



# Uma Arquitetura de Ambiente de Aprendizagem para aferir o conhecimento do aluno

**Leandro da Silva Foly**  
Faculdade Redentor  
Itaperuna – Rio de Janeiro  
[lefolly@gmail.com](mailto:lefolly@gmail.com)

**Geórgia Rodrigues Gomes**  
Universidade Cândido Mendes  
Campos dos Goytacazes – Rio de Janeiro  
[georgia@ucam-campos.br](mailto:georgia@ucam-campos.br)

**Resumo** *Com a evolução das pesquisas na área de Educação à distância (EAD), onde predominam as ferramentas tecnológicas mediadoras entre professores e alunos, a utilização de mineração de texto, juntamente com estruturas de Objetos de Aprendizagem (OAs) em uma aplicação de E-learning que ajude a aferir o conhecimento do aluno, é de grande importância, para que a mesma sirva de apoio ao processo de aprendizagem. Este artigo descreve uma arquitetura para estruturar o conhecimento do professor na forma de Objetos de Aprendizagem (OAs) e baseado na mesma, aferir o conhecimento do aluno. Tal arquitetura utiliza técnicas de Inteligência Computacional, em especial algoritmos de Mineração de Texto, para analisar e comparar textos dispostos em linguagem natural tanto por professores quanto por alunos em um Sistema de Gerência de Aprendizagem (LMS), sugerindo uma nota para o aluno. Esta arquitetura é demonstrada através de um protótipo e estudos de casos.*

**Palavras-Chave:** Educação à Distância, Objetos de Aprendizagem, Mineração de Texto

**Abstract** *With the research's evolution in the Distance Education (EAD) area, dominated by the technological mediator tools between teachers and students, the use of text mining, along with structures Learning Objects (LOs) in an E-learning application to help measure student knowledge, is of great importance, so that it can support the learning process. This article describes an architecture for structuring the professor's knowledge in a Learning Object (LO) approach, and based on it, measure student knowledge. This architecture uses Computational Intelligence techniques, in particular Text Mining algorithms to analyze and compare texts arranged in natural language by both teachers and students in a Learning Management System (LMS), suggesting a grade to the student. This architecture is demonstrated through a prototype and case studies.*

**Keywords:** Distance Education environment, Learning Objects, Text Mining.

## 1 Introdução

Existem vários contextos em que a tecnologia tem sido utilizada na educação: o primeiro que utiliza a tecnologia como apoio à sala de aula, e o segundo, cuja Internet é a principal articuladora. Este último contexto, cada vez mais utilizado, permite que se tenha uma verdadeira extensão da sala de aula presencial.

Essa evolução deu origem a um novo modelo de ensino chamado de Educação a Distância, onde as novas ferramentas têm papel primordial no processo, pois fazem o papel de mediador no processo de ensino-aprendizagem, além de permitir que o aluno desenvolva seus estudos de forma autônoma.

Nesse contexto, tem-se os Sistemas de Gerência de Aprendizagem (LMS - *Learning Management Systems*) que são softwares que automatizam a administração de eventos de ensino-aprendizagem. LMS's são a espinha dorsal de um projeto de *E-learning*, pois centralizam as funções e os processos de aprendizagem, registrando os passos de cada usuário, estatísticas de desempenho e cursos oferecidos[1].

Um dos conceitos pesquisados por vários autores no contexto de *E-learning* é o de Objetos de Aprendizagem (OAs - *Objetos de Aprendizagem*). Trata-se de materiais educacionais de apoio ao processo de ensino aprendizagem. A ideia é que eles sejam estruturados em fragmentos interligados, e são melhor aproveitados quando organizados em um LMS[2].

Ao trabalhar com um LMS deve-se traçar o modelo de tarefas percorrido pelo aprendiz (aluno) até o conteúdo, bem como a estrutura disponível no LMS para acesso ao conteúdo. Ele não será apenas utilizado por aprendizes que irão acessar o curso e utilizar suas ferramentas de interação, mas também será utilizado por tutores e administradores que estarão criando novos cursos e escolhendo as ferramentas desejadas para o tipo específico de projeto.

