alguns benefícios relativos ao uso da Inteligência Coletiva são: maior fidelidade dos usuários – quanto maior a interação com o sistema, mais os usuários o utilizam; maior probabilidade do usuário encontrar conteúdo de seu interesse – quanto mais filtradas as informações, mais fácil para a pessoa encontrar o que ela realmente procura; e melhoria em

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

resultados de buscas – quanto maior o conteúdo da aplicação, mais chances possui de estar nas posições iniciais de resultados de buscas (ALAG 2009).

2.3 Formas e técnicas de aplicação

Para a aplicação de conceitos de IC, um sistema deve obrigatoriamente fornecer formas de interação para o seus usuários. Segundo ALAG (2009), há três pré-requisitos que um sistema *web* deve implementar para ser possível aplicar os conceitos de IC. São eles:

a) Permitir a interação do usuário com a aplicação e com outros usuários, aprendendo com cada interação e contribuição;
b) Agregar as informações obtidas sobre os usuários e suas contribuições utilizando-se modelos úteis;
c) Influenciar esses modelos para recomendar conteúdos relevantes ao usuário.

Portanto, a área de IC não detém-se a apenas alguns algoritmos, mas sim a um amplo conceito, através do qual pode-se desenvolver sistemas coletivamente inteligentes, utilizando-se desde equações comumente aplicadas na estatística, a redes neurais, passando por algoritmos de mineração de dados e outras técnicas, usualmente pertencentes às áreas de Inteligência Artificial e Processamento de Linguagem Natural.

No caso do recurso para realizar a previsão de ações de usuários e/ou de entradas de dados, por exemplo, as técnicas que podem ser utilizadas são do campo da Inteligência Artificial incluindo árvores de decisão, programação genética e outras técnicas de aprendizagem de máquina.

As técnicas de aprendizagem de máquina permitem que os computadores aprendam de forma autônoma. Como exemplo, uma aplicação pode receber um conjunto de dados, e a partir dessas informações, prever os dados que serão recebidos no futuro. A aplicação efetua uma análise sobre eles, verificando padrões existentes nos mesmos, para então prever, a partir dos padrões encontrados, dados que serão informados posteriormente. Se as informações recebidas como entrada não forem aleatórias, essa previsão pode ser bem próxima da realidade (SEGARAN, 2007).

A aprendizagem de máquina pode ser utilizada em mais de um tipo de recurso de IC, sendo que para cada recurso há uma ou mais técnicas específicas. Por exemplo, as árvores de decisão podem ser usadas como forma de prever saídas de dados, utilizando uma série de regras preestabelecidas. Podem ser aplicadas em diversas áreas, incluindo análise de risco financeiro, previsão de tráfego de veículos, decisões médicas e análise de negócios (SEGARAN, 2007). Nas seções seguintes serão explicadas algumas técnicas e algoritmos geralmente aplicados na área de Inteligência Coletiva.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

2.3.1 Extração de palavras-chave

Em sistemas *web* os usuários estão acostumados a acessar e buscar conteúdos através de palavras que sumarizem a informação desejada. Essas palavras podem ser palavras-chave de textos ou categorias de produtos e serviços. De posse dessa informação – palavras-chave e/ou categorias – o sistema pode identificar similaridades entre textos, produtos ou serviços (ALAG, 2009).

Uma técnica da área de processamento de linguagem natural, a extração automática de palavras-chave auxilia no processo de escolha de palavras que possam ser de fato relevantes ao texto.

A realização dessa análise pode ser feita através de alguns passos. Primeiramente, extrair somente as palavras do texto, removendo outros caracteres como pontuações e números. Em seguida, deve-se convertê-las para caixa baixa, a fim de evitar duplicações. Após as etapas de obtenção das palavras, deve-se eliminar as palavras muito frequentes, como preposições e artigos, pois estas não influem no sentido do texto. Finalmente, convertê-las a seus radicais, ou seja, às raízes das palavras, excluindo-se assim plurais e possíveis distinções entre masculino e feminino (ALAG, 2009).

A identificação das palavras em um texto pode ser feita através da criação de uma lista, removendo os elementos que não sejam compostos por letras. Nesse processo, a seguinte expressão regular pode ser utilizada: `[a-zA-Z]+`. Esta expressão obtém elementos com uma ou mais letras, maiúsculas ou minúsculas. Percebe-se que palavras unidas por hifens, como "alto-falante", serão separadas, podendo assim, perder o significado desejado pelo usuário. Para solucionar esse tipo de problema, pode-se disponibilizar um recurso para cada pessoa remover as palavras-chave equivocadamente identificadas.

A conversão das letras das palavras para caixa baixa é comumente facilitada pela linguagem de programação utilizada e não vem ao caso explicar sobre seu funcionamento.

A eliminação das palavras muito frequentes é feita inicialmente criando uma lista de palavras como artigos, preposições, conjunções e verbos de ligação, que, caso sejam encontradas, são removidas da lista de palavras-chave. Essas palavras também são conhecidas como *stopwords*.

Para efetuar a radicalização existem algoritmos prontos, também conhecidos como algoritmos de *stemming*. Segundo VIEIRA e VIRGIL (2007), o algoritmo mais completo para a língua portuguesa é o RSLP (Removedor de Sufixos da Língua Portuguesa). Proposto por Viviane Orengo, é composto por 199 regras com exceções, as quais são tratadas com base em um dicionário de 32 mil palavras na língua portuguesa. Sua implementação está disponível na biblioteca NLTK (Natural Language Toolkit), que será explicada no Capítulo 4.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

O algoritmo RSLP obtém, por exemplo, das palavras “informar”, “informe” e “informado” o radical “inform” através da remoção dos sufixos. O sistema que utilizar esse recurso deve armazenar a palavra não radicalizada para exibição ao usuário e o radical deve ser armazenado para comparações entre textos através desses termos, ou seja, textos diferentes com palavras como “informar” e “informado” podem ser considerados parecidos, dependendo das demais palavras-chave identificadas.

Existem algoritmos completos para extração automática de palavras-chave, como por exemplo o algoritmo KEA, o qual foi originalmente construído para textos da língua inglesa e foi adaptado por DIAS (2004) para a língua portuguesa utilizando a linguagem de programação Java.

O KEA, porém, não apenas identifica as palavras que não são muito frequentes nos textos, mas também as analisa lexicalmente e utiliza aprendizagem de máquina para melhorar a extração (DIAS, 2004).

2.3.2 Verificação de similaridades

Um recurso comum em sistemas que utilizam Inteligência Coletiva é o de recomendação de itens ou usuários. Com ele é possível, por exemplo, em uma loja de produtos quaisquer, recomendar produtos parecidos com os já comprados pelo usuário.

Para haver a possibilidade de se recomendar um item, deve-se medir a similaridade entre os existentes, de modo que seja possível obter o grau de semelhança entre eles. Para fazer essa verificação de similaridades, pode ser utilizada a técnica conhecida como coeficiente de correlação de Pearson, que, em termos gráficos, tem por objetivo medir a proximidade dos dados a uma linha reta e esta deve estar localizada da forma mais próxima possível de todos os pontos no gráfico (SEGARAN, 2007).

O resultado do coeficiente de correlação de Pearson é um valor entre 1 e -1, no qual 1 indica total correlação entre as variáveis, 0 significa que não há relação entre elas e -1 indica correlação inversa. Quanto mais próximos os pontos do gráfico estiverem da linha reta, maior será a correlação. Por exemplo, a partir dos pontos [1,0] e [1,0], obtém-se a correlação 1, porém, ao compararmos [1,0] e [0,1], obtém-se a correlação inversa (-1), significando que os valores entre os dois pontos tendem a aumentar inversamente, quando o primeiro valor aumenta, o segundo diminui (SEGARAN, 2007).

