DAS Departamento de Automação e Sistemas
CTC Centro Tecnológico
UFSC Universidade Federal de Santa Catarina

Desenvolvimento de software para classificação automática de palavras-chave

Relatório submetido à Universidade Federal de Santa Catarina como requisito para a aprovação da disciplina:

DAS 5501: Estágio em Controle e Automação Industrial

Paulo Luis Franchini Casaretto

Florianópolis, agosto de 2012

Desenvolvimento de software para classificação automática de palavras-chave

Paulo Luis Franchini Casaretto

Orientadores:

Bruno Cavaler Ghisi da Resultados Digitais, Bacharel em Ciências da Computação

_____

Assinatura do Orientador

Prof. *Jomi Fred Hubner da UFSC*

_____

Assinatura do Orientador

Este relatório foi julgado no contexto da disciplina
DAS 5511: Projeto de Fim de Curso
e aprovado na sua forma final pelo
Curso de Engenharia de Controle e Automação

Agradecimentos

Resumo

O marketing tradicional, que antes era feito através de televisão, rádio, jornais, revistas, outdoors e panfletos já não é mais tão eficaz. As pessoas tem cada vez mais maneiras de ignorar as interrupções, maneira que classicamente faz-se propaganda. Entretanto, a Internet tem mudado radicalmente a forma com que uma empresa pode alcançar seus clientes.

A Internet é o meio onde ideias objetivas aliadas a uma boa execução têm muito mais valor do que exclusivamente investimentos financeiros. O Marketing Digital permite que tenhamos alto retorno sobre o investimento e também facilmente a mensuração desses resultados. Para as Pequenas e Médias Empresas (PMEs), os benefícios do Marketing Digital são ainda mais evidentes, já que em geral elas dispõem de poucos recursos para promover os seus produtos e serviços.

No entanto, muito empreendedores e responsáveis de marketing em empresas ainda não tiram proveito desse potencial, ou não utilizam isso de forma correta, seja por desconhecerem as técnicas e ferramentas ou mesmo por terem receio da complexidade do meio. Com o objetivo de simplificar e integrar as ações em Marketing Digital a Resultados Digitais [18] oferece o software conhecido como RDStation [19].

Um dos maiores desafio do Marketing Digital é atrair mais visitantes – que podem se tornar potenciais clientes – para o *website* da empresa. Neste ponto se destacam ferramentas de busca como Google e Yahoo. Um dos problemas que o RDStation se propõe a resolver é como conseguir melhores resultados provenientes dessas ferramentas de busca. O software hoje fornece muitas informações úteis para o usuário, no entanto, cabe ao próprio usuário decidir quais as estratégias a serem adotadas.

Para que o software entregue mais valor para o usuário objetivou-se que este deve também indicar quais são as estratégias que possuem um maior potencial. A partir de uma série de entradas relativas a situação atual do cliente, o software deve ser capaz de dizer qual a melhor a ser seguida para extrair o melhor resultado das ferramentas de busca.

Para isso, foi proposta uma extensão do software baseada em técnicas de Inteligência Artificial. A proposta foi então adequada a realidade da empresa e uma implementação inicial com o objetivo de teste foi realizada. Os testes não foram conclusivos devido ao pouco tempo para colher resultados estatisticamente mais expressivos. Contudo, os resultados empíricos são promissores.

Abstract

Marketing done through traditional means like TV, newspapers, magazines, billboards, and leaflets is no longer as effective. People are getting better and better at ignoring interruptions, which is the way traditional marketing is done. The Internet has radically changed the way a company can reach its customers.

The Internet is a medium where objective ideas coupled with good execution are far more valuable than purely financial investments. Digital Marketing allows us to have high return on investment and is also easy to measure those results. For Small and Medium Enterprises, the benefits of Digital Marketing are even more evident, since they usually have few resources to promote their products and services.

However, entrepreneurs and marketing staffdo not take advantage of this potential, or do not use it correctly, either through ignorance of techniques and tools or even for fear of the complexity of the environment. In order to simplify and integrate the actions in Digital Marketing, Resultados Digitais[18] provides the software known as RDStation [19].

A major challenge of Digital Marketing is to attract more visitors - who might be potential customers - to the company's website. This is where search engines, like Google and Yahoo, really shine. One of the problems that RDStation proposes to solve is how to get better results from these search engines. Currently, the software provides much useful information to the user, however, it is up to users themselves decide which strategies to be adopted.

For the software to deliver more value to the user it has been defined that it should also indicate what are the strategies that have more potential. Based on data for the current situation of the client, the software must be able to tell what is the best strategy for the customer to follow in order to get the best result of the search engines.

For this, it has been proposed an extension of the software based on artificial intelligence techniques. The proposal was then brought to the reality of the company and an initial algorithm, together with initial tests, was implemented. The tests were

inconclusive due to little time to gather more statistically significant results. However, the empirical results are promising.

Sumário

Simbologia

SaaS	*Software as a Service*
FB	Ferramenta de busca
PPC	*Pay per click*
HTML	*HyperText Markup Language*
HTTP	*HyperText Transfer Protocol*
JSON	*Javascript Object Notation*

Capítulo 1: Introdução

1.1: Objetivos e Motivação

Há alguns anos, a propaganda tradicional feita através de jornais, revistas, rádio televisão e mala direta era as única maneira de se fazer marketing. Com essas mídias é difícil alcançar compradores de um determinado segmento através de mensagens personalizadas.

O problema com o marketing tradicional é que ele não vem acompanhando as mudanças de hábitos da população, assim como as vendas locais mudaram com o surgimento do e-commerce e mídias de massa. O modelo de marketing de interrupção foi ultrapassado e não faz sentido na Internet, sendo cada vez mais caro e menos efetivo fazer marketing apenas com compra de propaganda. Na Internet, é muito mais difícil para uma marca interromper alguém e chamar a sua atenção. Ao invés de interrupção de mão única, o marketing digital entrega conteúdo útil no momento que um comprador precisa dele [2].

O Marketing Digital no entanto apresenta novas dificuldades. O modo com que as pessoas encontram e consomem informação e produtos é muito diverso. Por se tratarem de tecnologias muito recentes e heterogêneas executar uma estratégia de Marketing Digital completa e efetiva é complexa, e atualmente requer o uso de muitas ferramentas distintas e que não têm integração entre si.

Uma parte muito importante do Marketing Digital é a relação do website com ferramentas de busca. De acordo com entrevista feita em 2007 no mercado norte americano [3]:

- 74% dos participantes usaram ferramentas de busca para achar informações sobre negócios locais, em contra partida, 65% usaram páginas amarelas, 50% usaram páginas amarelas na internet, e 44% usaram jornais tradicionais;
- 86% dos entrevistados disseram que já usaram a internet para achar um negócio local, um aumento de 70% do ano anterior;
- 80% reportaram procurar por um produto ou serviço online e em seguida realizaram a compra *offline* localmente;

Ferramentas de busca na internet como Google e Bing são muito poderosas para a obtenção de novos visitantes para qualquer site. Contudo, o decaimento de número de visitas em relação a posição nas buscas é de ordem exponencial [17]. Logo, ter uma boa posição nas buscas orgânicas (não pagas oriundas naturalmente de mecanismos de busca) pode ser uma estratégia que permitirá ter mais visitas ao *website* e portanto mais oportunidades.

Um dos problemas que o *software* RDStation se propõe a resolver é como conseguir melhores resultados provenientes dessas ferramentas de busca. O *software* hoje fornece muitas informações úteis para o usuário, no entanto, cabe ao próprio decidir quais as estratégias a serem adotas. Para que o software entregue mais valor foi definido que este deve também indicar quais são as estratégias que tem mais potencial. A partir de uma série de entradas relativas a situação atual do cliente, o software deverá ser capaz de dizer qual a melhor estratégia a ser seguida para extrair o melhor resultado das ferramentas de pesquisa.

Neste contexto permitiu-se iniciar um projeto de uma extensão do software que pudesse, a partir das informações das palavras-chave, recomendar ao cliente qual a estratégia mais indicada para cada caso, ou seja, qual técnica trará os melhores resultados.

A partir dessa premissa, este trabalho tem por objetivo:

1. Propor uma técnica capaz de recomendar estratégias específicas a partir de dados relacionados;
2. Verificar a viabilidade de implementação da proposta;
3. Realizar testes para confirmar a validade da proposta;

1.2: Contextualização no curso

Para a realização deste trabalho, foram utilizados conhecimentos adquiridos em diversas disciplinas oferecidas pelo curso de Engenharia de Controle e Automação, sendo a maior parte delas da área de computação.

Os conhecimentos na área de Inteligência Artifical foram essenciais para o desenvolvimento do projeto, visto que as soluções propostas surgiram desta área do conhecimento. Contribuíram também as disciplinas de Fundamentos de Estrutura da

Informação e Metodologia para Desenvolvimento de Sistemas dada a forte natureza informática do problema.

Também foram importantes as disciplinas de Banco de Dados, Fundamentos da Estrutura da informação e Metodologia para desenvolvimento de sistemas.

1.3: Estrutura do Relatório

Este relatório está dividido em 7 capítulos.

No segundo capítulo são abordadas questões fundamentais sobre mecanismos de busca, seu funcionamento e a importância de um bom posicionamento nos resultados para as empresas com presença na internet.

No capítulo seguinte é explanado o funcionamento do software chamado RDStation. Neste capítulo o problema é contextualizado dentro do *workflow* do *software*.

O capítulo quarto expõe a visão da solução ideal, ou seja a forma com que idealmente o problema seria resolvido.

O quinto capítulo trata da implementação da solução proposta no capítulo anterior , dos problemas encontrados e como estes foram resolvidos.

Os resultados obtidos são discutidos e analisados no sexto capítulo.

O último capítulo refere-se as conclusões do trabalho e apresenta as perspectivas futuras de extensão do mesmo.

Capítulo 2: Mecanismos de Busca

O uso de ferramentas de busca já permeia a sociedade global. A maneira com que as pessoas trabalham, se divertem, compram, pesquisam e se interagem entre si mudou para sempre. Organizações de todo o tipo, e também indivíduos precisam ter uma presença na Web — e precisam que as ferramentas de busca as tragam tráfego [7].

Procurar informações utilizando mecanismos de busca tem se tornado parte de nossas vidas. A maioria dessas ferramentas responde a uma requisição com uma lista ordenada de páginas. Essa ordem reflete a relevância estimada das páginas em relação a pesquisa.

Utilizadores (pessoas) de mecanismos de busca inserem palavras-chave, e avaliam os resultados retornados, fazendo uma decisão sobre se selecionam ou não um dos resultados retornados ou reformulam a consulta. Mecanismos de busca atuam como intermediários de informação que facilitam o processo de busca de informações.

Mecanismos de busca são uma ótima maneira de se obter novos visitantes para um site, que dependendo da área de atuação, podem se converter em negócios ou em futuras oportunidades. No entanto, não basta estar dentro do conjunto de resultados. Devido ao decaimento exponencial do número de visitas em relação a posição nas pesquisas, é preciso estar bem posicionado para realmente aproveitar os benefícios. De acordo com B. Pan et Al. (2007) [5] os usuários tendem a confiar mais na ordem dos resultados do Google do que no resumo da página mostrada.

De acordo com E. Eric (2009) [6]

- 62% dos usuários de mecanismos de busca clicam em um resultados dentro da primeira página, e 90% dentro das primeiras três páginas;
- 41% dos usuários de mecanismos de busca que continuaram a sua pesquisa quando não acham o que procuram, informaram que mudaram seu termo de busca se não acham o que procuram na primeira página; 88% reportam fazer o mesmo depois de três páginas;

- 36% dos usuários concordam que “ver uma companhia listada dentro dos primeiros resultados me faz pensar que esta é uma companhia que está no topo, dentro de sua área”;

Além disso, os resultados de investimentos em otimização para ferramentas de busca têm a característica de terem resultados de longo prazo. Ao contrário da mídia tradicional, um bom posicionamento pode ser considerado um ativo, pois continua gerando tráfego mesmo após o cessar de todos os investimentos.

Para empresas com orçamento limitado, a otimização pode fazer toda a diferença. Investir em um *setup* orientado a otimização pode ter um grande impacto no futuro crescimento e visibilidade do website dessas empresas [7].

Fica claro, tendo em vista esses dados, a importância de se posicionar bem dentro dos resultados de mecanismos de busca. No entanto, não é trivial mudar seu posicionamento. De acordo com Siqueira (2011) , o ponto de partida em um trabalho de otimização é identificar por quais palavras você quer brigar. Mesmo para sites com alta autoridade perante o Google (*Pagerank*[1]), é impossível competir por todas as palavras relacionadas ao negócio. Em sites que estão começando e o *PageRank* ainda não é alto, esse fato se agrava ainda mais. Por isso é importante escolher quais são os termos de busca para os quais você quer que seu site seja listado como primeiros resultados do Google e então definir estratégias para alcançar esse objetivo [4] .

2.1: Funcionamento dos Mecanismos de Busca

Entender como funcionam os mecanismos de busca é importante para entender como se posicionar melhor nos mecanismos de busca. Essa seção, e o restante do trabalho, tratarão do funcionamento da ferramenta Google, que devido ao grande *market share* [8] é a ferramenta escolhida para os testes.

De acordo com [6] as ferramentas de busca têm dois grandes objetivos:

[1] *PageRank* é um algoritmo de análise de rede. O algoritmo atribui um valor numérico a cada elemento de um conjunto de elementos, com o objetivo de medir a sua importância relativa ao conjunto.

- Fazer a indexação de bilhões de documentos (páginas e arquivos) acessíveis na rede;
- Responder a requisições dos usuários provendo listas de páginas relevantes;

2.1.1: Busca Orgânica

Entende-se por busca orgânica quando são apresentados os resultados de forma “natural”, ou seja, quando a ferramenta de busca fornece o resultados de forma imparcial, de acordo com um algoritmo pré-estabelecido.

Essa forma contrasta com a busca paga, aonde existe um sistema de leilão que permite ao maior pagador ocupar as melhores posições na busca.