Os LMS's atuais consistem, na essência, em extensões da sala de aula, sejam porta-arquivos, blogs ou fóruns de discussão. Porém, se estas soluções utilizassem inteligência computacional, tal como, registrar de forma estruturada o conhecimento do professor, para que baseado na mesma, o aluno pudesse aferir de forma inteligente e automática seu conhecimento, seriam um grande apoio ao processo de ensino/aprendizagem.

De forma mais específica, há no domínio da Inteligência Computacional uma sub-área a ser explorada, que consiste em utilizar algoritmos de Mineração de Texto aplicados ao conteúdo inserido nos LMS. A Mineração de

Texto consiste em um conjunto de técnicas e processos aplicados para descobrir conhecimento inovador em textos. As técnicas de Mineração de Texto procuram extrair o conhecimento útil de textos não estruturados e semi-estruturados, ou seja, artigos científicos, textos simples, documentos de organizações, etc. Através da análise de textos é possível a descoberta de conceitos, classificações automatizadas e sumarizações para documentos não estruturados[3].

Este trabalho objetiva apresentar uma arquitetura para aferir o conhecimento do aluno de forma automática, apoiando assim, a Educação a Distância. Esta arquitetura possibilita ao professor criar OAs baseado na metodologia da CISCO, e utilizar técnicas de Mineração de Texto para comparar o conhecimento dos alunos com os OAs criados pelos professores, aferindo assim o conhecimento do mesmo de forma automática.

Alguns trabalhos abordando o uso de Mineração de Texto em ambientes de *E-Learning* foram encontrados, tais como em [4], [5] e [6], mas nenhum deles têm como objetivo aferir o conhecimento do aluno e não possuem proposta de estruturação do conhecimento através de OAs.

O restante do artigo está organizado conforme a seguir. A seção 2 apresenta a arquitetura proposta e as tecnologias utilizadas na mesma, a seção 3 descreve a implementação do protótipo construído baseado na arquitetura proposta, a seção 4 apresenta uma visão geral do estudo de caso realizado e resultados obtidos e finalmente a seção 5 contém a conclusão do trabalho bem como algumas sugestões de trabalhos futuros.

## 2. Objetos de Aprendizagem

Objetos de aprendizagem são materiais educacionais que servem para apoiar o processo de ensino-aprendizagem. Podem ser definidos como qualquer recurso, suplementar ao processo de aprendizagem, que pode ser reusado para apoiar a aprendizagem. A ideia básica é a de que os objetos sejam como blocos com os quais será construído o contexto de aprendizagem [7].

Objetos de Aprendizagem são mais eficientemente aproveitados quando são descritos através de metadados e armazenados em um repositório integrável a um LMS (*Learning Management System*).

Algumas características desejáveis de um Objeto de Aprendizagem são propostas em [8]: flexibilidade, facilidade para atualização, customização, interoperabilidade, aumento do valor de um conhecimento, indexação e pro-

cura. O resultado do desmembramento de informações é um conjunto de módulos independentes que são acessados através do LMS.

Neste contexto, o modelo de objetos de aprendizagem reutilizáveis desenvolvido pela CISCO Systems [9] traz uma arquitetura pré-definida para o armazenamento e acesso a OAs, que, alinhada a uma proposta educacional planejada, organiza e facilita o acesso aos fragmentos de informação requeridos no processo de ensino-aprendizagem.

Esta constitui de uma hierarquia de dois níveis para o desenvolvimento dos conteúdos reutilizáveis: RLOs (Objetos de Aprendizagem Reutilizáveis) e RIOS (Objetos de Informação Reutilizáveis), que são estruturados em 3 camadas: Itens de Conteúdo (*Content Items*), Itens de Prática (*Practice Items*) e Itens de Avaliação (*Assessment Items*). A camada *Itens de Conteúdo*, que varia conforme a natureza do RIO, é dividida segundo os cinco tipos de informação propostos em [10]: conceito, fato, procedimento, processo e princípio, conforme é visto na Figura 1.