Antes de se fazer os cálculos de semelhanças entre itens, é preciso possuir valores representativos desses itens, os quais, no caso do presente trabalho, serão as quantidades de vezes que cada palavra aparece em cada um de uma série de textos.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

O cálculo do coeficiente é feito através da seguinte equação (MARIANO, 2008):

$$w_{a,u} = \frac{\sum_{i=1}^{m}(r_{a,i} - \bar{r}_a) * (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^{m}(r_{a,i} - \bar{r}_a)^2}\sqrt{\sum_{i=1}^{m}(r_{u,i} - \bar{r}_u)^2}}$$

Onde $r_{a,i}$ é a quantidade de palavras *a* em um texto *i* e $\bar{r}_a$ é a quantidade média de palavras *a* em todos os textos.

Portanto, o primeiro passo para a verificação de similaridades é possuir uma base de textos. Em seguida deve-se verificar as palavras relevantes de cada texto, para desconsiderar palavras de uso comum que não afetam o sentido do texto. Para isso pode-se utilizar o método de extração de palavras-chave explicado na Seção 2.3.1. Com esses dados, deve ser criada uma matriz com os textos e as palavras, informando quantas vezes cada palavra aparece no texto. A Tabela 1 exibe um exemplo dessa matriz.

Tabela 1 Matriz que informa o número de vezes que cada palavra aparece em cada texto.

	“ideia”	“casa”	“água”	“lixo”	“carro”	“combustível”
Texto 1	1	3	1	0	0	0
Texto 2	2	0	1	0	2	1
Texto 3	0	1	0	2	0	0

Com os dados da Tabela 1, é possível utilizar o coeficiente Pearson entre um novo texto e cada um dos existentes e, então, identificar aqueles que apresentam maior semelhança. Por exemplo, a seguinte frase: “o lixo gerado na nossa casa deve ser reciclado” possui as seguintes palavras-chave: “lixo”, “casa” e “reciclado”. Ao comparar essa frase com os textos existentes utilizando Pearson, obtém-se o seguinte resultado:

Tabela 2 Resultado da comparação entre textos utilizando Pearson.

	Coeficiente Pearson
Texto 1	0,2402
Texto 2	-0,8911
Texto 3	0,6794

Dessa forma, conclui-se que o Texto 3 possui o maior grau de semelhança com a frase construída, pois a correlação calculada pelo coeficiente Pearson é a mais próxima de 1, se comparado aos resultados das correlações com os outros textos.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

2.3.3 Clustering

Como forma de agrupar os resultados de comparações, a fim de separar dados por categorias, pode-se utilizar a técnica de aprendizagem de máquina denominada *clustering*. Essa técnica consiste em verificar padrões em grandes conjuntos de dados e separá-los de acordo. Com esse recurso é possível separar os textos do exemplo da Seção 2.3.2 em grupos, chamados de *clusters* (ALAG, 2009).

O *clustering* é uma técnica de aprendizagem de máquina não-supervisionada. Isso significa que seu algoritmo não é treinado com exemplos corretos e não tem por objetivo prever saídas de dados, mas apenas classificá-los (SEGARAN, 2007).

Para utilizar o *clustering* no exemplo citado na Tabela 1, devem ser calculadas as similaridades entre as linhas da matriz criada, a fim de comparar os textos baseando-se no número de vezes que as palavras aparecem em cada um. Este cálculo pode ser feito através do coeficiente de correlação de Pearson. A partir desses resultados, é possível iniciar o processo de *clustering*.

Existe mais de uma técnica de *clustering*, sendo que as duas mais populares são *k-means* e *hierarchical clustering*. No caso de estudo proposto, será utilizada a técnica *k-means*, que segundo ALAG (2009) é tão eficiente quanto a outra, porém com maior simplicidade de implementação. Além disso, segundo SEGARAN (2007), o método *hierarchical clustering* necessita de muito processamento para sua execução.

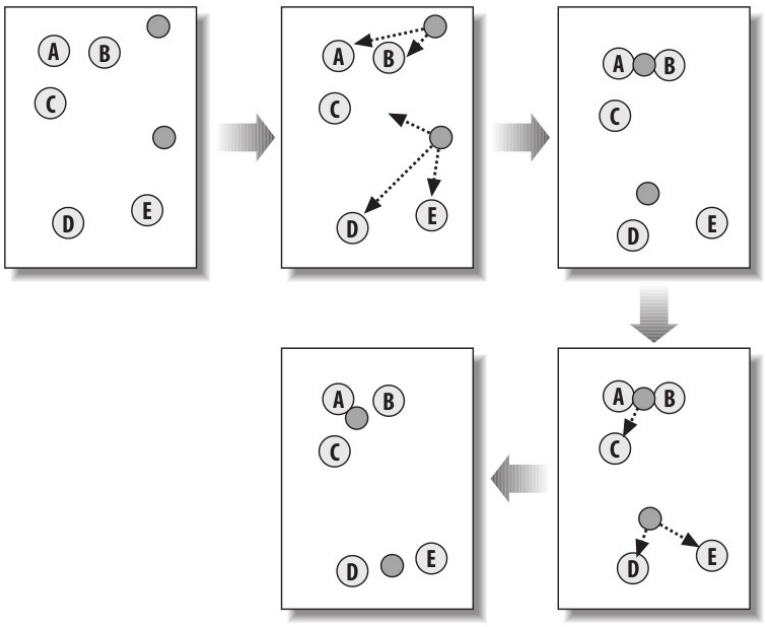

Figura 1. Exemplo de *k-means clustering* (SEGARAN, 2007).

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

O algoritmo *k-means clustering* inicialmente cria um número definido por *k* de pontos aleatórios no espaço, representando os centros dos *clusters*. Esses pontos são chamados de centroides. Então, cada texto (ou nó) é atribuído ao centroide mais próximo. Depois disso, os centroides são movidos para a posição central média entre os nós associados. A seguir a atribuição de textos aos *clusters* é refeita considerando suas novas posições. Esse processo é repetido até que as atribuições não alterem mais. O número de *clusters* deve ser testado com diferentes valores para verificar um que apresente melhores resultados.

Um exemplo pode ser observado na Figura 1. Inicialmente, o algoritmo posicionou dois *clusters* de maneira aleatória. Em seguida, verificou os dados mais próximos, o *cluster* superior encontrou os dados A e B e o inferior, os dados C, D e E. Então, os *clusters* foram reposicionados em pontos centrais aos dados, e repetiu-se o processo até não mais alterar suas posições.

Após a execução desse algoritmo, obtém-se os textos separados em grupos. Cada grupo contém um conjunto de textos semelhantes, os quais, portanto, podem ser utilizados como recomendações ao usuário.

2.3.4 Classificadores

Os algoritmos de classificação são meios de assimilar dados a suas categorias, também conhecidas como classe. Diferentemente do *clustering*, a classificação é uma técnica de aprendizagem de máquina supervisionada e, portanto, requer treinamento (MARMANIS e BABENKO, 2009).

A classificação é uma técnica para verificação de padrões, ela pode ser utilizada para classificar documentos, em técnicas reconhecimento de voz, OCR (Optical Character Recognition), classificação biológica, dentre outras. Dentre os algoritmos que realizam essa tarefa, pode-se citar *k-nearest neighbor* e *naïve* Bayes.

O classificador *k-nearest neighbor* tem por objetivo prever valores numéricos que serão informados por usuários. Por exemplo, no caso de uma aplicação de avaliação de filmes assistidos, prever a avaliação de um a cinco para um determinado filme que será dada por um determinado usuário.

A técnica *k-nearest neighbor* é realizada em dois passos, no primeiro são verificados os itens ou os usuários similares, no segundo passo é feita uma previsão baseando-se nos itens similares. A letra *k* presente no nome da técnica representa o número de itens similares que serão analisados no primeiro passo. O cálculo da similaridade pode ser feito através do coeficiente de correlação de Pearson, explicado na Seção 2.3.2 (SEGARAN, 2007).

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

O teorema de Bayes é basicamente uma equação utilizada para descobrir probabilidades inversas, ou seja, com ele é possível calcular a probabilidade de um certo conjunto de dados pertencer a um determinado padrão. O algoritmo *naïve* Bayes utiliza o teorema de Bayes para classificar de forma independente esses conjuntos de dados, ou seja, a probabilidade de um determinado conjunto de dados pertencer a uma determinada categoria independe da probabilidade de um outro conjunto de dados pertencer à mesma categoria (MARMANIS e BABENKO, 2009).