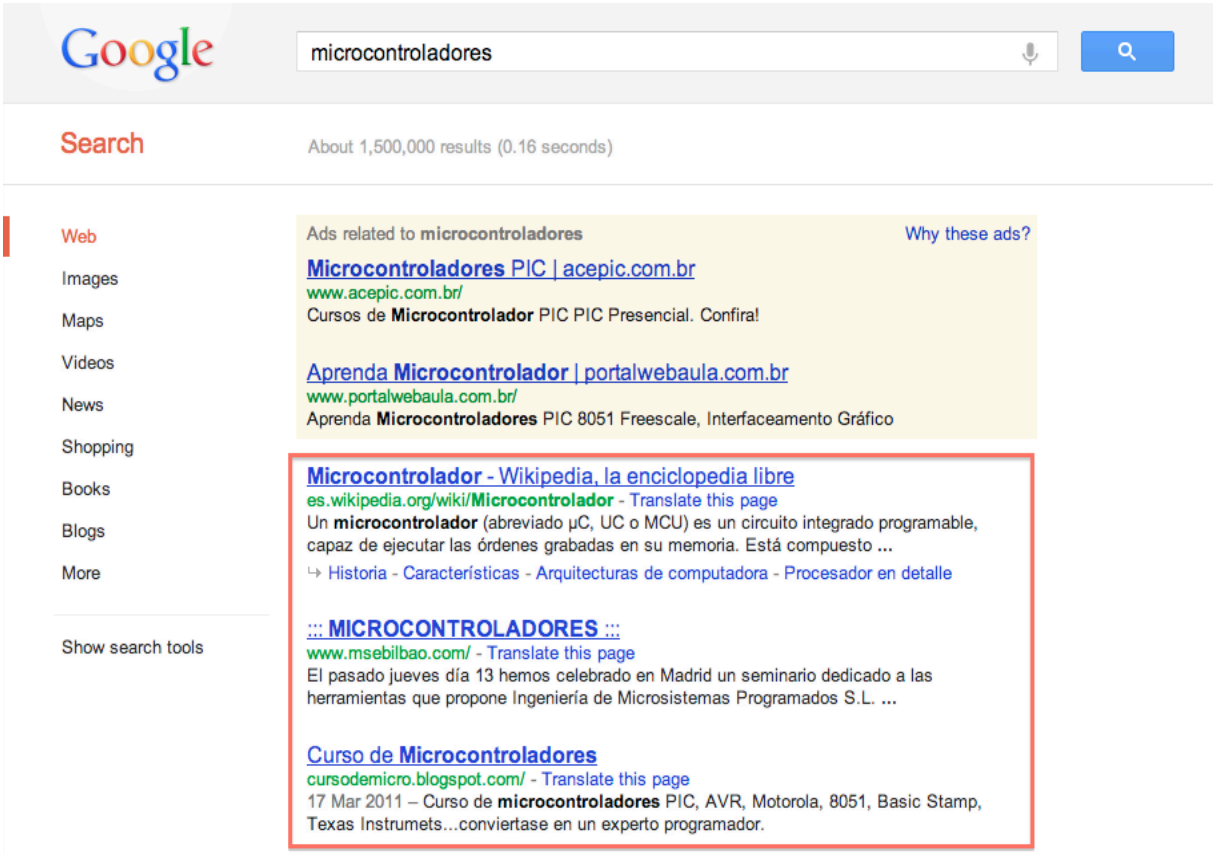

Figura 1 – Captura de tela com uma busca pela palavra microcontroladores

A Figura 1 mostra uma captura de tela com uma busca realizada no dia 26 de janeiro de 2012. A caixa vermelha evidencia os resultados orgânicos para a pesquisa pela palavra-chave "*microcontroladores*".

2.1.1.1: Indexação

Para atingir o primeiro objetivo as ferramentas de busca utilizam programas de computador conhecidos como *crawlers* ou *spiders [6]*. Neste capítulo o funcionamento desses programas será explicado de maneira muito simplificada.

Esses programas seguem o seguinte algoritmo :

- Acessar uma página na internet;
- Arquivar informações sobre essa página;
- Seguir os *links* para outras páginas e repetir o processo;

Dessa maneira, as ferramentas indexam uma quantidade massiva de dados sobre todas as páginas que são públicas por alguém ter colocado um *link* para ela.

2.1.1.2: Recuperação e Posicionamento

Quando um usuário faz uma pesquisa, espera encontrar no menor tempo possível as páginas mais relevantes. Para realizar essa tarefa as ferramentas de busca têm de percorrer todas as páginas do seu índice e de bilhões de documentos e primeiramente filtrar todas as páginas que não tem relação com a pesquisa, em seguida apresentar o resultado em ordem de importância percebida.

Os fatores usados para realizar essa ordenação não são públicos. O algoritmo usado pelo Google é proprietário e muda com frequência.

-

2.1.1.3: Otimização para Ferramentas de Busca

Baseado no conhecimento dos fatores que levam a um bom posicionamento nas ferramentas de busca, é natural pensar que exista alguma forma de influencia-lo.

De fato, existe hoje um conjunto de técnicas que ficaram conhecidas como SEO (Search Engine Optimization) ou em português Otimização para ferramentas de busca.

Essas técnicas são baseadas em experimentação e erro, e tem o objetivo de identificar os atributos que mais tem influência no posicionamento. Como exemplo podemos citar a pesquisa realizada pela empresa SEOMoz em 2011 [9]. De acordo com [6] estes foram os nove principais fatores que influenciam no *ranking*:

- Uso da palavra-chave na etiqueta HTML *title*:
  Presença da palavra-chave na etiqueta <*title*> do documento HTML
- Texto âncora dos *links* externos:
  Presença da palavra-chave nas etiquetas *a* (âncora) dos *links* de outros *websites* para esta página
- Autoridade global do *website*:
  Esta métrica se refere a autoridade do domínio. É uma medida da quantidade e qualidade dos links para qualquer página do domínio.
- Idade do site;
- Popularidade do link dentro da estrutura interna do *website*:
  As ferramentas de busca usam a estrutura interna do *website* classificar as páginas. Uma página que aparece na estrutura de navegação do website consequentemente tem links em muitas páginas internas, e é vista como uma página importante.
- Relevância tópica dos links externos:
  Links provenientes de websites aonde o tópico é não relacionado são benéficos, mas não tanto quanto *links* aonde o tópico é mais relevante.
- Popularidade de links na comunidade tópica:
  Esse fator se refere a construção de autoridade na comunidade tópica. Se um website tem muitos links de websites relacionados, isso demonstra um forte voto de confiança.
- Uso da palavra-chave no texto da página:
  A existência da palavra-chave no texto da página reforça a importância da página para esta palavra-chave.
- Autoridade dos sites que têm links para a página:

Outro fator importante é a importância dos *links* externos*, Links* de páginas que têm autoridade, trazem autoridade. Esse atributo têm consequência uma autoridade global maior.

2.1.2: Busca Paga

A busca paga, também conhecida como *links* patrocinados, é uma forma de publicidade contextual aonde são mostrados anúncios a um usuário que faz uma busca em uma ferramenta de busca.

No caso do Google, o sistema funciona com um sistema de leilões. De forma simplificada o funcionamento do sistema (chamado Google AdWords [20]) é o seguinte: A cada vez que um usuário faz uma busca, a ferramenta mostra os anúncios dos anunciantes que estão pagando mais por aquela determinada palavra-chave.

O valor é pago por clique, ou seja, a cada vez que um usuário clica em um anúncio, aquele anunciante paga uma quantia referente ao lance que deu.

A busca paga tem muito potencial a curto prazo. De acordo com [7] , links patrocinados podem ser altamente segmentados e produzir tráfego de excelente qualidade para seu site – muitas vezes superando a busca orgânica.

Contudo, devido a sua natureza, os resultados da busca paga cessam assim que os investimentos cessam. E ao longo do tempo, com o aumento da concorrência existe a tendência de que o custo de se manter uma campanha de busca paga se tornem proibitivos.

A Figura 2 mostra um exemplo de resultado de busca paga para uma pesquisa pela palavra “*microcontrollers*”. A caixa vermelha evidencia os anúncios.

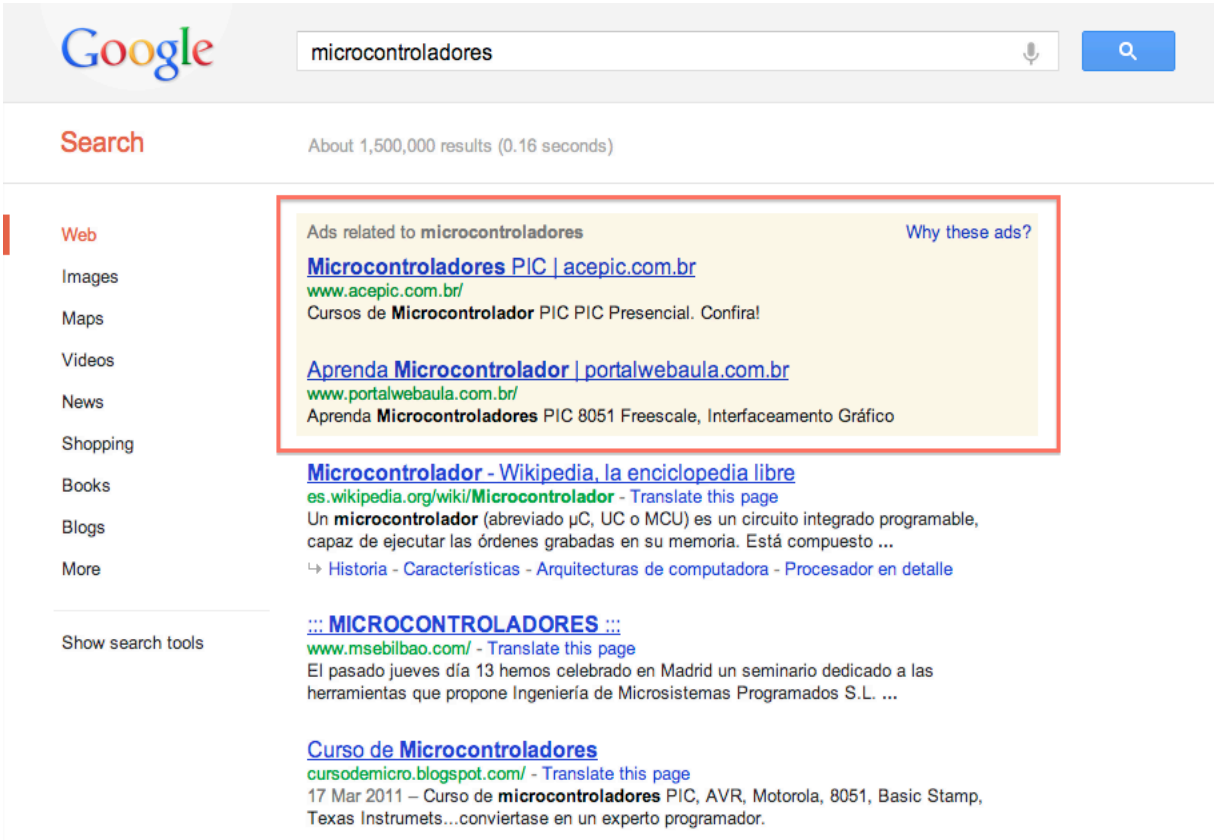

Figura 2 – captura de tela com uma busca pela palavra microcontroladores

Capítulo 3: Software RDStation

Baseado em um modelo Software como Serviço (SaaS) o software RDStation tem como objetivo ser uma plataforma de Marketing Digital integrada e intuitiva, orientada para que Médias e Pequenas empresas tenham resultados efetivos com Marketing Digital.

Suas funcionalidades variam entre diferentes áreas como produção de conteúdo, otimização para ferramentas de busca (SEO), monitoramento de mídias sociais, relacionamento (*e-mail* e mídias sociais), geração e nutrição de oportunidades e análise de desempenho.

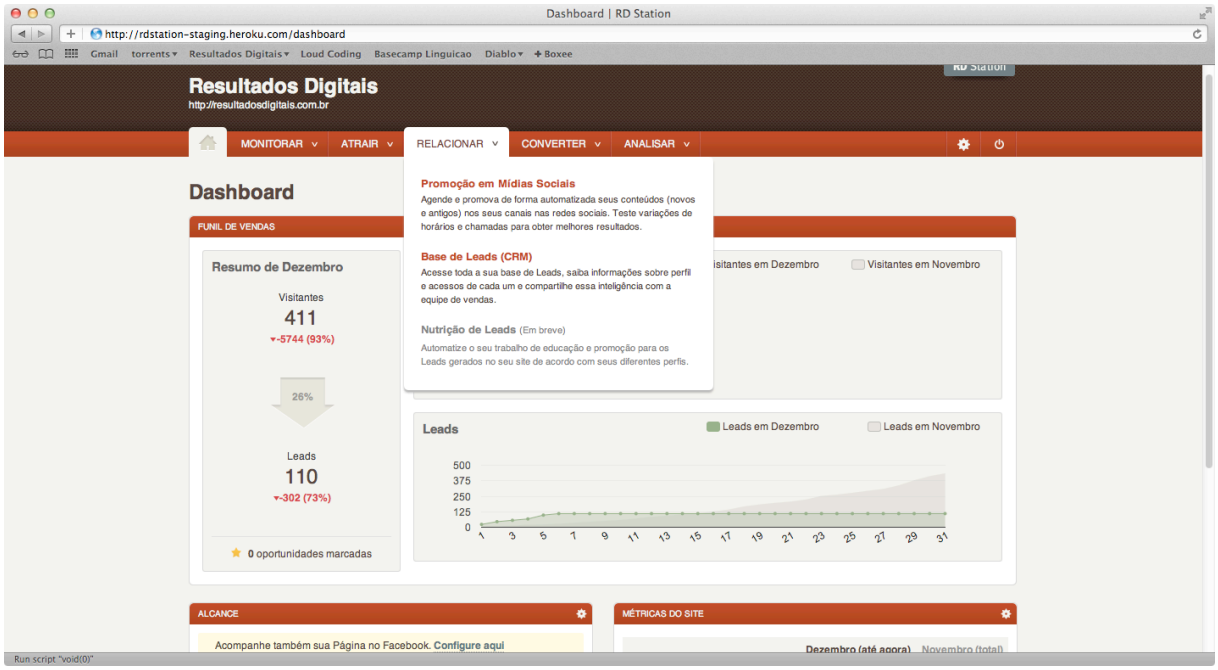

Figura 3 – Captura de tela do software RDStation

3.1: O Painel de Palavras-Chave

Com o objetivo de auxiliar os usuários a alcançarem os objetivos levantados pelo capítulo 2, existe dentro do software RDStation uma funcionalidade chamada Painel de Palavras-Chave.

Essa funcionalidade permite que o usuário gerencie melhor seus esforços em garantir seu melhor posicionamento no Google. O passo a passo de uso da ferramenta é o seguinte:

1. O usuário faz um levantamento de palavras-chave que tem potencial de geração de oportunidades de negócio.
2. As palavras-chaves são inseridas em uma área específica do software.
3. A cada palavra o usuário atribui uma relevância, ou seja, um percentual de importância que ela representa para seu negócio. (Ver Capítulo 4.2.1.4:)

O software então, assincronamente e de modo transparente para o usuário faz o levantamento de mais três informações para cada palavra-chave. O Custo por Clique Médio, o volume de buscas mensal, e concorrência (expressada em porcentagem).

Além disso o software faz uma pesquisa para saber se alguma das páginas do domínio do usuário está entre as 100 primeiras posições para uma pesquisa feita com esta palavra-chave. Caso positivo, o *software* guarda também o histórico desse posicionamento para que o usuário possa acompanhar a evolução dessa métrica.

A Figura 4 mostra um exemplo real da funcionalidade com os dados de cada palavra-chave (as próprias palavras-chave foram omitidas por questão de sigilo). Observa-se que não existem informações estratégicas, somente informativas.