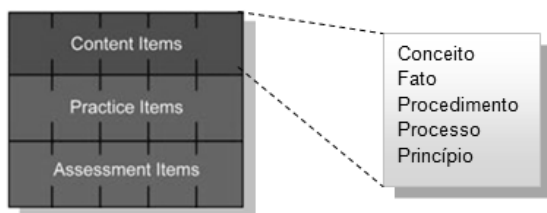


Figura 1: Itens de Conteúdo e sua composição

Uma das subdivisões da camada *Itens de Conteúdo* é o chamado *Conceito*. De acordo com a metodologia CISCO, um Conceito RIO é usado quando precisamos explicar um grupo de objetos, símbolos, ideias ou eventos que:

- São designados por uma única palavra ou termo;
- Compartilham uma característica comum;
- As variações de suas características são irrelevantes.

O Conceito, por sua vez, é composto dos sub-itens *Introdução*, *Definição*, *Fato*, *Exemplo*, *Contra-Exemplo* e *Analogia*, apresentados na figura 2:

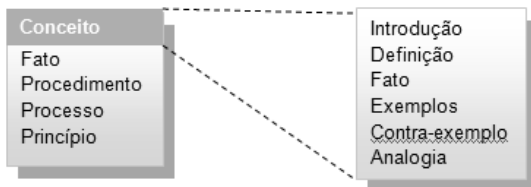


Figura 2: Conteúdo do item Conceito

A camada Conceito e sua estrutura serve de base e é

estendida no modelo desenvolvido neste trabalho.

### 3. Arquitetura Proposta

A arquitetura proposta em [11] é representada na Figura 3. Esta compõe-se de três camadas: camada de interface, camada de inteligência e camada de dados.

A camada de interface é onde professor e aluno interagem com o LMS, cada qual em uma interface diferente. Porém, ambos utilizam como método de entrada textos em linguagem natural, mediados por um ambiente Web.

Na interface professor, é criada a base de conhecimento em forma de OAs, necessária ao seu domínio de trabalho, de forma que sua experiência no domínio específico seja formalizada e armazenada no sistema. Além dos OAs, o professor também pode inserir perguntas no ambiente.

A interface aluno permite que sejam respondidas as perguntas previamente preparadas pelo professor especialista; o aluno deve dissertar sobre o tema arguido em um espaço de texto limitado, como se respondesse a questões de um exercício ou uma avaliação. O aluno também consegue obter sua nota sugerida em cada questão, na forma de um *feedback* com o retorno da aferição automática.

Na camada de inteligência utilizou-se um conjunto de algoritmos advindos do estudo de Mineração de Texto, de forma a criar uma massa de dados mais “precisa” para a comparação dos dois conjuntos textuais trabalhados no sistema (texto do professor / texto do aluno).

Esta camada analisa o texto fornecido pelo aluno e compara-o com a base de OAs do professor. De acordo com o percentual de aproximação das respostas do aluno, comparada com os OAs, é então, enviado para a camada de interface do aluno o resultado da aferição automática, com a sugestão de nota para cada questão respondida.

Uma série de passos foram traçados para que se obtivesse um bom resultado, sendo que, algoritmos diferentes foram utilizados para cada passo. Os passos dessa camada são apresentados na Figura 4.

- **Tokenização:** divisão do texto em pequenos fragmentos chamados *tokens*; O algoritmo identifica espaços e sinais de pontuação como vírgula e gerando um arquivo com um termo ou *token* por linha.
- **Filtragem de Stopwords:** eliminação de termos sem significado semântico, como artigos, preposições, conjunções, pronomes, etc. O objetivo é eliminar palavras que não são representativas ao documento, consequentemente diminuindo o número de palavras a serem analisadas [12]. O resultado é um arquivo

texto contendo os *tokens* relevantes à análise.

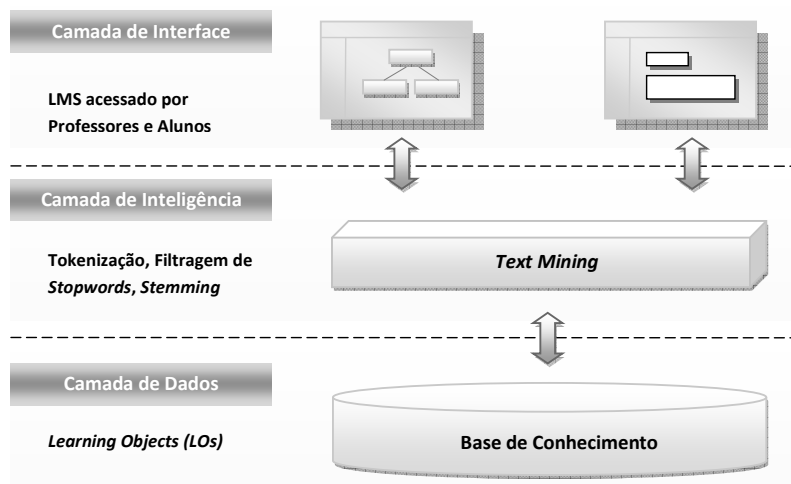


Figura 3: Arquitetura de três camadas proposta.