As redes neurais são técnicas de Inteligência Artificial que também são consideradas técnicas de classificação. Podem ser utilizadas na detecção de fraudes em sistemas computacionais, em filtros de mensagens indesejadas, para melhoria de sistemas de busca, dentre outras aplicações (MARMANIS e BABENKO, 2009).

De forma análoga à estrutura biológica do cérebro humano, a rede neural é constituída por nós, formando uma rede artificial dividida em camadas. Pode possuir uma camada com nós de entrada, uma camada escondida que redistribui as entradas para uma terceira camada, que por sua vez representa a de saída. O número de camadas pode ser diferente em cada técnica de redes neurais (ALAG, 2009).

Entre as técnicas existentes, pode-se citar *multilayer perceptron* e *radial basis functions*. As duas técnicas citadas podem ser utilizadas como modelos preditivos e como classificadores (ALAG, 2009).

2.3.5 Programação Genética

A programação genética é outro recurso que pode ser aplicado em IC. Também considerada uma técnica de aprendizagem de máquina, ela consiste na geração de algoritmos para resolver problemas, ou seja, um programa que cria outros programas.

Seu conceito foi baseado na evolução biológica e se dá pela criação de um grande número de programas que competem entre si para resolver uma tarefa específica. A partir dessa competição, é gerada uma lista com os algoritmos ordenados por seus desempenhos. Então os melhores programas são modificados através de mutação ou de *crossover*. A mutação consiste em criar novos programas baseados nos existentes, alterando algumas partes de maneira aleatória e *crossover* é basicamente a criação de novos programas a partir da união de partes dos melhores.

As etapas são repetidas, avaliando-se a qualidade dos algoritmos a cada iteração, chamada de geração, que termina ao atingir uma condição pré-definida, podendo ser uma solução perfeita, uma solução boa o bastante para resolver o problema, uma solução que não

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

melhorou por várias gerações ou ao atingir um número limite de gerações (SEGARAN, 2007).

Utilizada para descobrir possíveis soluções de problemas que até então não possuem solução, a programação genética pode ser aplicada, por exemplo, em pesquisas para determinar funções matemáticas ou então para resolver problemas que sofrem alterações aleatórias, como um jogo de tabuleiro (SEGARAN, 2007).

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

3 SERVIÇOS EXISTENTES

Muitos sistemas implementam recursos de IC para tornar seus sistemas melhores, fornecendo caminhos mais curtos para os usuários encontrarem as informações que procuram. Um fator importante em sistemas de IC é a possibilidade de interação, que pode ser feita através de comentários, avaliações ou através da geração de conteúdo pelos próprios usuários.

Sendo assim, como parte cotidiana da navegação na Internet hoje, os usuários utilizam sistemas que estão aprendendo com suas ações. Alguns exemplos serão explicados nessa seção.

A maior loja *online* existente, a Amazon.com (AMAZON, 2011), foi pioneira em seu sistema de recomendações de produtos. Através dele, quando muitos usuários que compraram um determinado produto, comprarem também outros determinados produtos, o sistema acaba percebendo que eles podem ser, de certa forma, semelhantes. Dessa maneira, o sistema cria listas de produtos relacionados e, quando o usuário acessar a página de um produto, é exibida a lista de itens similares ao item acessado. A Figura 2 apresenta uma pequena área da página de um produto, que é basicamente uma lista de produtos considerados similares.

Figura 2. Exibição das recomendações no site Amazon.com.

Como já citado no Capítulo 2, sistemas de busca podem utilizar a aprendizagem de máquina para aperfeiçoar seus resultados. É o caso do Google, cujos usuários de seu serviço de busca estão auxiliando o sistema a obter melhores resultados de forma transparente. Esse recurso pode ser implementado utilizando redes neurais, fazendo com que a rede aprenda a cada busca.

Em 2010 o Google estreou em seu popular serviço de *e-mail*, Gmail, um recurso que identifica os *e-mails* mais importantes. Chamado de "caixa de entrada prioritária", ele verifica as mensagens e os remetentes que são acessados pelos usuários com maior agilidade. Ou seja, aqueles *e-mails* que não são deixados para serem respondidos posteriormente.

Essas mensagens que são consideradas importantes pelo recurso são exibidas em uma seção acima da caixa de entrada. O usuário pode dizer que o recurso considerou erroneamente

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

uma mensagem como importante removendo um identificador de importância. Na Figura 3 pode ser verificado esse recurso, que também cria uma caixa ao centro com as mensagens favoritas e uma caixa ao final da página com as demais mensagens.

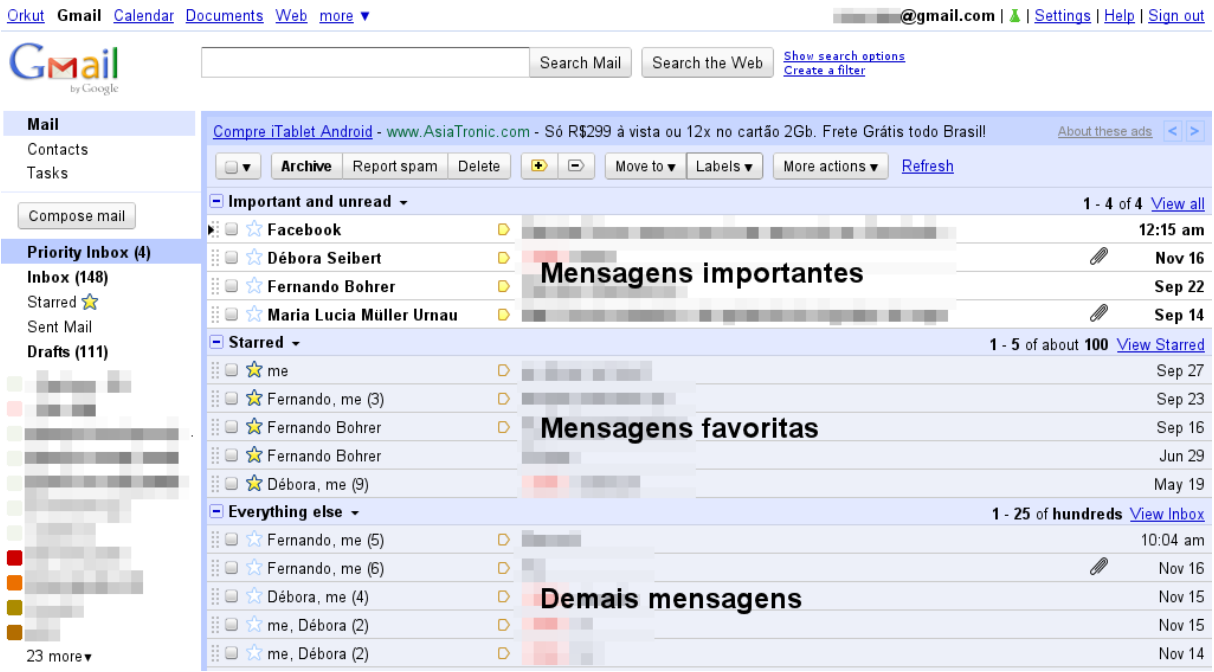

Figura 3. Imagem do da caixa de entrada prioritária do Gmail.

Outro sistema que utiliza as ações dos usuários para melhorar seus resultados é o Stack Overflow, no qual, além disso, seu próprio conteúdo é gerado pelo usuário. O sistema consiste em um banco de dados de perguntas criadas por qualquer pessoa, as quais são respondidas por outras pessoas. Tanto as respostas como as perguntas podem receber avaliações pelos usuários cadastrados. Quanto melhor avaliadas, mais ao topo das listagens as perguntas e as respostas aparecerão. Para melhorar a confiabilidade das avaliações, o usuário só tem a possibilidade de avaliar quando tiver recebido boas pontuações por suas perguntas ou respostas.