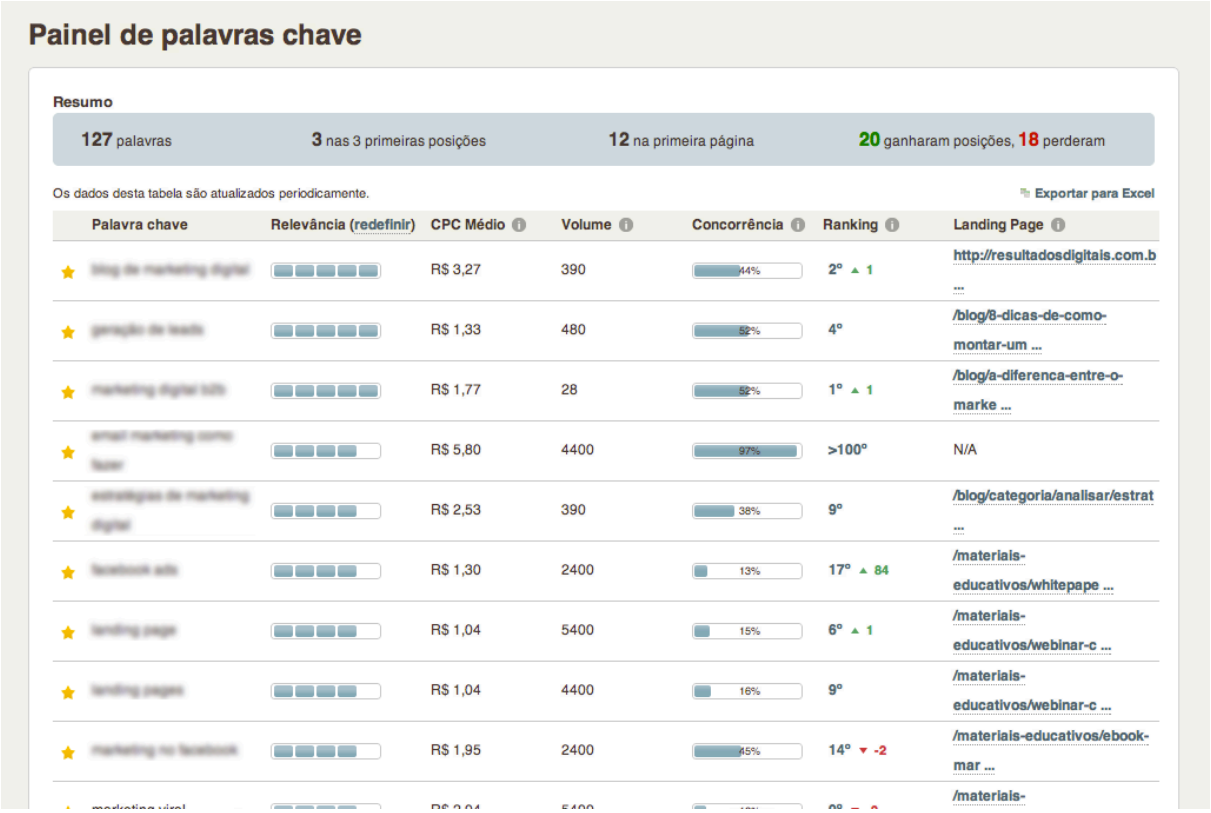

Figura 4 – Captura de tela do Painel de palavras-chave.

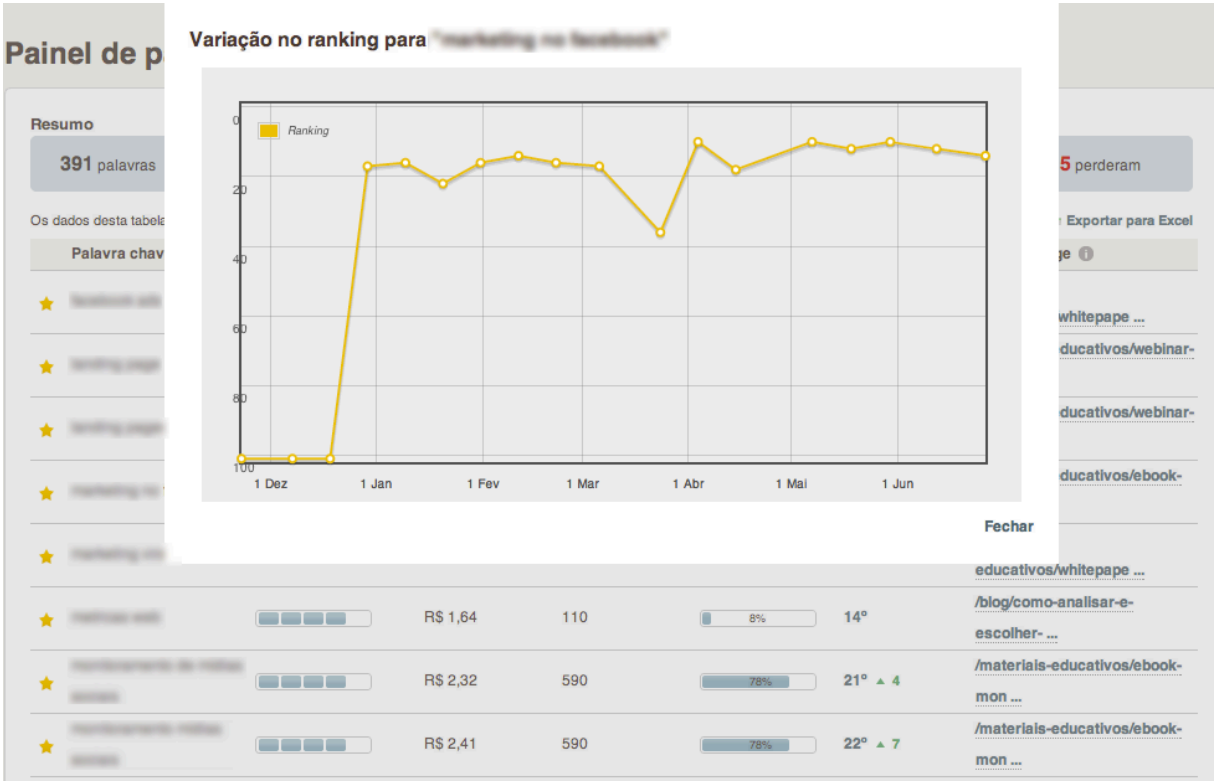

Figura 5 – Captura de tela do detalhe de posicionamento

3.2: Problemas com a solução atual

Como o algoritmo das ferramentas de busca não é de domínio público, e está continuamente mudando, o problema de decidir que ação tomar dada uma palavra-chave com uma série de atributos não é trivial.

Atualmente o Painel de Palavras-Chave do RDStation não fornece nenhuma informação acionável para seus usuários, ou seja, espera-se que o usuário interprete o conjunto de dados e decida quais palavras priorizar e que ações tomar. Como os usuários não tem o conhecimento de domínio necessário para fazer essas decisões, a funcionalidade não está atendendo as necessidades do público alvo.

Além disso, uma boa parte do processo de obtenção dos parâmetros de cada palavra-chave é feita hoje de forma manual. Ainda que de forma transparente para o usuário, a cada vez que um painel de palavras-chave é alterado, um colaborador da Resultados Digitais deve consultar outras ferramentas e em uma interface administrativa do *software*, inserir essas informações para que o cliente possa ver.

Esse processo manual é claramente não escalável, e atualmente é considerado, um gargalo na empresa. É muito importante portanto, automatizar este processo.

3.3: Detalhamento do problema

O problema então acontece em dois níveis. O primeiro nível se refere a automatização e expansão da coleta de dados referentes a cada palavra-chave. Esta é uma tarefa trivial do ponto de vista de tecnologia. Dado tempo suficiente, pode ser resolvida facilmente através de programação.

Em um segundo nível está o problema principal deste trabalho. Para que o Painel de Palavras-Chave possa atender as necessidades dos clientes é necessário que o software seja capaz de realizar a seguinte tarefa:

> Para cada palavra-chave inserida pelo usuário recomendar a ação de melhoria que mais trará benefícios caso seja realizada.

Além disso, tendo em vista que as ferramentas de busca estão em constante aprimoramento e mudança, é de grande importância que este *software* possua uma característica de evolução constante, aprendendo com seu erros e acertos.

O objetivo deste trabalho é então propor uma estratégia para este algoritmo central, que classifica as palavras-chave e aprende a partir dessas classificações.

Capítulo 4: Solução Proposta

Enxergando o sistema como uma caixa-preta temos (Figura 6):

- Entrada: Palavra-chave e seus dados relacionados
- Saída: Recomendação

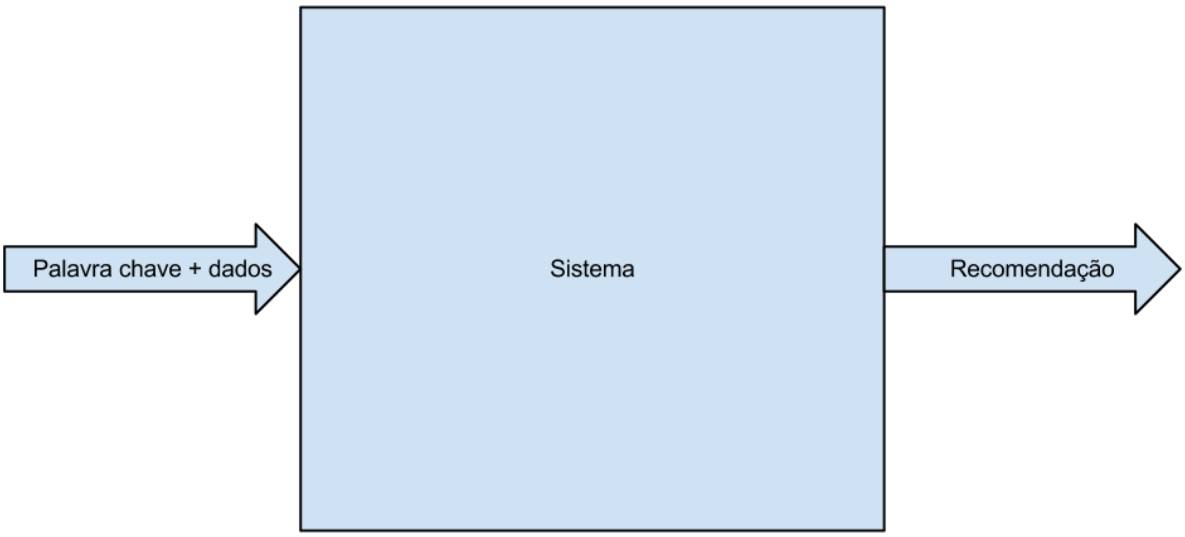

Figura 6 - Visão caixa-preta do sistema

O modelo que relaciona os atributos com a classe a qual a palavra-chave pertence não é conhecido. Contudo, um especialista na área, devido ao seu conhecimento prévio, é capaz de realizar essa classificação.

De fato, na Resultados Digitais há especialistas nesse campo, que com frequência realizam essa classificação para os clientes que contratam, além do software, um pacote de consultoria.

Pode-se pensar então que esses especialistas possuem dentro do seu conhecimento um modelo para realizar essas recomendações.

Traduzir esse conhecimento para um formato que uma máquina possa entender e replicar é uma tarefa complexa, muita vezes as decisões estão ligadas a conceitos muito abstratos como a intuição.

Dado a natureza do problema, pode-se dizer que é um **problema de classificação**.

De acordo com [11] , problemas aonde a tarefa é classificar exemplos em uma das categorias de um conjunto discreto, são comumente conhecidos como problemas de classificação.

Ou seja, para cada palavra-chave deve-se analisar seus atributos e decidir qual é o conjunto ao qual essa palavra-chave pertence. Este conjunto definirá qual recomendação o software dará ao usuário.

Ainda de acordo com [11] , a Aprendizagem de Máquina tem sido aplicada a uma variedade de banco de dados extensos afim de aprender modelos implícitos nos dados.

A partir disso, pode-se sugerir como solução para o problema a aplicação de um algoritmo de aprendizagem de máquina, de forma que este aprenda a classificar as palavras-chave de maneira eficiente.

Isso vai ao encontro do requisito de auto-aperfeiçoamento, uma vez que cada recomendação feita pelo sistema é uma oportunidade de aprendizado.

4.1: Aprendizagem de Máquina

A Aprendizagem de Máquina é uma técnica multidisciplinar que busca construir programas de computador que aprendem com a experiência.

De acordo com [11] , as técnicas de aprendizagem de máquina são especialmente úteis em problemas de mineração de dados aonde largos banco de dados podem conter padrões valiosos que podem ser descobertos automaticamente.

4.1.1: Definições do aprendizado

De acordo com [11] a definição de aprendizagem de máquina é:

*Um programa de computador aprende com a experiência **E** que diz respeito a uma classe de tarefas **T** com medida de performance **P**, se a sua performance em tarefas do tipo **T**, aumenta como medidas por **P** com a experiência **E**.*

Com base nessa definição, para este problema pode-se atribuir:

- Tarefa T: Classificar palavras-chave.
- Medida de performance P: Percentagem de palavras classificadas corretamente, ou seja, palavras para as quais a recomendação gere resultados positivos.
- Experiência: um conjunto de palavras-chave e seus atributos com as dadas classificações.

Define-se também resultados positivos como :

Uma mudança positiva no posicionamento em pesquisas pela palavra-chave de uma das páginas pertencentes ao domínio do cliente.

4.2: Variáveis usadas na classificação

Este capítulo aborda cada um dos atributos que serão utilizados para a classificação. Esses atributos foram julgados importantes pelos especialistas da empresa. Tomou se o cuidado de incluir a maior número de atributos possível. Caso algum atributo não seja realmente relevante, será simplesmente desconsiderado pelas técnicas de Aprendizagem de Máquina.

Para facilitar o entendimento, eles estão agrupados de acordo com suas características. Foram agrupados em Valor para o negócio, Dificuldade e Capacidade.

4.2.1: Valor para o negócio

Aqui se encaixam os atributos que demonstram um possível valor para o negócio. Ou seja, quanto mais altos os valores destas variáveis, mais interessante é para a empresa obter uma melhor classificação para essa palavra.

4.2.1.1: Volume de buscas mensais

Esse é o número mensal de buscas que são feitas na ferramenta Google.

Quanto mais alto este valor, mais tráfego trará a palavra-chave caso o cliente consiga um bom posicionamento. Usa-se aqui a correspondência ampla, que de acordo com [12] é quando os resultados da pesquisa podem também exibir formas no singular ou plural, sinônimos, variações (como *casa* e *casinha*), pesquisas relacionadas e outras variações relevantes.

4.2.1.2: Custo por clique aproximado

De acordo [12] com o CPC (custo por clique) é o valor aproximado que se pagaria case definisse um lance no Google AdWords para a palavra-chave.

O CPC nos dá uma informação indireta de qual é o valor da palavra-chave. Um valor de lance grande significa que muitas empresas estão competindo por esta palavra, o que sugere um valor de negócio alto.

Adicionalmente uma palavra que tem um valor alto de CPC e está bem posicionada nas buscas orgânicas está efetivamente economizando o valor do lance a cada vez que um visitante acessa o *website* por meio da busca.

4.2.1.3: Competição em busca paga

A competição é um valor percentual que informa quantos anunciantes há para a pesquisa paga para essa palavra-chave. Assim como o CPC, uma grande concorrência pode indicar alto valor de negócio.

4.2.1.4: Relevância

Aqui temos duas variáveis, que representam o valor da palavra para o usuário. A primeira é a relevância auto-atribuída, e a segunda é a taxa de conversão.

4.2.1.4.1: Auto atribuída

A Relevância auto-atribuída é informada pelo próprio usuário e reflete o quanto o usuário acredita ser o potencial dessa palavra para geração de oportunidades.

Essa atribuição é feita de acordo com uma metodologia desenvolvida pela Resultados Digitais. Para fazer essa atribuição o usuário é instruído a se fazer as seguintes perguntas.

1. Pela lógica, essa palavra-chave representa o meu produto ou serviço?
2. A palavra tem relação com o conteúdo do meu site?
3. Quando alguém pesquisa por essa palavra, o meu *website* seria um resultado interessante para o usuário?

Cada palavra deve ser julgada sob esses aspectos e receber uma nota de 1 a 5, onde 5 significa que a resposta para todas as pergunta deve ser muito positiva.

4.2.1.4.2: Taxa de Conversão de Visitantes

O segundo atributo relacionado a relevância é a taxa de conversão.

Ele representa a taxa em que os visitantes que entram pelo *website* por meio de busca pela palavra-chave “convertem”.

Essa conversão é muitas vezes associada a, por exemplo, um preenchimento de formulário. É aqui que o visitante deixa de ser um anônimo e se torna um *Lead*, ou seja, uma oportunidade ou potencial cliente. Isso demonstra que a palavra-chave está alinhada com as expectativas do visitante, ou seja, este visitante achou o *website* buscando pela palavra-chave em questão e julgou a informação apresentada interessante o suficiente para deixar seus dados.