- Análise de Sinonímia; neste passo é feita a busca por sinônimos na base de conhecimentos, para eliminar inconsistências. O resultado é um arquivo com termos únicos, ou sem sinônimos que levariam a análise diferenciada.
- *Stemming*: processo de reduzir todas as palavras ao mesmo stem, por meio da retirada dos afixos (sufixos e prefixos) da palavra, permanecendo apenas a raiz [13]. O resultado é um arquivo contendo *stems*, ou seja, o radical de cada termo. O objetivo deste passo é aumentar a precisão na comparação, já que termos correlatos podem possuir o mesmo *stem*.
- Definição de um peso para cada termo, baseado em sua frequência, e geração final de um arquivo com termos únicos e suas frequências.

A comparação é feita, então, entre a base de conhecimentos do professor e o texto digitado pelo aluno, mas em suas versões compactas produzidas pelos passos acima, o que, acredita-se potencializar o grau de precisão e a qualidade do processo de comparação, já que pode-se ter variâncias possíveis da forma de se expressar uma idéia,

dentro de um mesmo domínio.

Além do processamento dos algoritmos de Mineração de Texto, a camada de inteligência também realiza o pós-processamento, ou seja, implementa uma métrica de avaliação de resultados. A metodologia utilizada neste trabalho é caracterizada como a equação de Abrangência, pois leva em consideração o que deve ser descartado no processo.

De acordo com a arquitetura proposta no trabalho, é utilizada a equação de Abrangência, citada por [3], para fazer a comparação entre as massas textuais entradas no sistema, e conseqüentemente, realizar a aferição dos conhecimentos do aluno:

A = Massa de Dados do Aluno  
 B = Massa de Dados do Professor

$$\text{Abrangência (A)} = \frac{\text{N}^\circ \text{ de itens certos recuperados em A}}{\text{Número total de itens certos em B}}$$