O YouTube também é um serviço que utiliza a Inteligência Coletiva, mas dessa vez como filtro de informações. Adquirido pelo Google em 2006, o sistema, que permite a publicação de vídeos, exibe listas de vídeos relacionados ao lado direito do que está sendo assistido (Figura 4). Para verificar similaridades entre vídeos, o sistema utiliza as estatísticas de acesso dos mesmos. Por exemplo, um usuário acessa um determinado vídeo, em seguida acessa outro vídeo, ao passo que várias pessoas acessam esses dois vídeos em sequência, os dois são considerados semelhantes.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

Figura 4. Imagem do YouTube.

Também mantido pelo Google, o serviço reCAPTCHA tem como finalidade a digitalização de livros que contenham palavras não identificadas por um sistema computacional, utilizando o auxílio inconsciente dos usuários.

O CAPTCHA é um recurso utilizado em formulários de sites para evitar o envio de dados por sistemas automatizados. Ele consiste na utilização de uma imagem que contenha um texto praticamente ilegível, o qual deve ser digitado em um campo. Somente no caso do texto digitado ser o mesmo da imagem, os dados são enviados, obrigando assim que somente pessoas façam o envio dos dados.

No caso do reCAPTCHA, a imagem exibe duas palavras. Uma delas é conhecida pelo sistema e é utilizada para validação, assim como é feito em um CAPTCHA comum. A outra palavra é proveniente de um livro e é exibida para utilizar-se da ação do usuário para identificá-la, já que ela não foi identificada pelo sistema de digitalização, também conhecido como OCR. O usuário deve então digitar as duas palavras, se aquela que é conhecida pelo sistema é digitada corretamente, assume-se que a outra palavra também esteja correta. Essa mesma imagem é exibida para outros usuários e a palavra mais digitada é tida como a correta. Na Figura 5 pode ser visto o recurso reCAPTCHA, com uma ilustração destacando a palavra desconhecida pelo sistema, que no caso é "*morning*" (RECAPTCHA, 2011).

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

Figura 5. Imagem ilustrativa do sistema reCAPTCHA.

Como pode ser percebido nas descrições dos serviços, a implementação de técnicas de Inteligência Coletiva pode tornar a busca por determinadas informações mais rápida e, desta forma, aumentar a fidelidade dos usuários.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

4 DESENVOLVIMENTO DO SISTEMA IDEIA

Com base na pesquisa sobre Inteligência Coletiva, na qual verificou-se os algoritmos que podem ser utilizados para sistemas que pretendem implementar recursos provenientes dessa área, foi desenvolvido um sistema que aplica alguns dos conceitos de IC.

O sistema desenvolvido consiste em um portal para geração e visualização de ideias, com extração automática de palavras-chave e recursos de recomendações de textos similares. Sua funcionalidade principal é a possibilidade de adicionar ideias, baseada no princípio da geração de conteúdo pelo usuário para disponibilização de recursos de Inteligência Coletiva, como recomendações e filtros.

O sistema foi desenvolvido através do estudo dos conceitos frequentemente utilizados em sistemas de IC, criação de um protótipo com funcionalidades básicas e desenvolvimento do sistema. Seus recursos são demonstrados nas seções seguintes. Primeiramente, é apresentada uma visão geral sobre as funcionalidades do sistema, em seguida, as ferramentas utilizadas no desenvolvimento. Então é explicado seu funcionamento técnico e ao final funcionalidades que podem ser desenvolvidas em trabalhos futuros.

4.1 Visão geral

Um panorama do funcionamento do sistema será explicado baseando-se no diagrama da Figura 6, que representa o fluxo de ações previstas pelo projeto do sistema. Destas serão explicadas aquelas que foram desenvolvidas.

No diagrama estão listados os recursos em uma representação por camadas, nas quais determinadas ações possuem sub-ações, expressando assim uma hierarquia. Ou seja, as ações da segunda camada dependem da ação da primeira camada e assim por diante. Dentre essas ações, pode ser identificado, partindo-se do recurso central “Digitar texto”, o fluxo referente à postagem de ideias, extração de palavras-chave, recomendações e demais ações relacionadas.

Inicialmente, por exemplo, o usuário pode tomar a ação de digitar um texto. Ao passo que o texto foi digitado, o sistema pode partir para três outras ações: separar as palavras-chave, efetuada automaticamente pelo sistema; postar ideia, ação que deve ser tomada pelo usuário e sugerir usuários que escreveram textos parecidos, outra ação executada pelo sistema de forma transparente ao usuário. A ação de efetuar *login* pode ser feita a qualquer momento do fluxo, porém as ações que contêm uma imagem de um cadeado necessitam de autenticação prévia para serem executadas. Esses e outros recursos são descritos na Seção 4.3, bem como o funcionamento técnico dos mesmos.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

Figura 6. Representação do fluxo de utilização do sistema.

Destas ações, aquelas marcadas por um sinal de "correto" na Figura 6 foram desenvolvidas neste trabalho. As demais ações não foram desenvolvidas e não interferem no objetivo principal do trabalho, citado na Seção 1.1, e estão documentadas na Seção 4.4 para o caso de serem implementadas em trabalhos futuros.

A base de dados utilizada pelo sistema contém, além das tabelas criadas para cadastro de usuários e controles internos do *framework* utilizado no desenvolvimento, quatro tabelas cujos nomes estão dentro do padrão do *framework*, que é explicado na Seção 4.2. O diagrama das tabelas pode ser visto na Figura 7.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

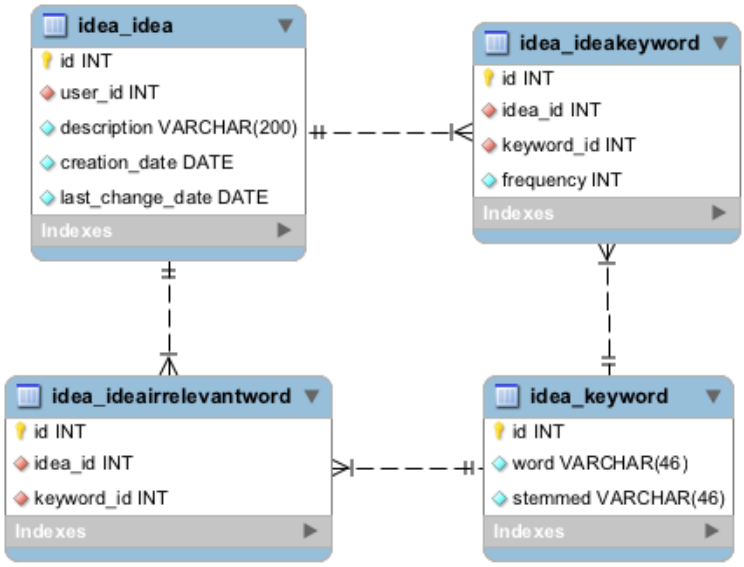

Figura 7. Diagrama das tabelas.

A tabela `idea_idea` é responsável por armazenar a ideia do usuário, bem como, o próprio usuário, a data de criação do texto e a data de alteração para o caso de haver a possibilidade de edição do texto, que não é o caso do sistema desenvolvido.

A tabela `idea_keyword` armazena todas as palavras que foram consideradas palavras-chave pelo sistema. Em `idea_ideakeyword` estão relacionadas as palavras-chave de cada texto e a frequência em que elas aparecem nos mesmos. Já a tabela `idea_ideairrelevantword` é responsável por armazenar as palavras-chave que foram consideradas irrelevantes pelo usuário em um determinado texto, havendo na mesma uma relação com a tabela de ideias e uma relação com a tabela de palavras-chave.

4.2 Ferramentas

Como o foco do presente trabalho é a aplicação dos conceitos de Inteligência Coletiva, as tecnologias foram escolhidas com a premissa de que fornecessem a possibilidade de interação transparente entre usuário e sistema.

As ferramentas utilizadas foram escolhidas devido também ao conhecimento prévio nas tecnologias. Como linguagem de programação foi utilizada a linguagem Python e como *framework* de desenvolvimento *web*, o Django, o qual facilita o desenvolvimento pois já implementa recursos básicos de sistemas *web*, como *templates*, módulo administrativo, gerenciamento de usuários e controles de segurança (SANTANA e GALESI, 2010).