Esse é um atributo interessante de ser analisado pois mostra o valor real da palavra-chave. Uma palavra com alta taxa de conversão tem grande valor para o negócio, mesmo que não tenha uma volume de buscas tão grande.

É importante porém que esse atributo tenha validade estatística, ou seja e aceitável que a taxa de conversão seja 75% quando tem-se somente 4 visitas ao site. Dentro do contexto do trabalho assumiremos que os valores entregues ao sistema atendem esse requisito.

4.2.2: Dificuldade

A dificuldade é um valor em percentual do grau de dificuldade de se obter um bom posicionamento. Ele é calculado a partir de uma série de outras métricas como a autoridade de página e de domínio dos 10 primeiros colocados no Google.

É um valor absoluto, no sentido de não levar em consideração a capacidade própria do usuário em alcançar as posições.

4.2.3: Capacidade

Os atributos pertencentes a este grupo tem relação com a capacidade do usuário para melhorar seu índice no resultado das buscas.

4.2.3.1: Posição atual

A posição atual reflete a melhor posição de qualquer página pertencente ao domínio do usuário quando realizada uma busca por essa palavra-chave.

4.2.3.2: Autoridade do domínio

A autoridade do domínio perante o Google é um forte indicativo de capacidade. Por exemplo, o domínio “wikipedia.org” tem uma autoridade muito grande e por isso (aliado a outros motivos) aparece sempre bem posicionado em uma grande variedade de pesquisas.

Esse atributo está fortemente relacionado ao número de *links* externos que o domínio recebe.

4.2.3.3: Atributos de capacidade associados a página

Os próximos atributos estão associados a uma página específica. Neste caso será usada sempre a página com melhor posicionamento nas pesquisas. Caso não existe uma página posicionada estes atributos não poderão ser obtidos, e portanto serão tratados como desconhecidos.

4.2.3.4: Autoridade da página

Como a autoridade do domínio, está fortemente ligado ao número de *links* recebidos, porém, aqui somente a própria página é levada em consideração.

4.2.3.5: Nota de análise estrutural da página

A nota de análise estrutural da a informação de quão bem otimizada está a página para as ferramentas de busca. Pode assumir valores de 0 a 100, aonde 100 representa uma página “perfeita” .

4.2.3.6: Sinais Sociais

Pesquisas recentes ([16]) mostram uma relação forte entre as interações entre as mídias sociais e as páginas com o posicionamento nas ferramentas de busca. Esse atributo mede o nível dessas interações. Os valores estão entre 0 e 100, aonde 100 representa fortíssimo nível de interação.

Valor para o negócio	• Volume de buscas mensais • Custo por clique aproximado • Competição em busca paga • Relevância
Dificuldade	
Capacidade	• Posição atual • Autoridade do domínio • Atributos de capacidade associados a página • Autoridade da página • Nota de análise estrutural da página • Sinais Sociais

Tabela 1 – Resumo das variáveis usadas na classificação

4.3: Recomendações Possíveis

A função principal do sistema é fornecer uma recomendação sobre qual a melhor estratégia para alavancar os resultados perante os mecanismos de busca.

As recomendações podem ser de três tipos, sendo a) criar novo conteúdo, b) otimizar conteúdo existente e c) nenhuma ação.

4.3.1: Criar conteúdo novo

A recomendação neste caso é criar um novo conteúdo disponível na *Web*, que esteja otimizado para a palavra-chave.

4.3.2: Otimizar conteúdo existente

A estratégia aqui é otimizar o conteúdo que já esta aparecendo nos resultados das ferramentas de busca. Espera-se que trabalhos futuros tragam melhorias no sentido de especificar qual técnica trará os melhores resultados.

4.3.3: Nenhuma ação

Significa que a palavra-chave é provavelmente muito difícil de ser conquistada e/ou não traria resultados expressivos. Então é preferível que esta palavra seja ignorada em favor de outras.

4.4: Arquitetura do Software

Neste capítulo apresenta-se de maneira geral a arquitetura proposta do *software*.

O sistema será composto por três módulos

- Classificador : É o responsável pela classificação de cada novo exemplo. Realiza também o auto-aperfeiçoamento baseado no resultado da avaliação.
- Avaliador : Responsável por avaliar as classificações.
- Banco de dados : É o ponto comum entre o avaliador e o classificador, registra todas as classificações.

A Figura 7 mostra o fluxo de informações dentro do sistema.

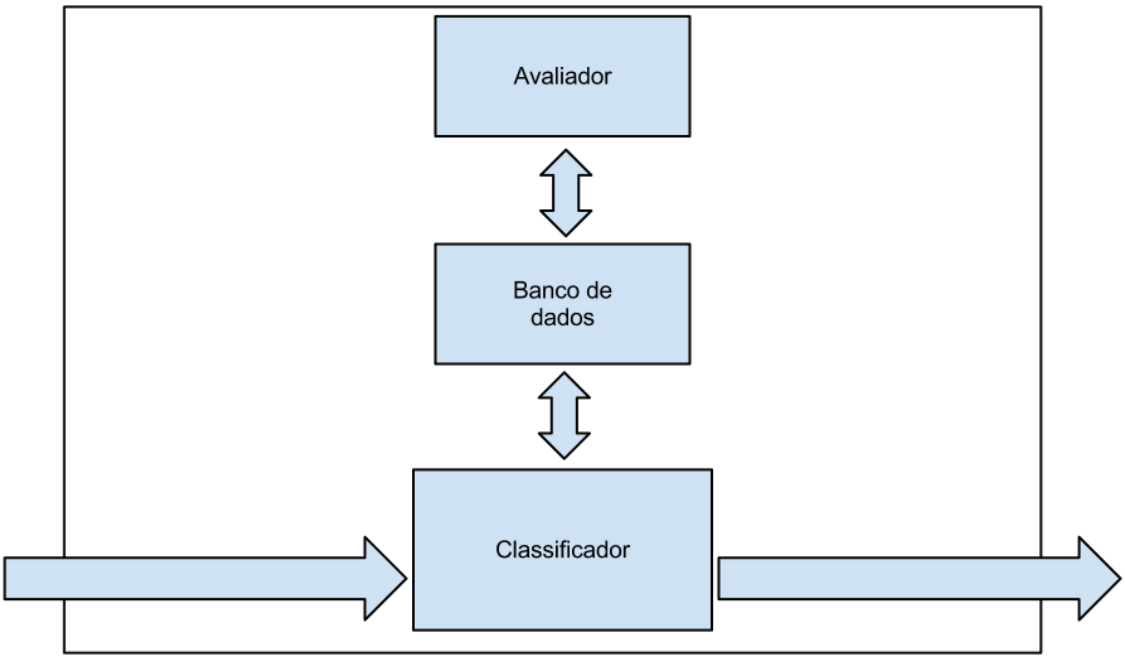

Figura 7 – Fluxo de Informações dentro do sistema

4.4.1: Classificador

O Classificador é a peça central do *software*. É dele a responsabilidade de fornecer as recomendações para cada nova palavra-chave.

A estratégia para a classificação de palavras-chave se baseia em implementação de uma técnica de Inteligência Artificial, gerada com técnicas de Aprendizagem de Máquina.

Existem diversas técnicas, cada uma com suas características, pontos fortes e fracos. É então interessante, avaliar qual dessas técnicas melhor se adapta com o problema em questão.

Para isso originalmente objetivou-se testar cada uma das técnicas e comparar os resultados qualitativos e quantitativos para com isso tomar a decisão de qual é a mais adequada.

Além disso, o módulo classificador tem também a função de auto-aperfeiçoamento, ou seja, a partir de novas classificações, e a avaliação do impacto destas, o módulo pode aprender continuamente de modo a sempre melhorar suas recomendações.

4.4.2: Banco de dados

O banco de dados tem dois propósitos. O primeiro é fazer o registro das classificações para essas possam ser usadas futuramente para o aperfeiçoamento do classificador.

A segunda é servir de ponte para a comunicação entre o Classificador e o Avaliador. Ou seja, como os dois módulos não possuem um canal de comunicação direta, é realizando consultas e inserindo informações no banco periodicamente que ocorre a troca de informações.

4.4.3: Avaliador

O módulo Avaliador é o responsável por avaliar as classificações feitas pelo módulo classificador. É uma tarefa um tanto complexa e se dá em dois níveis.

No primeiro nível é necessário que se avalie se a sugestão foi implementada, ou seja, se o cliente seguiu a recomendação. Para cada recomendação existe uma avaliação diferente.

- Criar conteúdo novo : Verificar se o cliente criou e disponibilizou uma página nova que tem potencial para se posicionar bem em pesquisas por essa palavra-chave.
- Otimizar Conteúdo existente : Verificar se o cliente fez alguma das estratégias mencionadas no capítulo 2.1.1.3: . Essa tarefa apresenta grandes desafios. Por exemplo caso a otimização seja conseguir mais *links* externos para a página, não há hoje como monitorar isso.
- Nenhuma ação : Neste caso a avaliação é bem subjetiva. É preciso avaliar se “não fazer nada” com essa palavra-chave possibilitou concentrar esforços em palavras com mais potencial e portanto obter melhores resultados em menor tempo.

Observa-se que essas tarefas são complexas tanto no domínio da variedade quanto em profundidade. Isso sugere uma dificuldade grande de avaliação automática. De fato, é muito mais razoável que essa tarefa seja desempenhada por um humano pelo menos um primeiro momento.

A partir deste contexto surge a necessidade da existência de uma interface para que o operador humano possa avaliar as classificações. Portanto modificou-se a estrutura geral do sistema para a apresentada na (Figura 8) . Observa-se que existe mais uma entrada de informação de fora do sistema.

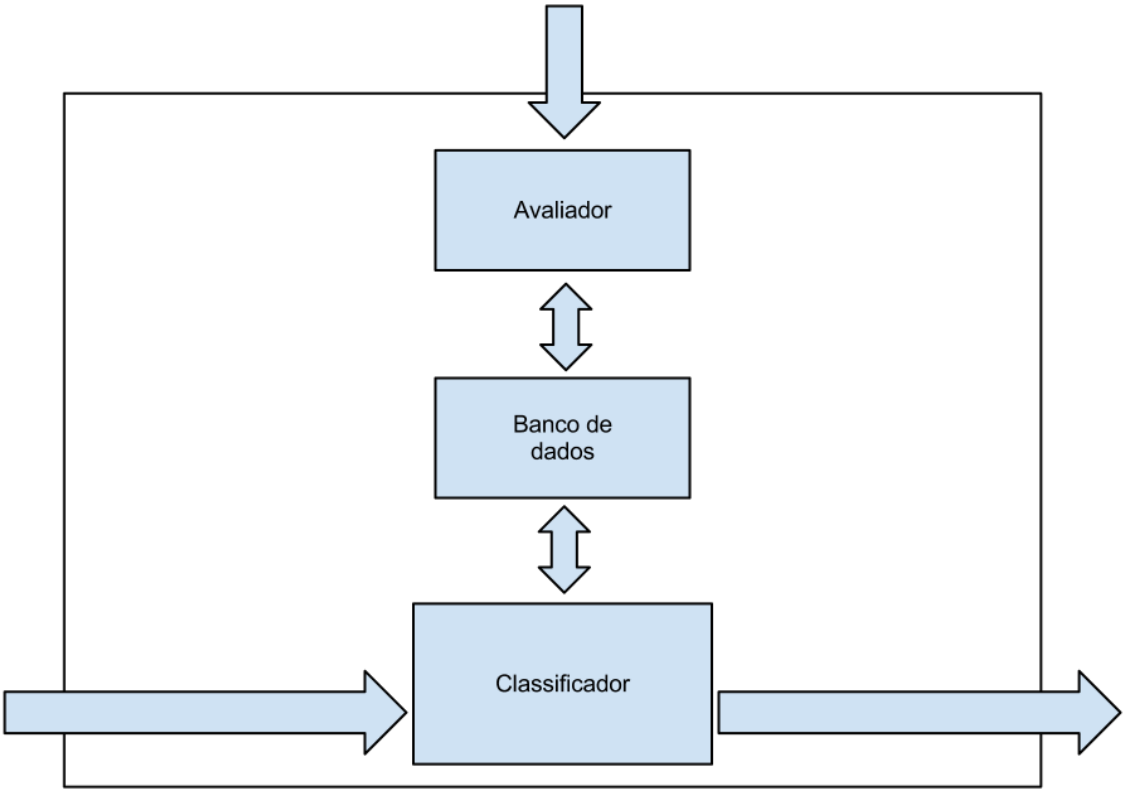

Figura 8 – Sistema expandido com interface para avaliação humana

Capítulo 5: Desenvolvimento

Para testar e validar a proposta apresentada optou-se por desenvolver uma versão de testes não integrada com o sistema final.

Essa implementação utilizou a linguagem Ruby [21], apoiada pelo *framework*[2] Ruby on Rails[22] (Rails).

Rails é um framework para aplicações *Web* que inclui todo o necessário para criar aplicações web suportadas por banco de dados de acordo com o paradigma Model-View-Controller (MVC) [14].

5.1: Classificador

Para desenvolver o módulo classificador de acordo com a proposta original, seria necessário um conjunto grande de dados para avaliar as várias técnicas de aprendizagem de máquina. Contudo, como observa-se no capítulo a seguir, isso não foi possível.

5.1.1: O problema da falta de dados

As técnicas de Aprendizagem de Máquina dependem fortemente de um conjunto de dados que represente a experiência a partir da qual o algoritmo aprenderá. Esse conjunto de dados deve ser expressivo o suficiente para permitir que o algoritmo possa construir um modelo confiável. Contudo , na Resultados Digitais até o momento não houve a preocupação do registro de recomendações feitas pelos especialistas aos clientes.

Dessa forma, pode-se ver a necessidade de se obter um conjunto de dados inicial para que as técnicas de Aprendizagem de Máquina possam ser aplicadas.

Uma forma de solucionar este problema seria gerar aleatoriamente um conjunto de exemplos que seriam classificados por um especialista. No entanto, devido ao custo de se disponibilizar um especialista pelo tempo necessário para se dedicar a essa tarefa ser proibitivo, essa possibilidade foi descartada.

[2] Plataforma reutilizável usada para desenvolver aplicações, produtos e soluções [23].

Para resolver este problema, propôs se então uma alteração na solução.

Ao invés de um algoritmo de classificação gerado a partir de um conjunto de exemplos, inicialmente o algoritmo será gerado a partir do conhecimento dos especialistas humanos.

Esse algoritmo começará a classificar os exemplos gerados pelos clientes, fornecendo recomendações. Essas recomendações terão um certo desempenho de acordo com o critério definido no capítulo 5 que poderão então, quando atingirem um número suficiente de exemplos ser usadas como entrada para a solução original do problema.

A maior vantagem dessa estratégia é a geração de valor a curto prazo, ou seja, em pouco tempo a ferramenta trará recomendações para os clientes, efetivamente resolvendo o problema. Contudo, sabe-se que esse algoritmo provavelmente terá falhas e tendo em vista a característica mutável das ferramentas de busca, é interessante que a solução original seja aplicada e eventualmente substitua a implementação inicial.

5.1.2: Algoritmo inicial

Para obter o algoritmo inicial marcou-se uma reunião com os especialistas na área que chegaram a um consenso. O algoritmo (representado pela Figura 9) pode ser descrito em linguagem natural da seguinte forma:

> Se não houver nenhuma página entre as 100 primeiras posições, a recomendação é criar um conteúdo novo. Caso contrário analisa-se uma variável X. Caso X seja menor do que 30 a recomendação é otimizar a página existente, caso contrário a recomendação é não fazer nada.