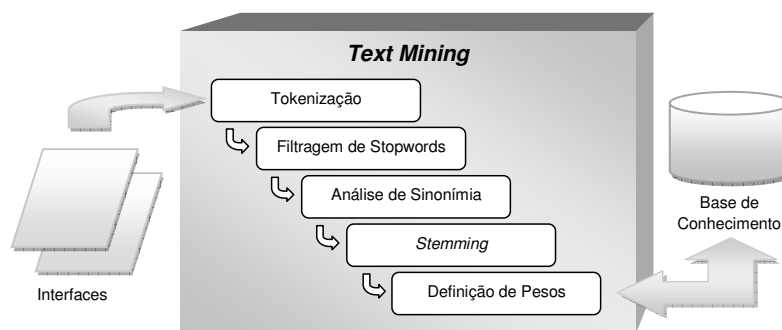


Figura 4: Passos realizados pela camada de inteligência

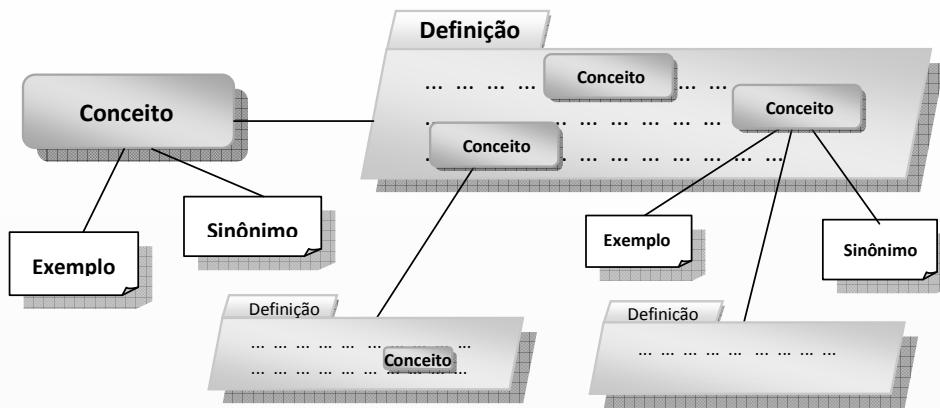


Figura 5: Relação entre Conceitos, Definições, Exemplos e Sinônimos

Onde compara-se A, que é a massa gerada da análise textual do aluno (Mineração de Texto aplicada à entrada textual do aluno) com B, que é a massa gerada pela análise textual do professor (Mineração de Texto aplicada aos campos dos OAs cadastrados pelo professor).

A equação de abrangência gera como resultado um percentual que é convertido na nota sugerida ao aluno, sendo enviada à camada de interface e apresentada ao mesmo.

Na camada de dados, definiu-se uma estrutura de dados para armazenar os OAs do professor, que foi definida com base na arquitetura da CISCO Systems [9]. Na Figura 5 é apresentada a estrutura de armazenamento interna do sub-item *Conceito* do OA.

O OA possui um conceito, cada conceito possui uma definição relacionada, além de um ou mais exemplos e/ou sinônimos. Cada definição, porém, pode conter, em sua descrição textual, outros conceitos (pertencentes a outros OAs), e estes possuem suas respectivas definições e estão relacionados a outros exemplos/sinônimos, constituindo, assim, uma árvore de conhecimento dinâmica, que será expandida ou alterada de acordo com as informações inseridas pelo professor.

#### 4 Implementação do Protótipo

O protótipo funciona como um LMS (*Learning Management System*), em um ambiente Web a ser acessado via navegador. Por esse motivo, adota a arquitetura cliente-servidor, onde as requisições são enviadas pela Internet ao provedor Web que hospeda a aplicação, processadas neste servidor e retornadas à máquina cliente.

A aplicação foi desenvolvida na ferramenta NetBeans IDE v.6.7.1, utilizando-se a linguagem Java, SDK v.6.

Para o ambiente Web, foram usadas as tecnologias JSP (*Java Server Pages*) na construção das páginas, e Servlets armazenados no servidor para processar todas as requisições feitas ao LMS. Utilizou-se o servidor Web TomCat v.6.0, e o servidor de Banco de Dados MySQL v.5.0.

**OA: Requisitos Não-Funcionais** [Voltar para OAs](#)

Definição	Exemplos	Sinônimos
Definição:  <b>São restrições sobre os serviços ou as funções oferecidas pelo sistema. Eles incluem restrições de timing, restrições sobre o processo de desenvolvimento e padrões. Aplicam-se, frequentemente, ao sistema como um todo. Em geral, não se aplicam às características ou serviços individuais de sistema.</b>		
<b>Os termos abaixo não foram cadastrados. Clique em cada um para cadastrar ou ignorar:</b>		
declarações	<input type="button" value="Cadastrar"/>	<input type="button" value="Ignorar"/>
serviços	<input type="button" value="Cadastrar"/>	<input type="button" value="Ignorar"/>
fornecer	<input type="button" value="Cadastrar"/>	<input type="button" value="Ignorar"/>
reagir	<input type="button" value="Cadastrar"/>	<input type="button" value="Ignorar"/>
entradas	<input type="button" value="Cadastrar"/>	<input type="button" value="Ignorar"/>

Figura 6: Tela da Interface do Professor

Na Interface Professor, o mesmo deve cadastrar sua(s) disciplina(s) trabalhada(s). Uma vez criada(s) a(s) disciplina(s), inicia-se o processo de criação dos OAs, com-

postos de conceitos que por sua vez têm uma definição, exemplo(s) e sinônimo(s). Ao entrar com a definição, o sistema identifica outros termos presentes dentro do texto digitado que podem se tornar OAs, e cabe ao professor decidir se cadastra os novos OAs ou não. O professor também cadastra as perguntas, e posteriormente, acessa os resultados obtidos pelos alunos ao responderem às perguntas.