Cada projeto criado utilizando Django contém uma série de aplicações, que podem ser diferentes *websites* que se intercomunicam. No caso do sistema desenvolvido por esse

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

trabalho, foi criado um projeto e dentro dele a aplicação "*idea*" que é, basicamente, o sistema Ideia. Dessa forma, é possível utilizar "*idea*" em outros projetos desenvolvidos com o *framework* Django.

Como sistema de banco de dados, foi utilizado o sistema PostgreSQL. As tabelas foram criadas através do Django, que gera os comandos SQL a partir de classes Python chamadas de modelos. A nomenclatura das tabelas, portanto, segue o padrão do *framework*: `aplicação_modelo`.

Para a implementação de recursos que executam no lado do cliente, ou seja, no navegador, foi utilizada a biblioteca da linguagem JavaScript, Dojo Toolkit. Sua escolha se dá pela sua vasta gama de recursos, incluindo uma tecnologia importante para sistemas de IC, o AJAX (DOJO, 2011).

Para a utilização do algoritmo RSLP, citado na Seção 2.3.1, não foi realizada sua implementação, pois este é disponibilizado pela biblioteca de código aberto NLTK. Essa biblioteca consiste em uma série de ferramentas para processamento de linguagem natural e foi desenvolvida em Python, sendo facilmente utilizada pelo sistema, bastando incluí-la ao código (BIRD, KLEIN e LOPER, 2009).

4.3 Funcionalidades

O sistema conta com uma tela principal simples (Figura 8), com um campo no qual pode ser inserido um texto; uma área logo abaixo deste campo onde o sistema exibe as palavras-chave, que também podem ser adicionadas manualmente pelos usuários; um botão para limpar os dados do formulário, que apaga além do texto digitado, as palavras-chave do texto e um botão enviar, que executa a ação de salvar a ideia na base de dados.

Em todas as telas há uma barra superior com *links* para a página principal, autenticação e criação de cadastro, com exceção do *link* para a página atual. No caso do usuário estar autenticado, a barra superior conta com apenas um *link* para sair, já que os recursos principais do sistema são acessíveis através de uma única tela, e uma mensagem de boas-vindas.

A aparência do sistema e sua interface foi desenvolvida para que a atenção do usuário se voltasse à digitação do texto, com um campo central, que ao passo que a página é carregada, recebe o foco do cursor para iniciar a digitação.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

Figura 8. Tela principal do sistema.

A tela de criação de cadastro utiliza um modelo padrão de registro de usuário do *framework* Django, sendo gerado automaticamente pelo mesmo. No formulário constam os campos de usuário, senha, primeiro nome, último nome (em tradução automática e literal, feita pelo Django, de "*last name*") e endereço de e-mail (Figura 9).

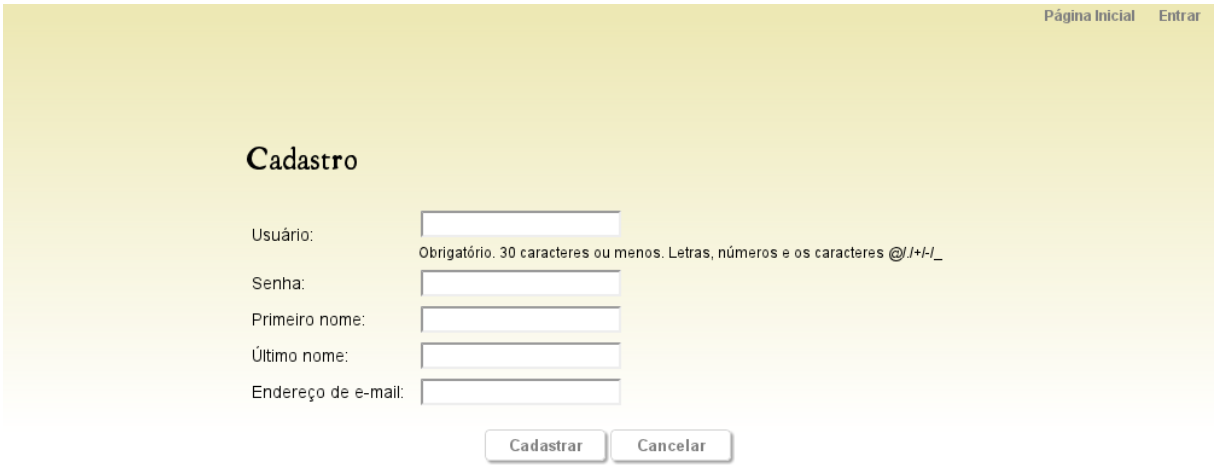

Figura 9. Tela de criação de cadastro.

A tela de autenticação conta com os campos de usuário e senha, além de um *link* para a criação de novo registro no sistema. Essa tela pode ser visualizada na Figura 10.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

Figura 10. Tela de autenticação.

Dado o início do preenchimento do texto pelo usuário, o sistema aguarda o pressionamento de uma tecla que não seja uma letra, para efetuar a ação de extração de palavras-chave. Assim é evitado que palavras sejam identificadas pela metade, o que aconteceria caso a extração acontecesse a cada intervalo de tempo pré-definido.

Portanto, a cada palavra digitada, é executada uma requisição AJAX para obter as palavras, dentre as já digitadas, que são consideradas relevantes. O primeiro passo dessa verificação é obter apenas as palavras do texto, excluindo-se números e sinais gráficos, as quais tenham como tamanho entre três e quarenta e seis letras, que segundo PRIBERAM (2003) é o tamanho da maior palavra da língua portuguesa presente em um dicionário. Em seguida, para cada palavra é executada a função `checkwordisuseful` que informará se ela é considerada relevante.

Essa função primeiramente verifica se a palavra está presente na lista de *stopwords*, em caso positivo, é considerada irrelevante. As *stopwords* são palavras que não possuem influência no sentido do texto e no sistema desenvolvido são representadas por duas listas. A primeira foi construída no desenvolvimento do sistema e contém artigos, preposições, conjunções e pronomes. A segunda lista é obtida através biblioteca NLTK, a qual contém uma lista com 203 palavras que, pela análise da biblioteca, costumam aparecer em altas frequências em textos de língua portuguesa (BIRD, KLEIN e LOPER, 2009). Caso ela não esteja em nenhuma dessas listas, a rotina prossegue efetuando uma consulta na base de dados para verificar, através do radical da palavra em questão, se ela foi mais vezes usada como palavra-chave ou se foi mais vezes removida pelo usuário. Essa verificação é possível, pois em cada palavra que o sistema elenca como relevante e exibe na tela, existe um *link* para remoção da mesma. Quando é removida, o sistema armazena essa informação na sessão, para quando a ideia for salva, armazenar na tabela `idea_ideairrelevantword`. Caso a

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

palavra nunca tenha sido utilizada como palavra-chave, a função `checkwordisuseful` retorna falso como resultado. A radicalização é feita através da biblioteca NLTK que possui uma implementação do algoritmo RSLP, citado na Seção 2.3.1.

As palavras-chave são exibidas abaixo do campo de digitação (Figura 11). Como já citado, caso o usuário não confirme a relevância das palavras identificadas como importantes, ele pode removê-las. Também é possível adicionar outras que não foram sugeridas ou digitadas, através do botão “Adicionar”, que exibe um campo ao seu lado esquerdo para o usuário digitar a palavra (Figura 11). Ao passo que esse campo perde o foco ou que o usuário aperte a tecla “Enter”, a informação é armazenada para ser salva quando a ideia for enviada. As palavras adicionadas pelo usuário podem ser removidas, porém sua remoção não implica nas futuras extrações automáticas, pois ela não é adicionada à lista de termos removidos.

Figura 11. Campo para adicionar palavra-chave é exibido à esquerda do botão "Adicionar".