A variável X é calculada da seguinte maneira:

$$X = 0.75 * Rk + 0.25 * Qa$$

Aonde, Rk representa o ranking atual e Qa é calculada pela seguinte fórmula.

$$Qa = 0.4 * Pa + 0.3 * PG + 0.3 * SS$$

Aonde Pa representa a Autoridade da página, PG a nota de análise estrutural da página, e SS a nota de sinais sociais.

Com base nisto, primeiramente surge a necessidade de incluir a variável X no conjunto de atributos. Essa variável é especial pois não entrará como informação externa, mas será calculada a partir dessas.

Em seguida, partindo da observação de como foi estruturada o algoritmo imaginado pelos especialistas, propôs-se uma implementação usando a técnica de aprendizado de máquina conhecida como Árvores de Decisão. A árvore de decisão tem uma característica muito relevante para este problema que é a facilidade de entendimento do algoritmo por um ser humano. Ao contrário de por exemplo uma Rede Neural [10], a Árvore de decisão pode ser interpretada facilmente. Isso é importante neste ponto pois deseja-se obter um algoritmo que seja o mais próximo possível, idêntico se possível, ao proposto pelos especialistas. Essa facilidade da Árvore de Decisão tornará possível a avaliação do algoritmo *a posteriori*.

Além disso o algoritmo de aprendizado C4.5 é capaz de lidar com o problema de atributos com valores desconhecidos, o que reforça mais ainda a adequação da técnica a esse problema.

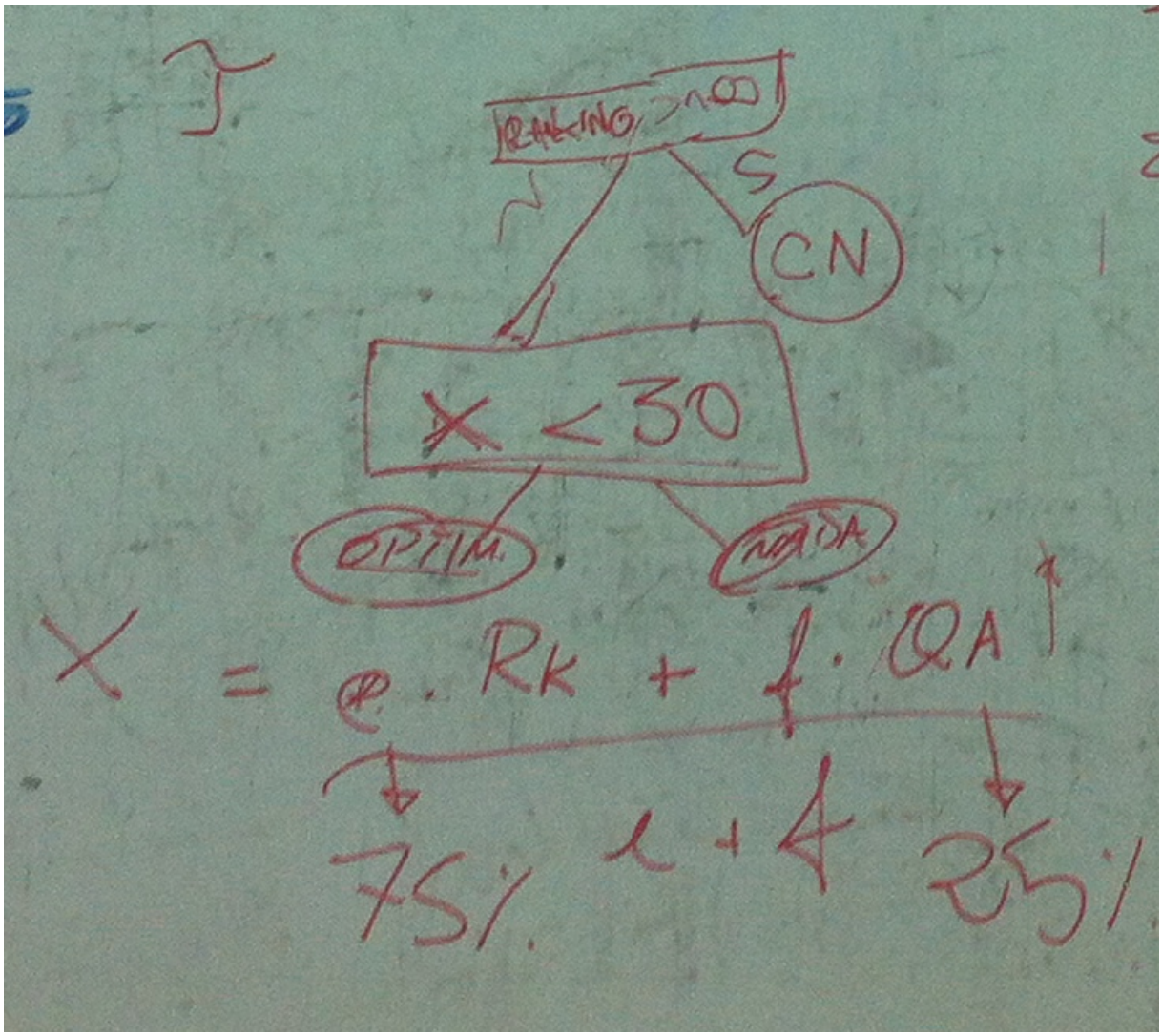

Figura 9 – Foto do resultado da reunião para determinar o algoritmo inicial

5.1.2.1: Árvores de Decisão

Árvore de decisão é uma técnica de aprendizado de máquina para construir modelos de predição de um conjunto de dados. Os modelos são obtidos por particionamento recursivo do espaço de dados e a associação a um modelo simples de predição a cada partição. Como resultado o particionamento pode ser representado graficamente como uma árvore de decisão [8].

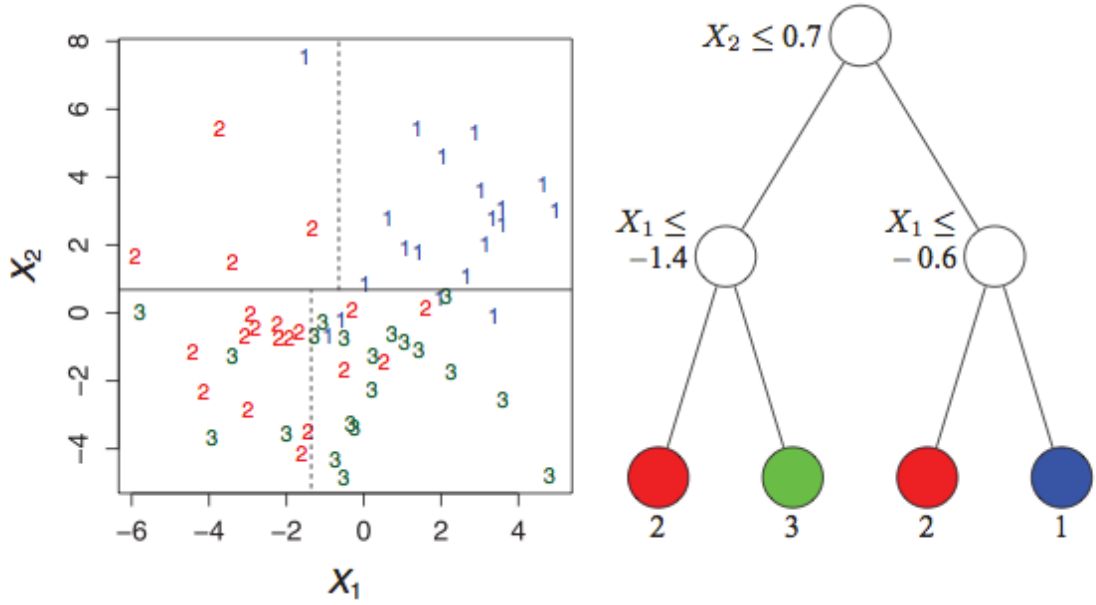

Figura 10 – Partição e árvore correspondentes [10]

Em um problema de classificação tem-se uma amostra de n observações de uma variável de classe Y que pode assumir valores 1,2,..., k e p variáveis preditoras, X1,...,Xp. O objetivo é achar um modelo para predizer o valor de Y dados novos valores de X. A Figura 10 mostra um exemplo aonde existem 3 classes e duas variáveis. A figura da esquerda mostra os pontos e partições que eles representam e a da direita mostra a árvore de decisão correspondente.

5.1.2.1.1: Algoritmo C4.5

C4.5 é um algoritmo usado para gerar uma árvore de decisão que foi desenvolvido por Ross Quinlan [24]. O algoritmo constrói árvores de decisão de um conjunto de dados de treinamento usando o conceito de entropia da informação.

Os dados de treinamento são um conjunto $S = s_1,s_2,...$ de exemplos previamente classificados. Cada elemento $si = x_1,x_2,...$ é um vetor aonde $x_1,x_2,...$ representam atributos da amostra. Os dados de treino são aumentados com um vetor $C = c_1,c_2,...$ aonde $c_1,c_2,...$ representam a classe da qual o exemplo pertence.

A cada nodo da árvore, o algoritmo escolhe um atributo dos dados que melhor divide o conjunto de exemplos de acordo com o critério de ganho de informação normalizado .

O ganho de informação de um conjunto ‘Ex’ aplicado a um atributo ‘a’ é calculado como a diferença normalizada entre a entropia do conjunto e a entropia dos subconjuntos. Pode ser representado pela seguinte fórmula, aonde H representa a entropia.

$$IG(Ex,a) = H(Ex) - \sum_{v \in values(a)} \frac{|\{x \in Ex | value(x,a) = v\}|}{|Ex|} \cdot H(\{x \in Ex | value(x,a) = v\})$$

O ganho de informação representa a redução na entropia quando aplicado o particionamento .

Sendo assim, o atributo que possuir o maior ganho de informação normalizado é escolhido para fazer a decisão. O algoritmo então continua iterando nos subconjuntos resultantes até que todos os exemplos de um subconjunto pertençam a uma mesma classe (*i.e.* entropia igual a 0), aonde então é colocada uma “folha” na árvore que representa uma decisão para aquela classe [11].

Em termos de tecnologia utilizou-se a biblioteca de Aprendizagem de Máquina disponibilizada pela ferramenta Weka [13]. Essa biblioteca fornece a implementação do algoritmo C4.5 e pode ser integrada diretamente ao código. Apesar de não estar licenciada para aplicações comerciais, para o contexto desta pesquisa a ferramenta foi de extrema serventia.

O algoritmo C4.5 é aplicado sobre um arquivo .arff presente no sistema sempre que o servidor é iniciado. A árvore resultante é preservada durante toda a execução, de modo que para mudar a árvore é preciso reiniciar o servidor para que este utilize novamente o arquivo relacional e gere a árvore novamente.

5.1.3: A geração da árvore inicial

Uma possível estratégia para termos as mesmas decisões do algoritmo é converte-lo em um conjunto de regras e implementar estas regras como estruturas se/senão (if/else). Para o algoritmo em questão têm-se as seguintes regras:

- Se Ranking > 100 Saída é igual a “Cria conteúdo novo”;
- Se Ranking > 100 E X < 30 Saída é igual a “Otimiza conteúdo existente”;
- Se Ranking > 100 E X > 30 Saída é igual a “Nada a se fazer”;

Contudo, essa técnica tem uma característica pouco desejada. Não é possível realizar o auto-aperfeiçoamento a medida que surgirem novos exemplos.

Procurou-se então transformar o algoritmo em uma árvore de decisão, de modo a usar a técnicas de aprendizado de máquina (como originalmente previsto) para realizar este auto-aperfeiçoamento. O algoritmo inicial já tem características de uma árvore de decisão e fazer a tradução deste para uma árvore foi uma tarefa trivial. A Figura 11 mostra o algoritmo traduzido para uma árvore de decisão.

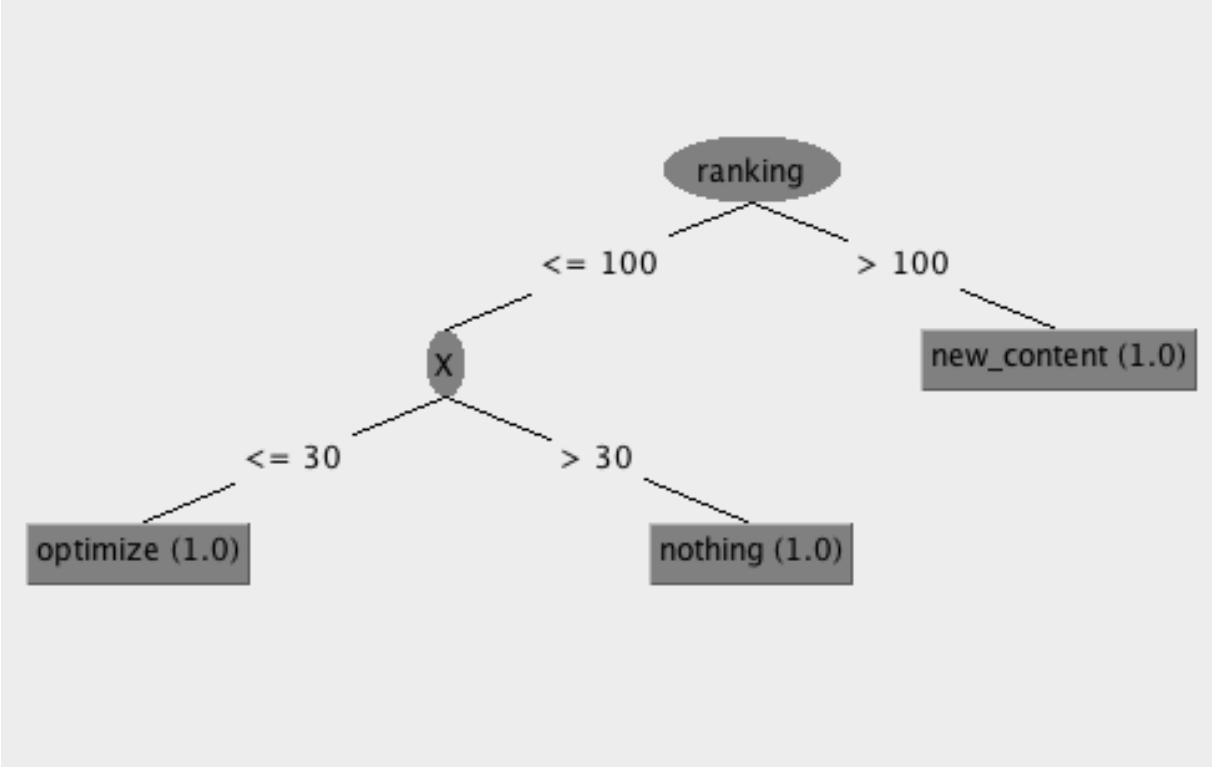

Figura 11 – Algoritmo inicial traduzido para uma árvore de decisão

Para que se possa aplicar as técnicas de aprendizagem de máquina de forma a possibilitar o auto-aperfeiçoamento, deve-se construir a árvore a partir de um conjunto de exemplos artificiais. Assim, a medida que surgirem novos exemplos, estes serão acrescentados ao banco de dados, e uma nova árvore possa será gerada a partir disso.