A Figura 6 apresenta a tela da interface *Professor*, onde é apresentada a definição do OA “Requisitos Não-Funcionais”, e logo abaixo, os termos que podem ser ignorados, ou cadastrados, de acordo com o contexto utilizado pelo professor.

O protótipo é configurado de forma que à medida que o professor segue ignorando os termos usualmente irrelevantes ao domínio trabalhado, tais termos não são mais apresentados ao professor ao cadastrar novos OAs. Ou seja, o sistema automaticamente retira os termos ignorados anteriormente, facilitando com isso a criação de novos OAs pelo professor.

Na Interface Aluno, o mesmo deve efetuar o cadastro e logar-se no sistema. Após, deve escolher uma disciplina dentre as inseridas na plataforma, o que fará com que o sistema apresente as perguntas previamente cadastradas no ambiente pelo professor. Cabe ao aluno responder às perguntas, em linguagem natural, na forma de um exercício navegável. A figura 7 apresenta esta interface com o aluno.

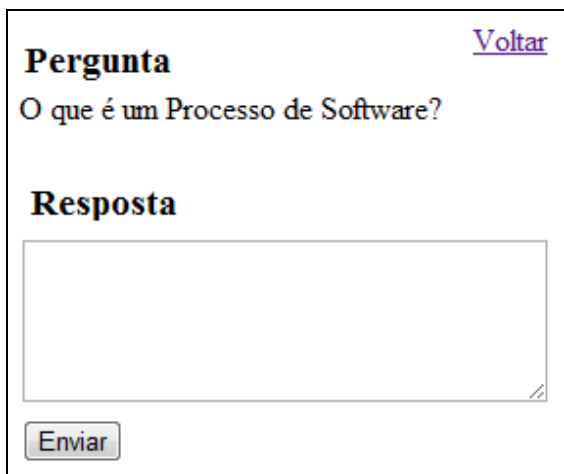


Figura 7: Interface para responder às perguntas do OA

À medida que as questões são respondidas, as respostas são transferidas para o servidor e devidamente armazenadas. O módulo de Inteligência pode ser acionado tanto por professores quanto por alunos, que obterão os resultados da comparação entre o conteúdo cadastrado

pelo professor e o texto entrado pelo aluno. Neste ponto, são utilizadas as técnicas de Mineração de Texto já citadas, que formarão a massa de dados a ser comparada pela métrica de abrangência utilizada no trabalho.

O protótipo utiliza dois algoritmos de *stemming*: O algoritmo de Porter [2] adaptado à língua portuguesa e o algoritmo de Orengo [14], que busca reduzir o *overstemming* do processo [15].

As respostas às perguntas, e também a sugestão de nota em cada questão respondida pelo aluno podem então ser consultadas pelo mesmo, e também pelo professor, através da tela apresentada na Figura 8.

## 5 Estudo de Caso

Foram realizados dois estudos de caso com o protótipo, um no domínio da Computação, com a disciplina Engenharia de Software, e outro no domínio da Química, com a disciplina Operações Unitárias. Os professores encarregaram-se de preparar a base de OAs, e elaborar as perguntas relativas aos OAs armazenados. A ferramenta foi disponibilizada aos alunos, que puderam experimentar o protótipo, na forma de exercícios avaliativos.

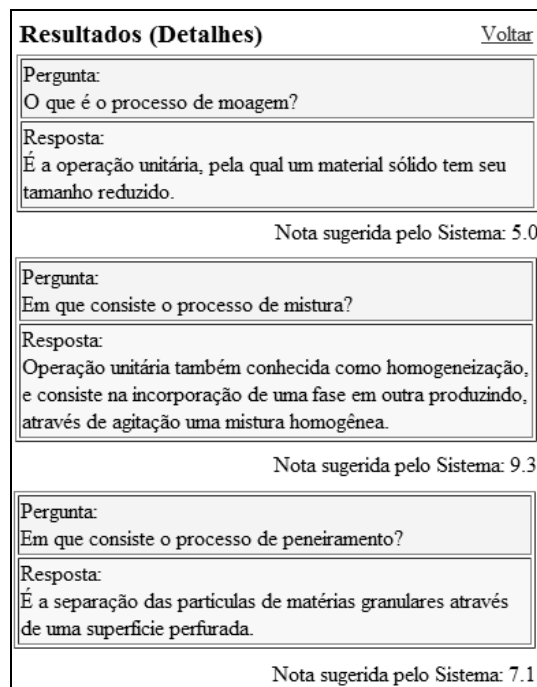


Figura 8: Exemplo de saída com os resultados de um aluno

Após responderem ao questionário, o protótipo identificou qual ou quais OAs corresponde(m) à pergunta, através de uma busca na base de OAs construída pelo