Ao mesmo tempo que é feita a extração automática das palavras-chave, a partir da identificação de pelo menos uma palavra, o sistema verifica se existem ideias parecidas cadastradas, através da função `getsimilarideasbydistance`, que recebe por parâmetro uma lista de termos, que são as palavras-chave do texto digitado, e um limite máximo de textos similares a serem obtidos, que possui valor padrão cinco. O sistema radicaliza as palavras informadas e busca na base de dados as ideias que contenham ao menos um desses radicais. Nessa consulta também são obtidas suas palavras-chave, o radical de cada uma e a frequência em que cada uma delas aparece em cada texto. Em seguida os textos são

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

organizados em uma matriz semelhante à Tabela 1, porém acrescentando-se às colunas os radicais das palavras, para possibilitar a verificação de semelhanças entre textos que contenham a palavra-chave "exemplo" com textos que contenham a palavra-chave "exemplar", por exemplo. A frequência dos radicais é dada pela maior frequência dentre as palavras que possuam o radical em questão, em cada texto. Finalmente são comparadas todas as ideias encontradas com a lista de palavras-chave do texto digitado, obtendo-se o coeficiente de correlação Pearson para cada comparação. Os resultados são então ordenados, deixando por primeiro os textos que apresentam um maior valor de correlação, ou seja, que possuam maior similaridade. A implementação da correlação Pearson em Python foi obtida de SEGARAN (2007).

Caso sejam encontradas ideias similares, o usuário pode visualizar um trecho inicial de cada uma delas em uma lista à direita do campo de texto (Figura 12). Nessa mesma lista é possível visualizar à esquerda do trecho um percentual indicando o nível de semelhança com o texto sendo escrito. Esse percentual é obtido através da alteração do resultado do cálculo de correlação Pearson para um valor entre 0 e 100. Como a correlação pode ser de -1 a 1, foi somada a ela o valor 1, o resultado, que é um valor entre 0 e 2, foi então dividido por 2 e finalmente multiplicado por 100.

Figura 12. Representação das funcionalidades de extração de palavras-chave e recomendações.

No canto inferior direito de cada ideia presente na lista é exibido o primeiro nome do autor (Figura 12). Cada uma das ideias é um *link* que, ao ser acessado, exibe o texto completo

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

e suas palavras-chave, bem como o nome completo do usuário criador, na mesma página e na mesma área da listagem. Para voltar à lista há o *link* "Voltar" logo abaixo do nome do autor. Por se tratar de uma requisição AJAX que altera apenas os dados da direita da página, o texto escrito pelo usuário no campo principal e as palavras-chave de seu texto permanecem inalterados (Figura 13).

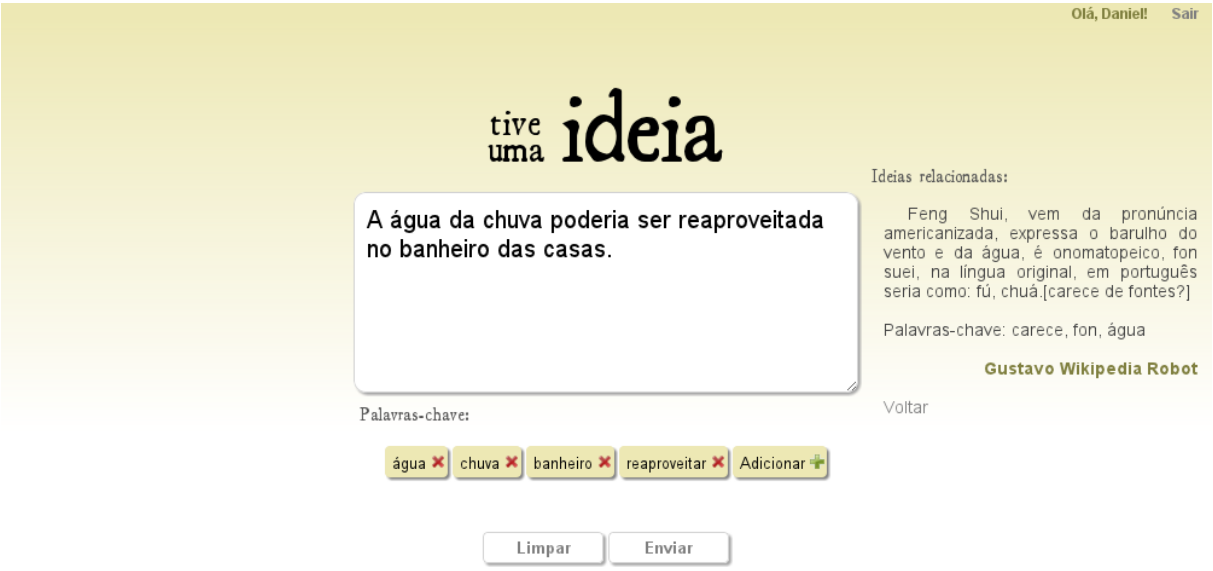

Figura 13. Visualização da ideia completa à direita do campo de digitação.

A busca por textos similares foi inicialmente desenvolvida utilizando a técnica de *clustering k-means*, a qual, ao contrário da técnica criada, obtinha todas as ideias e as separava em grupos. Pelos testes realizados, ao utilizar essa técnica em aproximadamente 800 textos, com 5 mil palavras-chave, o tempo necessário era de 2 horas. Foi tentado otimizar o código utilizando a biblioteca de otimização *psycho* e as estruturas de dados otimizadas da biblioteca *collections*, porém sem sucesso, dando uma diferença de tempo de apenas alguns minutos. Como a busca por ideias similares precisa ser realizada em tempo real, enquanto o usuário digita seu texto, o método de *clustering* se tornou inviável e, inclusive, percebeu-se que realizava trabalho desnecessário para o recurso em questão, pois gerava grupos para todas as ideias do sistema e não somente para aquelas que poderiam ter relação com o texto, como é o caso da solução encontrada.

Ao passo que o usuário decida cadastrar a ideia, esta ação pode ser tomada através do botão "Enviar" (Figura 12). Ao ativá-lo, o sistema verifica se algum texto foi digitado e se pelo menos duas palavras-chave foram adicionadas. Caso contrário, é exibida uma mensagem

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

informativa e a ideia não é salva. No caso do usuário não estar autenticado no sistema, ele será redirecionado à página de autenticação (Figura 10).

Se o usuário estiver autenticado, tiver digitado algum texto e haja pelo menos duas palavras-chave definidas, a ideia é salva no sistema, exibindo uma mensagem no topo da tela (Figura 14). A ação de salvar também armazena na base de dados cada palavra-chave e as palavras que foram removidas manualmente pela pessoa.

Figura 14. Mensagem no topo da tela informando que a ideia foi salva.

Na próxima seção são apresentadas as funcionalidades que podem ser desenvolvidas em trabalhos futuros para melhoria do sistema.

4.4 Trabalhos futuros

Sistemas com produção de conteúdo pelo usuários e que se baseiam na interação fornecem possibilidades de desenvolvimento de funcionalidades diversas, podendo até desviar o foco do desenvolvimento, fazendo com o que o sistema não contemple a proposta inicial de desenvolvimento. Visando manter-se no escopo do projeto, algumas funcionalidades poderiam contribuir com a melhoria do sistema, aumentando a interação pelos usuários e utilizando mais conceitos de IC. Para explicá-las, são utilizadas como base algumas telas criadas no projeto do sistema, que não foram implementadas na sua totalidade.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

Para aumentar a interatividade do sistema, poderia ser implementado um recurso de colaboração entre usuários para a criação coletiva de ideias. Para isso seria interessante o usuário definir ao publicar sua ideia, se ela pode ou não ser editada por outras pessoas, conforme tela do protótipo na Figura 15. No caso da edição ser permitida, para evitar que a ideia perca o contexto desejado, as edições devem ser aprovadas pelo autor para serem incorporadas ao texto.