5.1.3.1: O conjunto de exemplos inicial

Para gerar o conjunto de exemplos que fosse capaz de gerar a árvore inicial, realizou-se um estudo de forma empírica. A estratégia adotada primeiramente foi a seguinte:

1. Gerar um número N de exemplos com dados aleatórios;
2. Classificar estes exemplos usando um script de computador que implementava o algoritmo no formato de regras (if, then ,else);
3. Salvar o resultado em formato de aquivo “.arff” que pode ser usado pela ferramenta Weka [13] ;
4. Utilizar a ferramenta para gerar a árvore ;
5. Comparar o resultado obtido com o esperado ;

Os resultados para N igual a 10, 100, 500 e 1000 podem ser vistos nas figuras Figura 12, Figura 14, Figura 15 e Figura 15 , respectivamente.

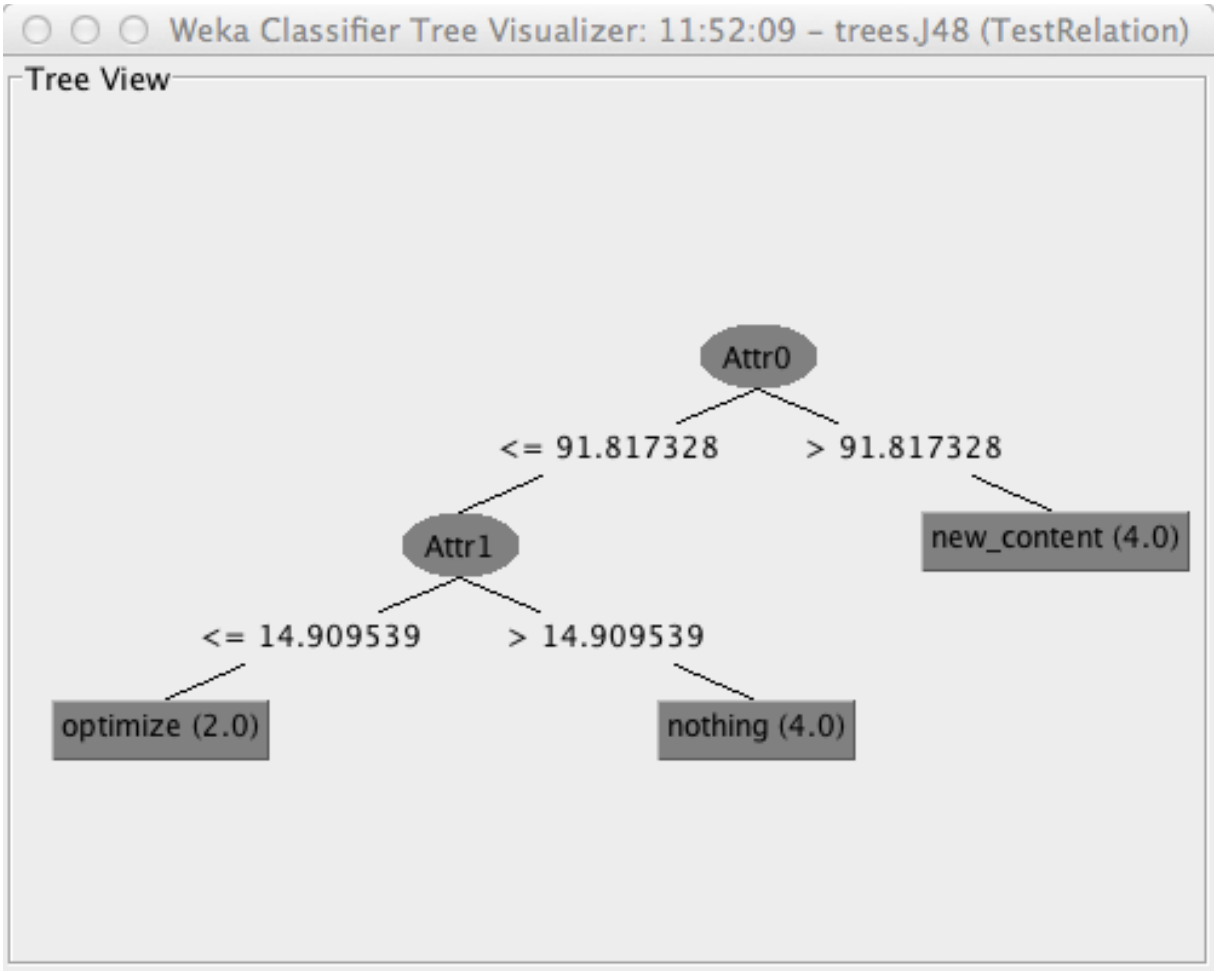

Figura 12 – Resultado da geração da árvore para N = 10

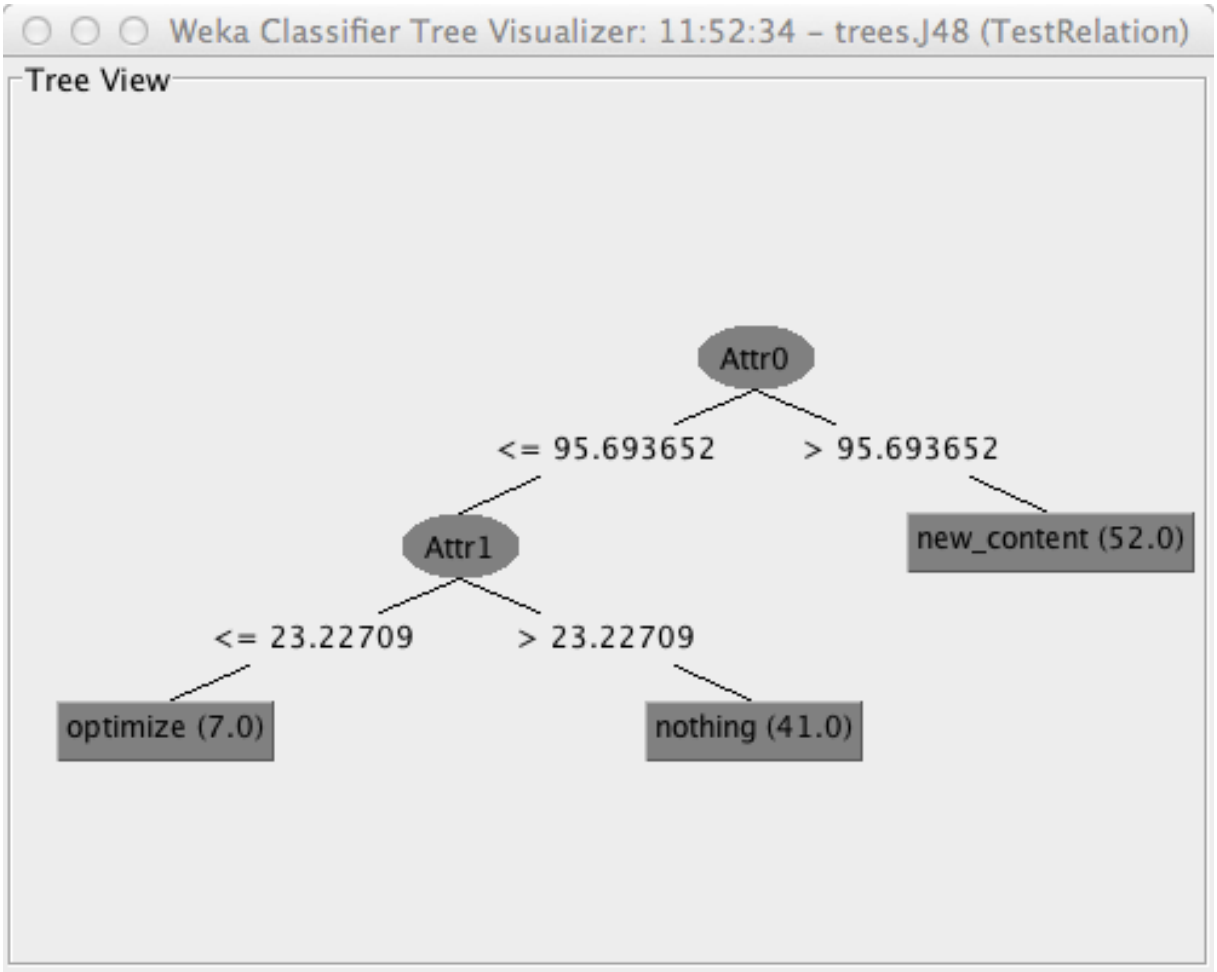

Figura 13 – Resultado da geração da árvore para N = 100

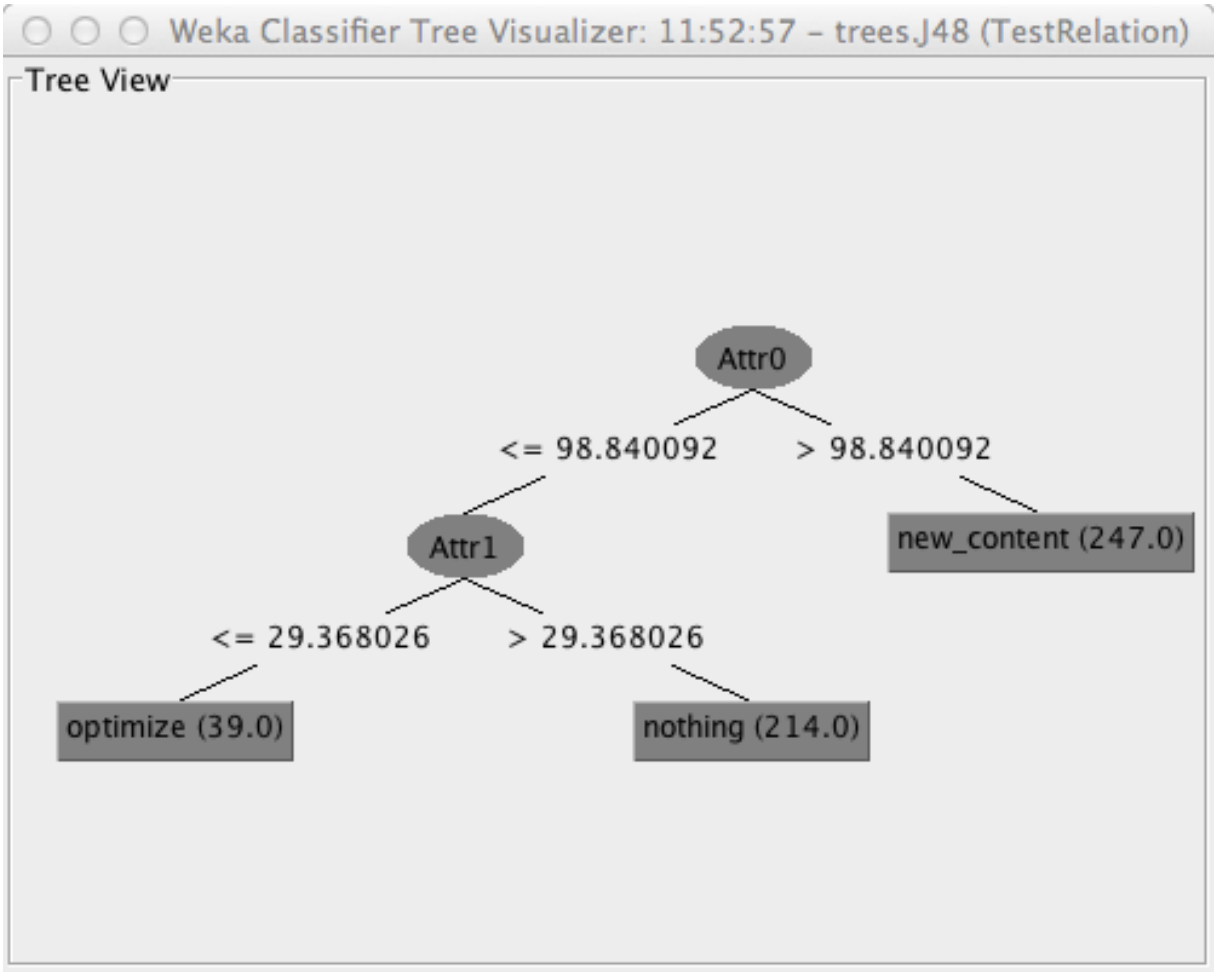

Figura 14 – Resultado da geração da árvore para N = 500

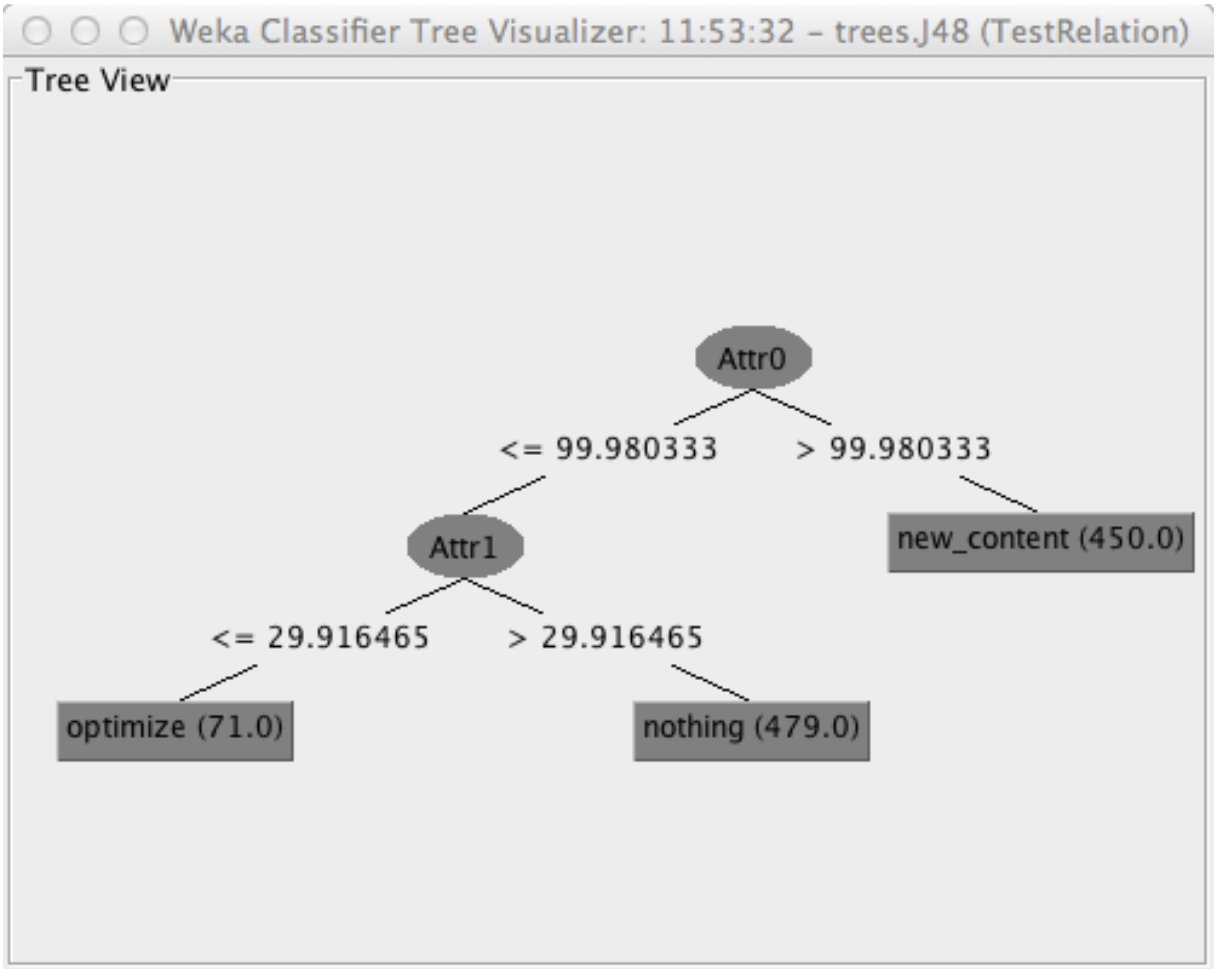

Figura 15 – Resultado da geração da árvore para N = 1000

Observa-se que quanto maior o valor de N, mais próximos chegamos da árvore desejada. Contudo, um grande conjunto de exemplos artificiais pode restringir o aprendizado da árvore, ou seja, quando tivermos novos exemplos, a árvore pouco mudará.