professor. O Módulo de Inteligência então transformou o conceito do OA encontrado em *tokens* em um vetor, e em seguida aplicou ao vetor os algoritmos de Mineração de Texto. Foi realizado o mesmo processo com o texto entrado pelo aluno, ou seja, também foi gerado um vetor e este submetido ao processo de Mineração de Texto.

Os professores prepararam os OAs, contendo cada um sua definição, exemplos e sinônimos relacionados, e também as perguntas referentes ao domínio trabalhado. Os questionários foram então submetidos às turmas, cada uma com aproximadamente 20 alunos. Os mesmos acessaram a plataforma, cadastraram-se e inscreveram-se na devida disciplina. Estes foram submetidos às perguntas, no formato de uma avaliação teórica.

Após a geração dos vetores, foi feita a comparação: verificou-se quais *stems* presentes no vetor do aluno estavam presentes no vetor do professor. Em seguida foi aplicada a métrica de abrangência, a qual dividiu-se o número de *stems* corretos presentes no vetor do Aluno pelo número de *stems* presentes no vetor do Professor. O resultado, sendo um número entre 0 e 1, foi multiplicado por 100 a fim de obter-se a nota sugerida ao aluno.

## 6 Resultados Obtidos

A tabela 1 apresenta os resultados obtidos por cinco dos alunos submetidos ao questionário. Nela pode-se observar a aplicação da Métrica de Abrangência no *stem* gerado para o aluno a fim de obter a sugestão de nota.

A nota sugerida ao final do processo é mostrada ao professor, de forma que ele possa verificar o desempenho da ferramenta. A mesma saída do processamento é também visualizada pelo aluno, com o intuito de fornecer um *feedback* sobre seu grau de domínio no assunto estudado.

Aluno	Stems corretos no vetor aluno	Equação de Abrangência	Nota sugerida pelo sistema
Aluno 1	12	12/14	8,6
Aluno 2	8	8/14	5,7
Aluno 3	10	10/14	7,1
Aluno 4	11	11/14	7,9
Aluno 5	9	9/14	6,4

Tabela 1: Notas sugeridas pelo protótipo

Torna-se importante observar que o fato de a nota de um aluno em uma determinada questão ser baixa não é necessariamente proporcional ao número de palavras

escritas para a resposta. Um dos fatores que influenciam na nota é a completude de informações do OA informada pelo professor no protótipo. Da mesma forma como uma apostila que não refletisse todo o conteúdo importante de um livro incidiria em um aluno que não teria toda a bagagem necessária para uma avaliação.

## 7 Conclusões

Este artigo descreveu uma proposta de arquitetura a ser utilizada em um ambiente de *E-learning*, que permite estruturar o conhecimento do professor em OAs e aferir os conhecimentos do aluno em um determinado domínio, de forma automática.

A utilização de OAs, baseados na metodologia CISCO, contribuiu no planejamento da arquitetura construída, além de tornar a implementação das classes mais robusta, segundo o paradigma orientado a objetos. Mas o principal benefício foi a implementação do reuso dos fragmentos de informação ao montar a estrutura de um curso, criando-se uma rede de conhecimentos navegável e dinâmica.

Os recursos de Mineração de Texto utilizados demonstraram que a área desenvolveu-se muito nos últimos anos, de forma que é possível aplicar seus recursos com resultados satisfatórios. O recurso de *Stemming*, especialmente o algoritmo de Porter para a língua portuguesa, mostrou-se bastante eficiente.

Alguns trabalhos futuros podem ser considerados, como apresentar ao aluno outros tipos de retorno, sejam na forma de conceitos não abordados em sua resposta, mas que constam na base de conhecimentos do professor, sejam na forma de sugestões de leitura baseadas na relação entre os conceitos cadastrados pelo professor no ambiente.