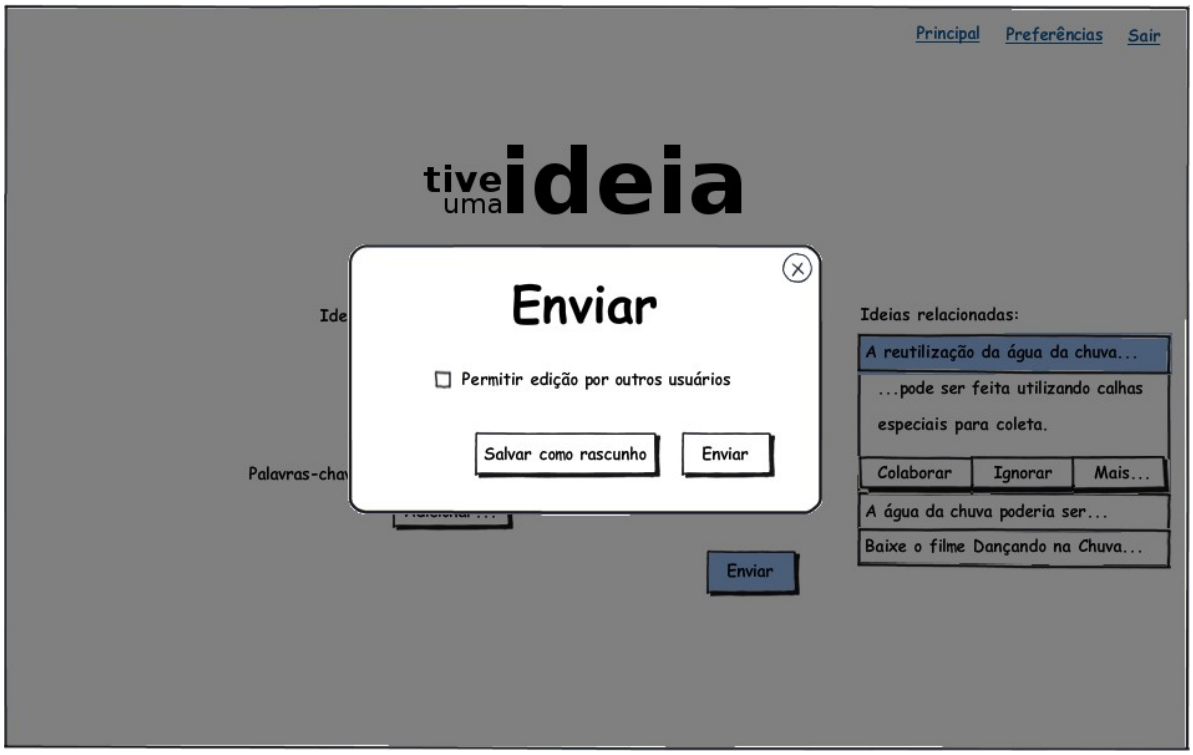

Figura 15: Opções de envio de uma ideia.

Outra opção que aparece na janela de postagem da Figura 15 é "Salvar como rascunho". Essa opção salvaria a ideia, porém não a publicaria. O texto salvo poderia ser visto e editado na tela principal no campo de criação de ideia, portanto o usuário poderia salvar apenas uma ideia por vez. Na edição do texto salvo as opções seriam as mesmas presentes na criação de um novo texto.

Como forma de melhorar a busca por textos semelhantes, poderia ser fornecida uma ação em cada ideia da lista de ideias similares para o usuário informar se ela possui de fato relação com a que está sendo escrita (ação "Ignorar" da Figura 16). Ao passo que o usuário execute essa ação, a informação deve ser armazenada na base de dados e posteriormente utilizada de forma similar ao que foi feito na extração automática de palavras-chave, que permite remover as palavras identificadas como relevantes pelo sistema.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

A Figura 16 foi criada como protótipo da tela de exibição de ideias similares e apresenta ainda outras opções que não foram desenvolvidas. As opções forneceriam ainda ao usuário as possibilidades de colaborar com a ideia similar, convidar o autor a participar do desenvolvimento de sua ideia, enviar mensagem ao usuário ou sinalizar a ideia como *spam* (lado inferior direito da Figura 16).

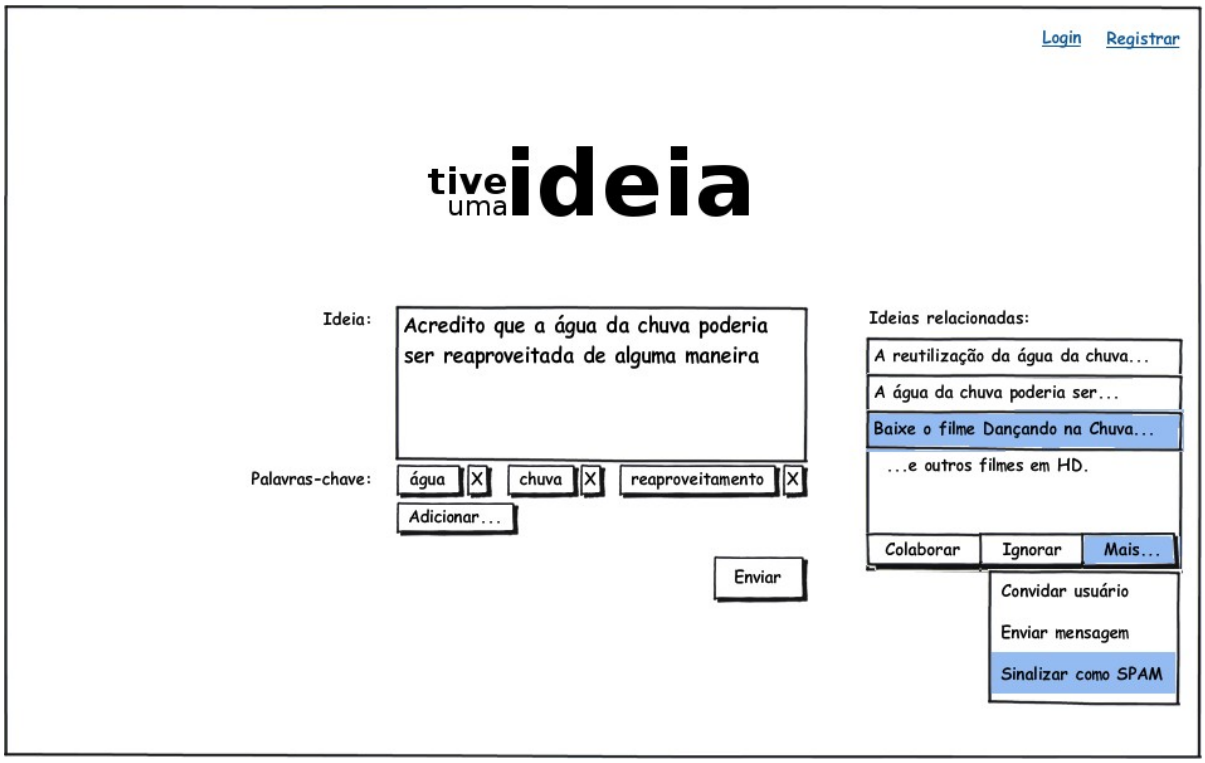

Figura 16. Representação das funcionalidades na lista de recomendações.

A opção colaborar redirecionaria o usuário para uma página com as informações da ideia em questão (Figura 17). Nessa página, caso o autor tivesse permitido a edição, o usuário poderia editar o texto e as palavras-chave através dos *links* "Editar" acima do campo da ideia e do campo de palavras-chave, de maneira similar a sistemas como a Wikipédia, porém neste caso seria necessária a autorização do autor para que a edição fosse incorporada.