A estratégia adotada finalmente se baseia em uma mudança em um parâmetro do algoritmo C4.5 chamado número mínimo de instâncias[3] por folha.

[3] Esse parâmetro dita qual o número mínimo de instâncias que uma folha deve conter, caso não atinja o mínimo, essa folha é cortada e não aparece.

Ao adotar o valor 1 para este parâmetro podemos construir uma árvore com um exemplo para cada folha, neste caso três exemplos foram suficientes para gerar a árvore.

Começando com valores limítrofes as divisões de cada nodo da árvore e então refinando-os por tentativa e erro, chegou-se aos exemplos:

- Ranking = 101, Classe = “Novo Conteúdo”
- Ranking = 100, X = 30, Classe = “Otimiza Conteúdo”
- Ranking = 100, X = 31, Classe = “Nada a se fazer”

Para os valores faltantes não se atribuiu valor (*i.e*. foram tratados como desconhecidos). Como podemos ver na Figura 16 obteve-se com sucesso a árvore desejada.

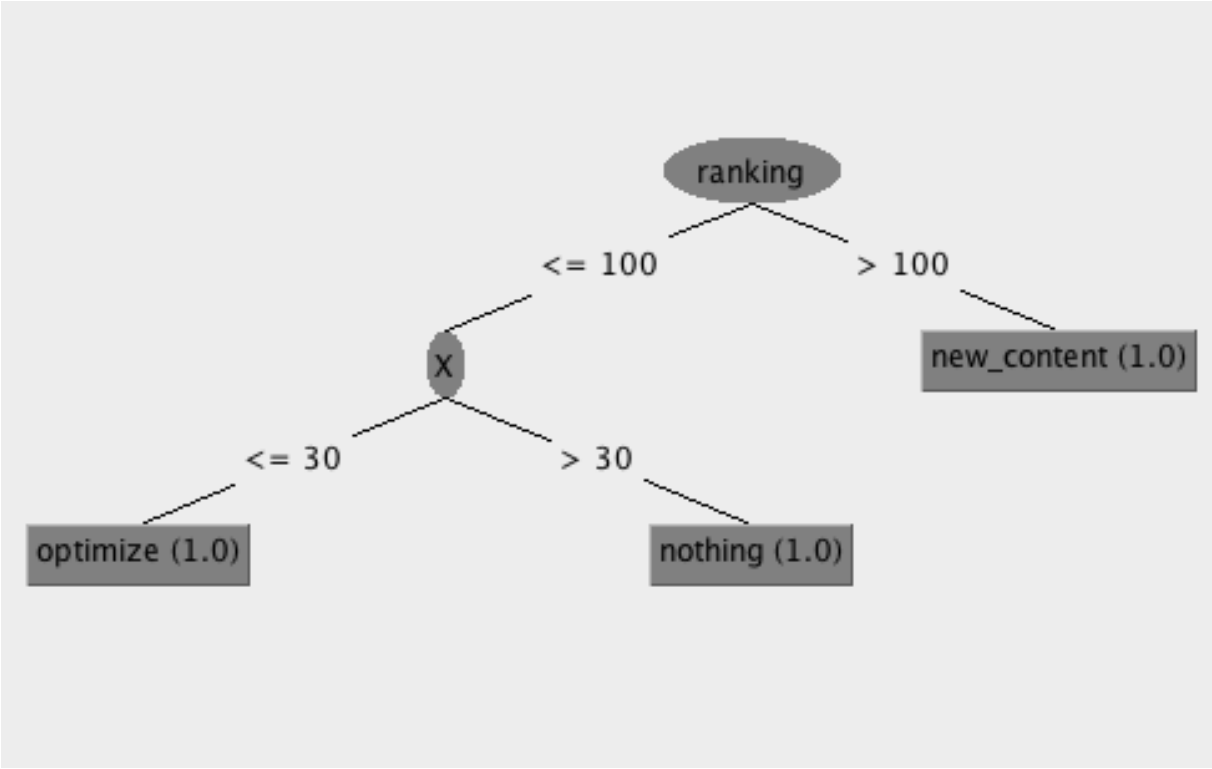

Figura 16 – Árvore inicial obtida com os exemplos artificiais

5.1.4: Classificação de exemplos

Para o desenvolvimento deste protótipo, optou-se por usar uma arquitetura cliente-servidor usando HTTP (Protocolo de transferência de Hipertexto) como protocolo de comunicação.

A entrada de dados para o sistema de classificação se dará por meio de uma requisição HTTP com método POST.

O sistema espera receber uma requisição com o cabeçalho:

Content-Type: application/json

Que como observa-se a seguir especifica que o corpo da requisição está no formato JSON.

O corpo da requisição deve estar no formato JSON.

O formato JSON [15]

JSON (JavaScript Object Notation - Notação de Objetos JavaScript) é uma formatação leve de troca de dados. Para seres humanos, é fácil de ler e escrever. Para máquinas, é fácil de interpretar e gerar. Está baseado em um subconjunto da linguagem de programação JavaScript, Standard ECMA-262 3a Edição -Dezembro - 1999. JSON é em formato texto e completamente independente de linguagem, pois usa convenções que são familiares às linguagens C e familiares, incluindo C++, C#, Java, JavaScript, Perl, Python e muitas outras. Estas propriedades fazem com que JSON seja um formato ideal de troca de dados.

JSON está constituído em duas estruturas:

- Uma coleção de pares nome/valor. Em várias linguagens, isto é caracterizado como um *object*, record, struct, dicionário, hash table, keyed list, ou arrays associativas.
- Uma lista ordenada de valores. Na maioria das linguagens, isto é caracterizado como uma *array*, vetor, lista ou sequência.

Estas são estruturas de dados universais. Virtualmente todas as linguagens de programação modernas as suportam, de uma forma ou de outra. É aceitavel que um formato de troca de dados que seja independente de linguagem de programação se baseie nestas estruturas.

A estrutura é formada por um dicionário com uma entrada, aonde a chave é "*keyword*" e o valor é um dicionário contendo entradas para cada um dos atributos necessários para a classificação. Caso algum atributo não esteja presente, assume-se que este é desconhecido. A Tabela 2 mostra um exemplo de corpo de requisição.

A resposta segue uma estrutura parecida, a diferença é que não existe mais o dicionário "pai", mas somente o dicionário com os atributos da palavra-chave. Além disso aparem alguns campos adicionais. O mais importante deles, tendo em vista a integração do sistema, é o "*keyword_class*" que representa a recomendação para esta palavra-chave. A Tabela 3 mostra um exemplo do corpo de uma resposta.

```
{
        "keyword":{
                "volume":"10",
                "cpc":"10",
                "competition":"10",
                "relevance":"10",
                "conversion_rate":"10",
                "difficulty":"10",
                "ranking":"10",
                "domain_authority":"10",
                "page_authority":"10",
                "page_grade":"10",
                "social_signals":"10",
        }
}
```

Tabela 2 – Formato de requisição

```
{
"acted_upon":false,
"auxiliary":10.0,
"competition":10.0,
"conversion_rate":10.0,
"cpc":10.0,
"created_at":"2012-07-09T01:51:58Z",
"date_classified":"2012-07-09T01:51:59Z",
"difficulty":10.0,
"domain_authority":10.0,
"id":28,
"keyword_class":"optimize",
"page_authority":10.0,
"page_grade":10.0,
"ranking":10.0,
"relevance":10.0,
"result":null,
"right_class":null,
"social_signals":10.0,
"updated_at":"2012-07-09T01:51:59Z",
"volume":10.0
}
```


Tabela 3 – Formato da resposta

5.1.5: Auto-aperfeiçoamento

A medida que novas classificações são feitas e avaliadas, surgem novas oportunidades de aprendizado. Contudo, com poucos exemplos, não existe a possibilidade de avaliação dos resultados de uma maneira estatisticamente significante. Além disso, os atributos do algoritmo de aprendizado ainda não foram testados na prática.

Logo, não é interessante que o sistema tenha a capacidade de se auto-modificar em um primeiro momento, pois corremos o risco de termos uma árvore pior que a anterior.

Dado este fato, optou-se por adicionar um gatilho manual no ciclo de auto aperfeiçoamento. Dessa maneira a nova árvore pode ser reavaliada por um ser humano antes de substituir a antiga.

A geração da árvore ocorre sempre no inicializar do sistema. A partir de um arquivo que contêm exemplos em um formato específico, o algoritmo C4.5 cria a árvore que seja utilizada pelo resto da vida deste processo.

O auto-aperfeiçoamento ocorre ao mudarmos este arquivo de exemplos, de modo com que o algoritmo gere uma árvore possivelmente melhor que a anterior que foi gerada com um conjunto de exemplos diferente.

O processo acontece da seguinte maneira:

1. O comando de início do processo é enviado ao sistema de maneira *off-line.*
2. O módulo classificador então consulta o banco de dados. As palavras-chave escolhidas são de dois tipos. O primeiro tipo são as palavras-chave para as quais o sistema “acertou” a classificação, ou seja, as palavras-chave que foram avaliadas e tiveram um resultado positivo. O segundo tipo são as palavras que tiveram um resultado negativo mas receberam a classificação alternativa do avaliador e portanto podem ser usadas para o aprendizado.
3. Os resultados da consulta são então traduzidos para o formato Attribute-Relation File Format (“.arff”) e salvo em um arquivo temporário.

4. Esse arquivo é salvo e utiliza-se então o sistema Weka de forma isolada para gerar a visualização da árvore para que que um especialista o analise. A partir do resultado desta análise, substitui-se ou mantêm-se o arquivo anterior.
5. Esse arquivo é lido pelo sistema quando inicia (passa para modo *online*) e a partir dele é gerada a árvore que será usada para a classificação.

5.2: Banco de dados

O banco de dados não tem requisitos fortes e por simplicidade e conveniência, optou-se por usar um banco de dados relacional simples.

O modelo usado foi muito simples e consiste em apenas um modelo (*Keyword*) e campos para cada um dos atributos.

Além disso, foram adicionados os seguintes campos:

- acted_upon – campo booleano que representa se o usuário executou a recomendação para a palavra-chave
- date_classified – campo de data que representa a data de classificação da palavra-chave, serve também para indicar que a palavra foi classificada
- result – campo de texto que representa o resultado da ação realizada, ou seja, se os esforços deram resultados positivos, indefinidos ou negativos. Está limitado a assumir os valores “positive”, “negative” e “undefined”.
- right_class - A árvore de decisão tem um processo de aprendizado “direto”, ou seja, os exemplos devem dizer para a árvore qual a escolhe deve ser feita.

A Figura 17 mostra a visualização do modelo.

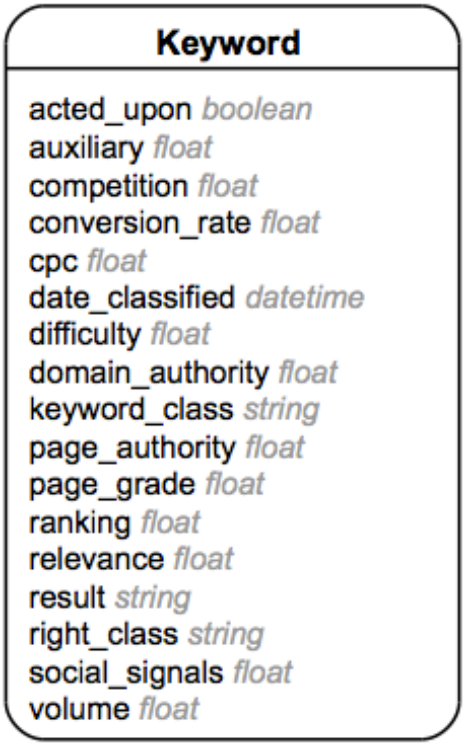

Figura 17 – Visualização da modelagem do banco de dados.

5.3: Avaliador

Para a avaliação das palavras-chave, foi implementado uma interface *web* para o sistema. O especialista usando um navegador *web*, acessa um endereço previamente estabelecido e então tem acesso as funções administrativas.

A interface é protegida por um esquema de segurança que utiliza nome de usuário e senha. Cada novo colaborador deve receber essa informações para poder utilizar o sistema.

Quando é acessado, o sistema apresenta ao usuário três grupos de palavras-chave.

- Palavras que foram recentemente classificadas e ainda não foram avaliadas. São caracterizadas por terem o campo “acted_upon” nulo. Para essa palavras o usuário deve marcar aquelas que estão sendo implementadas de fato para que passem ao segundo grupo.
- Palavras que estão esperando os resultados dos esforços. São caracterizadas pelos campos “acted_upon” verdadeiro e “result” nulo. Quando houver uma conclusão para o resultado da recomendação

para a palavra-chave, o usuário deverá informar qual foi para que essa passe para o último grupo.

- Palavras que tem resultado negativo e tem o campo “right_class” nulo. Essas palavras devem ser atualizadas com a recomendação que o especialista daria para esta palavra-chave.

Com essas intervenções, o sistema atualiza o banco de dados para que o módulo classificador possa se atualizar.

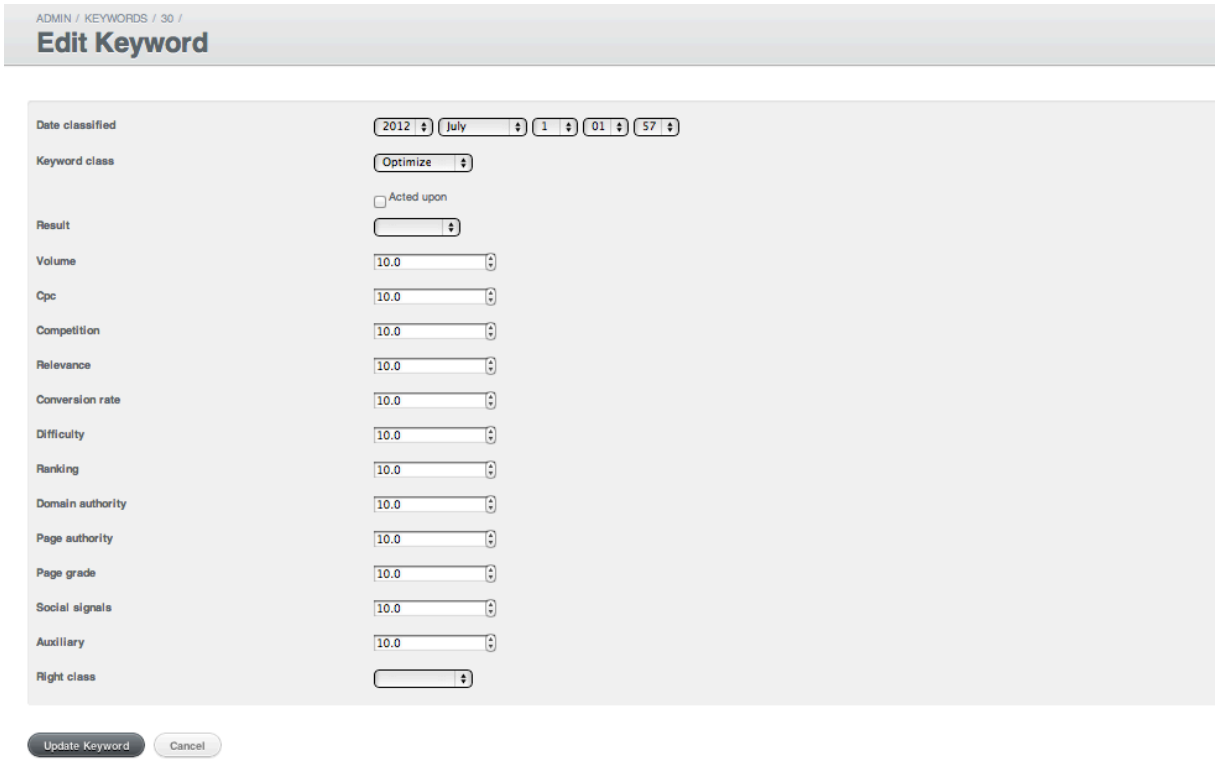

Figura 18 – Captura de tela da interface de avaliação

Capítulo 6: Resultados

Foram escolhidas três palavras-chave de um dos clientes da Resultados Digitais para testar a solução proposta. Para manter o sigilo, as palavras-chave não foram reveladas, e o trabalho refere-se a elas como palavra-chave 1, palavra-chave 2 e palavra-chave 3. Os dados de cada palavra-chave estão na Tabela 4.