Também pode-se acrescentar ao protótipo um módulo que utilize recursos de aprendizagem de máquina, de modo que a ferramenta aprenda com seu uso ao longo do tempo. À medida que o professor pontue o desempenho da ferramenta após o processamento dos algoritmos de Mineração de Texto, a ferramenta deve levar em consideração este *feedback* do professor e acrescentá-lo como peso à métrica de Abrangência utilizada, tornando a ferramenta, à medida que vai sendo utilizada, cada vez mais eficiente em relação ao critério adotado pelo professor.

## Referências

- [1] Guest, S.; Juday, J. Guia para definição de um projeto de e-learning. E-learning Brasil News,

- ano 1, número 11, janeiro/2002. Disponível em [www.elearningbrasil.com.br/news/news11/artigo\\_1.asp](http://www.elearningbrasil.com.br/news/news11/artigo_1.asp) Acesso em novembro de 2008.
- [2] Porter, M. F. An Algorithm for suffix stripping, *Program*, 14, nº3, 130-137, 1980. Disponível em <http://www.tartarus.org/~martin/PorterStemmer/def.txt>. Acesso em Março de 2009.
- [3] Gomes, G. R. Integração de Repositórios de Sistemas de Bibliotecas Digitais e de Sistemas de Aprendizagem. Tese (Doutorado em Informática), Pontifícia Universidade Católica, Rio de Janeiro, 2006.
- [4] Huang, C. J.; Chen, H. X.; Chen, C. H. Developing argumentation processing agents for computer-supported collaborative learning. *Expert Systems with Applications* 36, Institute of Learning Technology, Taiwan, 2009.
- [5] Lau, R. Y. K. et al. Toward a Fuzzy Domain Ontology Extraction Method for Adaptive e-Learning. *IEEE Transactions on Knowledge and Data Engineering*, vol 21, no 6, 2009.
- [6] Moura, E. E. Uma Ferramenta de Mineração de Textos para Assistir as Dúvidas dos Alunos no Processo de Ensino em Ambientes Virtuais de Aprendizagem, Campos dos Goytacazes, Rio de Janeiro, 2006.
- [7] Tarouco, L. M. R. et al. Projeto CESTA – Coleção de Entidades de Suporte ao uso de Tecnologia na Aprendizagem. S.ed.: Porto Alegre, 2003. Disponível em: <http://www.cinted.ufrgs.br/CESTA>. Acesso em março de 2009.
- [8] Lucena, B. Novos Fragmentos no E-Learning: Desafios e Oportunidades para o Design. *Revista Brasileira de Aprendizaje Abierto y a Distancia*, 2003.
- [9] CISCO Systems Reusable Information Object Strategy. *Designing Information and Learning Objects Through Concept, Fact, Procedure, Process, and Principle Templates*, version 4.0, 2001.
- [10] Barritt, C., Lewis, D. *Reusable learning objects strategy: definition, creation process, and guidelines for building*. Version 4.1, november 2001. Cisco Systems, Inc.; Disponível em: [http://www.reusablelearning.org/docs/Cisco\\_rlo\\_roi\\_v3-1.pdf](http://www.reusablelearning.org/docs/Cisco_rlo_roi_v3-1.pdf) Acesso em agosto de 2005
- [11] Foly, L. S. Uma Arquitetura para Aferir o Conhecimento do Aluno em um Ambiente de Aprendizagem, Universidade Cândido Mendes, Campos dos Goytacazes, Rio de Janeiro, 2010.
- [12] Dias, M. A. L. Extração Automática de palavras-chave na língua portuguesa aplicada a dissertações e teses da área de engenharias. UNICAMP – Universidade Estadual de Campinas – Campinas, São Paulo, 2004.
- [13] De Lucca, J. L.; Nunes, M. G. V. Lematização versus Stemming, USP, UFSCar, UNESP, São Carlos, São Paulo, 2002.
- [14] Orengo, V. M; Huych, C. A Stemming Algorithm for the Portuguese Language, School of Computing Science, Middlesex University, London, England, 2001.
- [15] Chaves, M. S. Um Estudo e Apreciação sobre Algoritmos de Stemming para a Língua Portuguesa, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, Rio Grande do Sul, 2007.