Nessa mesma tela haveria uma área abaixo da lista de palavras-chave, onde usuários cadastrados poderiam adicionar comentários em qualquer ideia, mesmo ela não sendo editável pelos outros. Esse tipo de recurso é comum em *sites* cujos conteúdos são criados pelos usuários, como o YouTube, o Flickr e *blogs*, e tem sido utilizado inclusive por *sites* de notícias, fornecendo um espaço para as críticas dos leitores.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

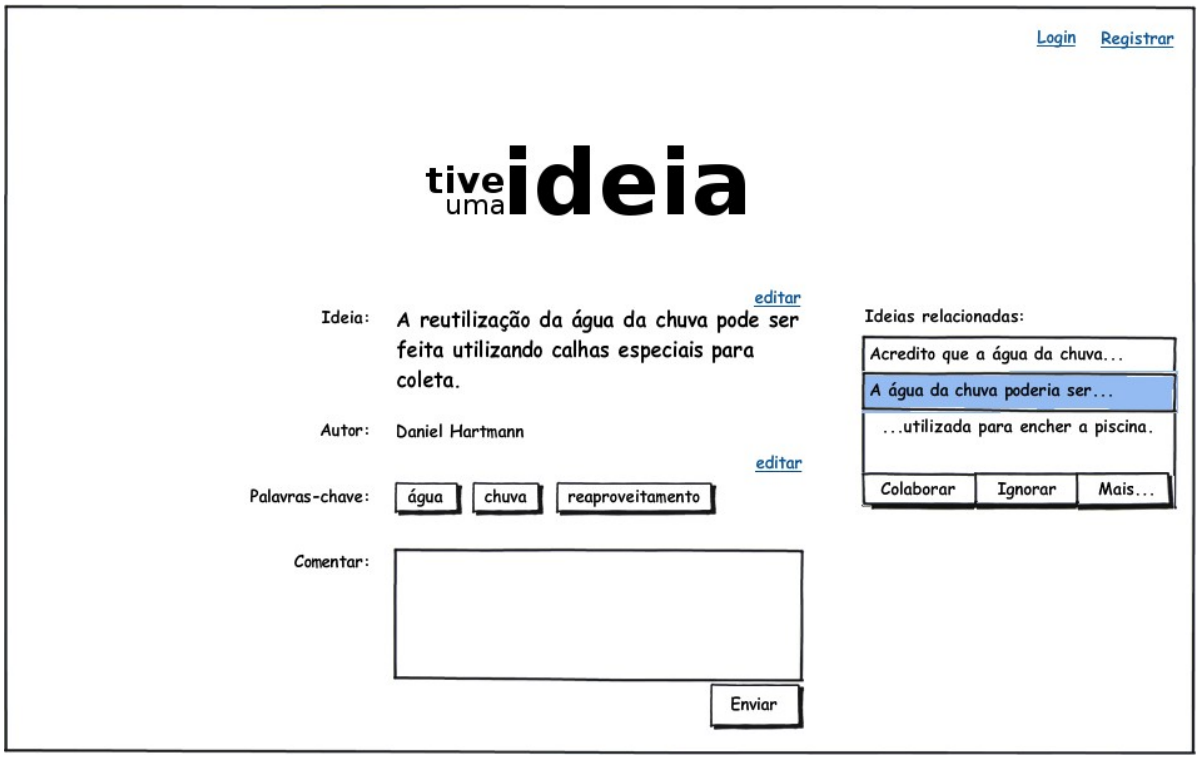

Figura 17. Tela principal de colaboração.

Com a implementação dos recursos citados nesta seção, o sistema utilizaria mais conceitos de coletivismo, o que, conforme visto na Capítulo 2, tenderia à produção de ideias mais completas e, possivelmente, a uma expansão do número de usuários do *site*, pelo fato de permitir maior interação entre eles.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

5 CONSIDERAÇÕES FINAIS

Como pode ser observado no Capítulo 3, a utilização de recursos de IC já se tornou comum em sistemas *web* populares, fazendo com que conceitos dessa área tornem-se praticamente obrigatórios para sistemas modernos.

A constante atualização e a concorrência entre os navegadores existentes também auxilia e incentiva a criação de sistemas mais interativos, pois facilita a implementação de recursos existentes e fomenta a criação de novos recursos.

Além do avanço tecnológico, pode-se citar que a Internet está cada vez mais convergindo ao coletivismo, pois as aplicações conhecidas como redes sociais estão sendo cada vez mais utilizadas, tornando-se potencialmente interessantes de explorá-las através de conceitos de IC, por haver uma grande quantidade de conteúdo sendo gerado diariamente.

Outro fator importante que deve ser citado é que grandes empresas como o Google estão continuamente lançando novos recursos que envolvem IC, como foi descrito no Capítulo 3, o que leva a conclusão de que a área ainda pode ser muito explorada, tendo potencial para gerar uma série de recursos inéditos.

Este trabalho procurou enriquecer as pesquisas sobre IC, visto que a escassez de material nessa área é grande. Dessa maneira, pode ser utilizado como referência inicial para desenvolvimento de sistemas de Inteligência Coletiva.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)

REFERÊNCIAS

ALAG, S. **Collective Intelligence in Action.** 2009. 1 ed. Greenwich: Ed. Manning, 2009. 397 p.

ALEXA. The Web Information Company. **Top Sites.** 2011. Disponível em: <http://www.alexa.com/topsites>. Acesso em: 12 jun. 2011.

AMAZON. Amazon.com: Online Shopping for Electronics, Apparel, Computers, Books, DVDs and more. 2011. Disponível em: <http://www.amazon.com/>. Acesso em: 12 jun. 2011.

BIRD, S; KLEIN, E.; LOPER, E. **Natural Language Processing with Python.** 2009. 1 ed. Sebastopol: Ed. O'Reilly Media, 2009. 479 p.

DIAS, M. A. L. **Extração Automática de Palavras-chave na Língua Portuguesa Aplicada a Dissertações e Testes da Área das Engenharias** . 2004. 127 p. Dissertação (Mestrado em Engenharia Elétrica) – Faculdade de Engenharia Elétrica e de Computação (FEEC) , UNICAMP, Campinas, 2004.

DOJO. The Dojo Toolkit. **Features.** 2011. Disponível em: <http://dojotoolkit.org/features/>. Acesso em: 12 jun. 2011.

GROSSMAN, L. TIME. **Time's Person of the Year: You**. 2006. 1 p. Disponível em: <http://www.time.com/time/magazine/article/0,9171,1569514,00.html>. Acesso em: 19 jun. 2011.

LÉVY, P. **A Inteligência Coletiva: por uma Antropologia no Ciberespaço**. 1994. 6 ed. São Paulo: Ed. Edições Loyola, 2010. 212 p.

MARIANO, R. G. **Desenvolvimento de uma família de sistemas de recomendações baseados na tecnologia da Web Semântica e seu reuso na recomendação de instrumentos jurídico-tributários.** 2008. 152 p. Dissertação (Mestrado em engenharia) – Curso de Pós-Graduação em Engenharia de Eletricidade, Universidade Federal do Maranhão, São Luís, 2008.

MARMANIS, H; BABENKO, D. **Algorithms of the Intelligent Web.** 2009. 1 ed. Greenwich: Ed. Manning, 2009. 345p.

PRIBERAM. Priberam - Dúvidas linguísticas. **Maior palavra da língua portuguesa.** 2003. 1 p. Disponível em: <http://www.flip.pt/Duvidas-Linguisticas/Duvida-Linguistica.aspx?DID=485>. Acesso em: 19 jun. 2011.

RECAPTCHA. Digitizing Books One Word at a Time.**What is reCAPTCHA.** 2011. Disponível em: <http://www.google.com/recaptcha/learnmore>. Acesso em: 04 abr. 2011.

SANTANA, O.; GALESI, T. **Python e Django: Desenvolvimento ágil de aplicações web.** 1 ed. São Paulo: Novatec Editora, 2010. 279 p.

SEGARAN, T. **Programming Collective Intelligence**. 1 ed. Sebastopol: Ed. O'Reilly Media, 2007. 307 p.

VIERA, A.F.G.; VIRGIL, J. **Uma revisão dos algoritmos de radicalização em língua portuguesa**. Information Research, 12(3) artigo 315. Disponível em: <http://informationr.net/ir/12-3/paper315.html>. Acesso em: 31 out. 2010.

WIKIPEDIA. The Free Encyclopedia. **Wikipedia: About.** 2011. Disponível em: <http://en.wikipedia.org/wiki/Wikipedia:About>. Acesso em: 14 jun. 2011.

BDU – Biblioteca Digital da UNIVATES (http://www.univates.br/bdu)