Palavra-chave	1	2	3
Relevância	0.5	1	1
Taxa de conversão	N/A	N/A	N/A
Volume	165000	390	260
CPC	2.13	3.27	3.21
Concorrência	0.16	0.55	0.53
Dificuldade	0.42	0.28	0.35
Posição	101	2	11
Autoridade do Domínio	36	36	36
Autoridade da Página	N/A	44	30
Nota estrutural	N/A	75	58
Sinais Sociais	N/A	80	35
X	N/A	64.1	29.1

Tabela 4 – Dados das palavras-chave usadas na fase de teste

A classificação para cada palavra-chave foi a seguinte:

- Palavra-chave 1 : Criar novo conteúdo
- Palavra-chave 2 : Nenhuma ação a ser tomada
- Palavra-chave 3: Otimizar o conteúdo existente.

Na fase de avaliação, em consulta aos especialistas, as classificações foram julgadas como corretas. As recomendações porém não puderam ser implementadas, isto se deve a dificuldade de adequação dessas ações dentro da agenda de marketing da empresa. Dito isto, para fins de teste, assume-se que os resultados das ações são positivos, ou seja, o módulo avaliador apontou resultados positivos para as palavras-chave.

Ao realimentar o sistema com as novas classificações, observa-se que a árvore resultante é idêntica a atual. Isso é esperado, dado que nenhuma palavra-chave trouxe novas informações, ou seja, houve apenas o reforço das hipóteses da árvore. Isso sugere que o algoritmo inicial da árvore tem valor, contudo, como não temos um número razoável de exemplos, essa conclusão não tem expressividade estatística.

Como todos os exemplos foram julgados corretos pelos especialistas, não tivemos a oportunidade de observar o comportamento de aprendizado da árvore quando esta se depara com um caso aonde errou. Com esse intuito, foi realizado o seguinte experimento:

1. O sistema foi reinicializado ao seus valores iniciais
2. As palavras-chave de teste foram reinseridas no sistema
3. No modulo de avaliação, avaliou-se as palavras 1 e 3 normalmente
4. A palavra 2 no entanto recebeu a classificação artificial como se não tivesse dado resultados positivos, e recebeu como classe correta (*right_class*) Otimizar
5. O sistema foi realimentado de modo a gerar uma nova árvore.

O resultado pode ser visto na Figura 19. Observa-se que o nodo relativo a variável X foi removido. Isso aconteceu devido ao comportamento de poda do algoritmo C4.5. De fato, ao repetirmos o experimento com a opção de poda desligada, a árvore resultante (Figura 20) é maior, tendo um ramo aonde antes existia só uma folha.

Como não existem exemplos suficientes para subdividirmos o conjunto de treinamento em outro de teste, não podemos afirmar qual dessas árvores tem o melhor desempenho. Disto, surge uma importante conclusão, é importante testar e

avaliar os parâmetros de geração da árvore, pois estes influenciam muito no formato, e portanto no processo decisório da árvore.

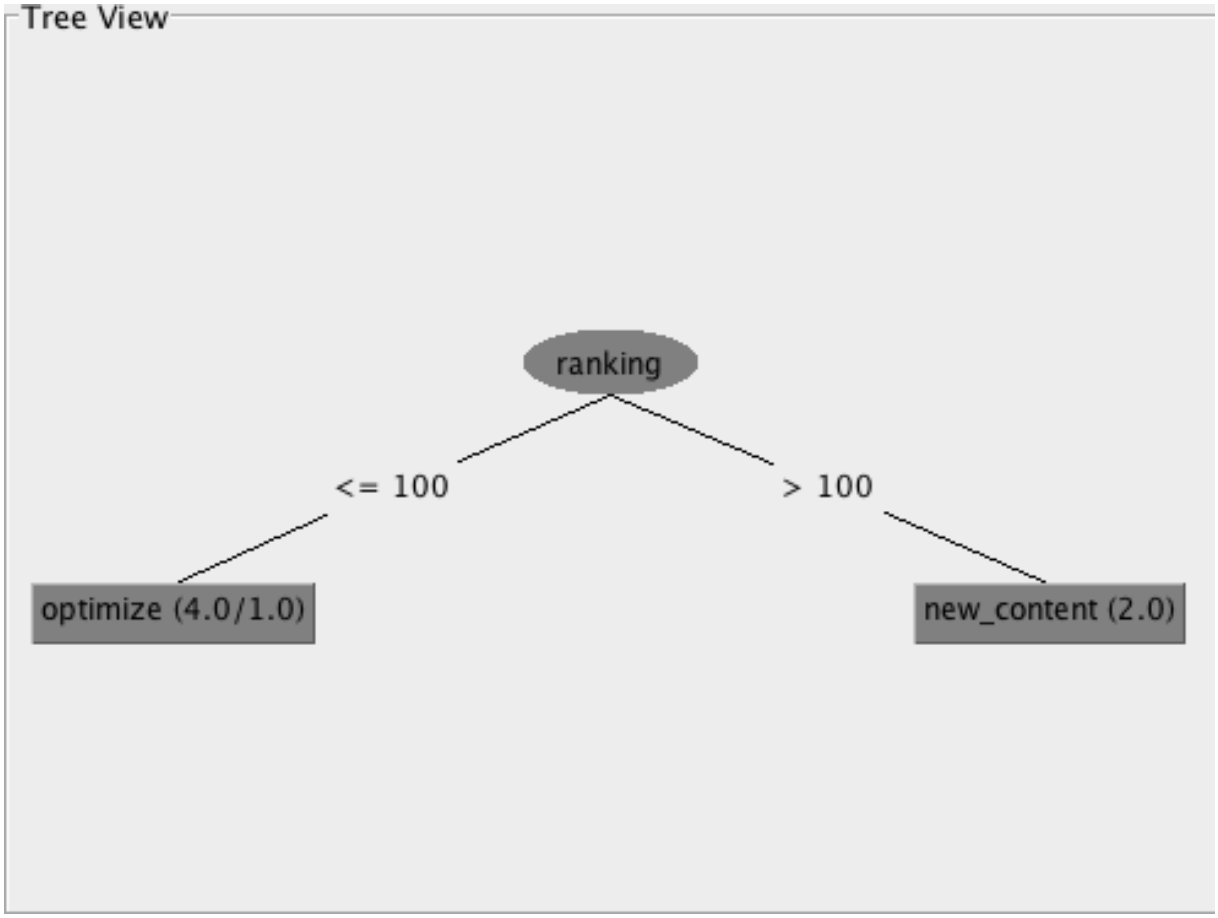

Figura 19 – Árvore resultante do experimento feito com a avaliação artificial

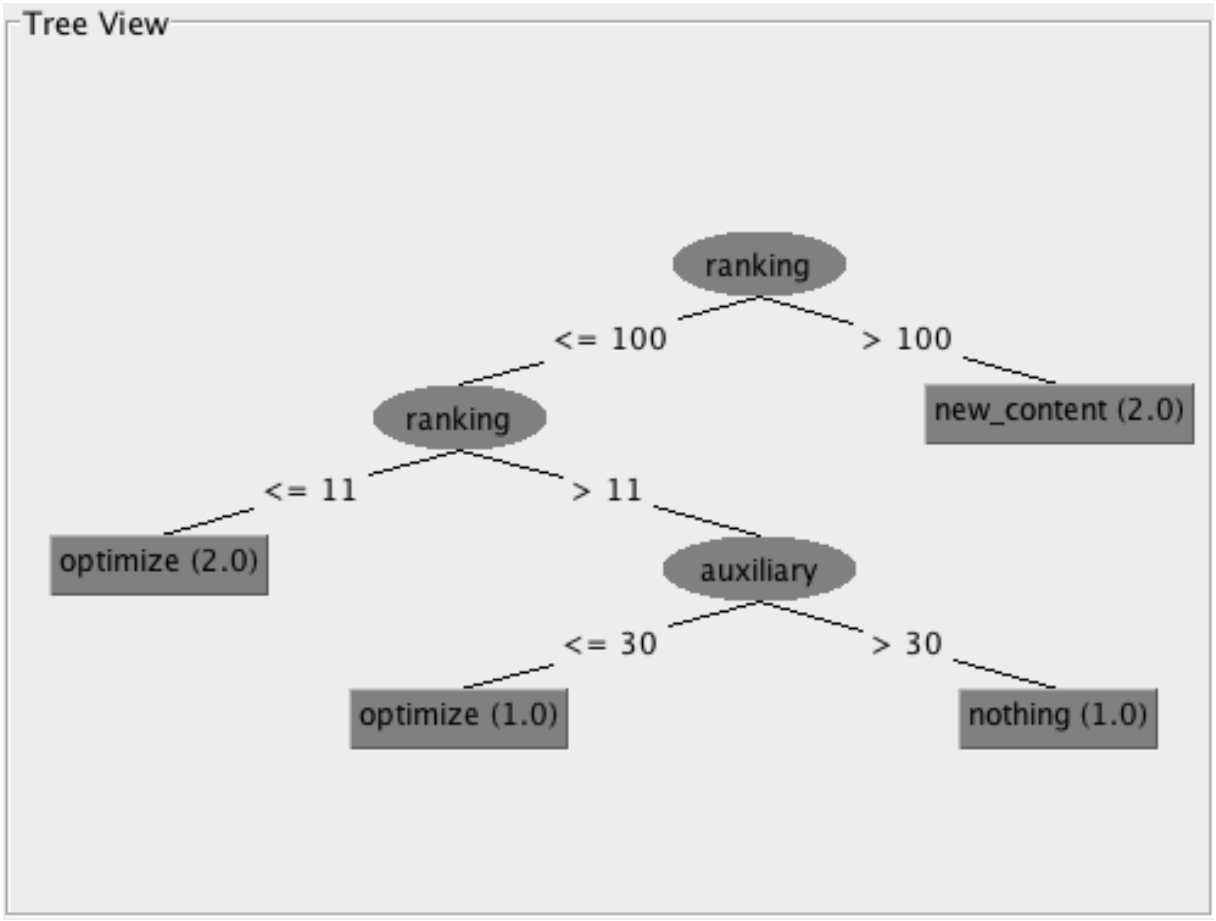

Figura 20 – Árvore resultante com opção de poda desligada

Capítulo 7: Conclusão e perspectiva de trabalho futuros

Apesar dos resultados dos testes com palavras-chave não terem expressividade estatística, o principal problema foi resolvido com sucesso. O sistema está apto a dar recomendações baseado em um modelo definido por especialistas humanos. Do ponto de vista da empresa, isto traz muito valor, pois é um diferencial que torna o produto muito mais atrativo. Houve também aprendizado no que diz respeito a forma com que a solução atual evolui, e um trabalho futuro deverá ser realizado para avaliar os parâmetros do aprendizado a medida que a árvore cresce.

Além disso foram dados passos muito importantes para a construção de um algoritmo que tem o potencial de ser muito eficiente. Possivelmente a maior contribuição do trabalho é a possibilidade de registro de todas as recomendações e seus respectivos resultados. A partir do momento que estes dados adquirirem volume expressivo, as técnicas de Inteligência Artificial podem ser avaliadas e aplicadas como foi originalmente proposto.

Como próximo passo temos a integração com o sistema RDStation.

O sistema é passível de muitas melhorias. O módulo avaliador deverá ser eventualmente automatizado. Isso eliminará o ser humano de mais uma parte do processo e permitirá melhor escalabilidade. A ferramenta WEKA, não possui uma licença comercial, logo, deverá ser substituída. Elimina-se assim também, a dependência do arquivo ARFF.

A medida que a experiência, do software e da empresa aumenta, as recomendações poderão ser melhoradas, fornecendo estratégias cada vez mais específicas.

Bibliografia

[1] Halligan, Brian, e Darmesh Shah. *Inbound Marketing: Get Found Using Google, Social Media, and Blogs.* Wiley, 2009.

[2] D. M. Scott, “The New Rules of Marketing and PR”. Wiley, 2011

[3] S. Gregg, “Survey: Search Now Top Resource For Local Information”, 2007 disponível em <http://searchengineland.com/survey-search-now-top-resource-for-local-information-12396>

[4] A. Siqueira, “Como escolher as melhores palavras-chave para brigar pelos resultados no Google” , http://resultadosdigitais.com.br/blog/como-escolher-as-melhores-palavras-chave-para-brigar-pelos-resultados-no-google/ acessado em 15/11/2011

[5] Pan, B., Hembrooke, H., Joachims, T., Lorigo, L., Gay, G., and Granka, L. (2007). In Google we trust: Users' decisions on rank, position, and relevance. *Journal of Computer-Mediated Communication*, *12*(3), article 3. http://jcmc.indiana.edu/vol12/issue3/pan.html

[6] E. Eric , “The Art of SEO”, Oreilly Media, 2009

[7] John I. Jerkovic, “SEO Warrior”, O'Reilly Media, 2009

[8] http://www.netmarketshare.com/search-engine-market-share.aspx?qprid=4&qptimeframe=Y&qpsp=2011

[9] http://www.seomoz.org/article/search-ranking-factors, acessado em 29/05/2012

[10] Wei-Yin Loh, “Classification and regression trees” , Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,Vol 1,Nº 1 2011

[11] Mitchell, Tom M., “Machine Learning”, McGraw-Hill, 1997

[12] Ajuda Google Adwords (http://support.google.com/adwords/bin/answer.py?hl=pt-BR&answer=2497836&from=6100&rd=1) Acessado em 3/07/2012

[13] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); “The WEKA Data Mining Software: An Update; SIGKDD Explorations”, Volume 11, Issue 1.

[14] https://github.com/rails/rails, Acessado em 7/07/2012

[15] http://www.json.org/json-pt.html, Acessado em 7/07/2012

[16] http://www.seomoz.org/blog/facebook-twitters-influence-google-search-rankings , Acessado em 7/07/2012

[17] Estudo realizado por empresa SlingshotSEO, disponível em http://www.slingshotseo.com/wp-content/uploads/2011/07/Google-vs-Bing-CTR-Study-2012.pdf

[18] Resutlados Digitais - http://resultadosdigitais.com.br/

[19] RDStation http://www.rdstation.com.br/

[20] Google AdWords http://adwords.google.com.br/

[21] Ruby http://www.ruby-lang.org/

[22] Ruby on Rails http://rubyonrails.org/

[23] Riehle, Dirk (2000), *Framework Design: A Role Modeling Approach*, Swiss Federal Institute of Technology

[24] Quinlan, R., “C4.5: Programs for machine learning”, Morgan Kaufmann Publishers, 1993