

**UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS DA
COMPUTAÇÃO**

VANESSA MARTINS CALDAS

**UMA FERRAMENTA DE AVALIAÇÃO AUTOMÁTICA DE MAPAS
CONCEITUAIS COMO AUXÍLIO AO ENSINO EM AMBIENTES DE
EDUCAÇÃO A DISTÂNCIA**

Belém
2010

**UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO**

VANESSA MARTINS CALDAS

**UMA FERRAMENTA DE AVALIAÇÃO AUTOMÁTICA PARA
MAPAS CONCEITUAIS COMO AUXÍLIO AO ENSINO EM
AMBIENTES DE EDUCAÇÃO A DISTÂNCIA**

Belém
2010

**UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO**

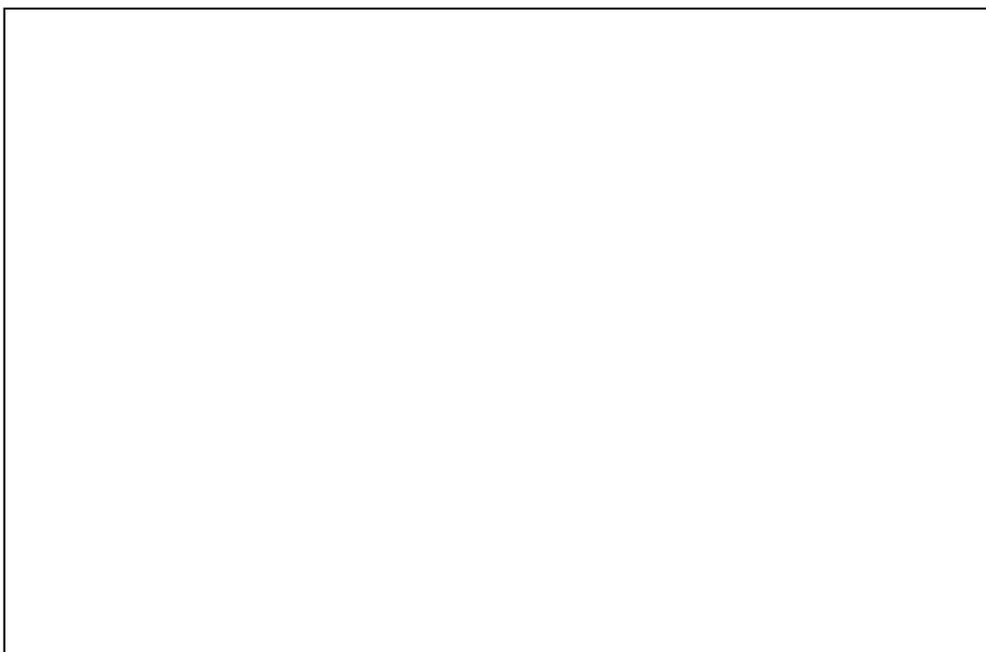
VANESSA MARTINS CALDAS

**UMA FERRAMENTA DE AVALIAÇÃO AUTOMÁTICA PARA
MAPAS CONCEITUAIS COMO AUXÍLIO AO ENSINO EM
AMBIENTES DE EDUCAÇÃO A DISTÂNCIA**

Dissertação submetida à Banca Examinadora do Programa de Pós-Graduação em Ciência da Computação da UFPA para a obtenção do Grau de Mestre em Ciência da Computação.

Belém
2010

(ficha catalográfica) – fornecida pela bibliotecária do ppgcc
“Verso da folha anterior”



Caldas, Vanessa Martins

Uma Ferramenta de Avaliação Automática para Mapas Conceituais como Auxílio ao Ensino em Ambientes de Educação a Distância/ Vanessa Martins Caldas; orientador, Eloi Luiz Favero - 2011.

Dissertação (Mestrado) - Universidade Federal do Pará, Instituto de Ciências Exatas e Naturais, Programa de Pós-Graduação em Ciência da Computação, Belém, 2011.

1. Educação na Computação. 2 Avaliação Automática. I. Título.
-

**UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO**

**UMA FERRAMENTA DE AVALIAÇÃO AUTOMÁTICA PARA
MAPAS CONCEITUAIS COMO AUXÍLIO AO ENSINO EM
AMBIENTES DE EDUCAÇÃO A DISTÂNCIA**

AUTORA: VANESA MARTINS CALDAS

DISSERTAÇÃO DE MESTRADO SUBMETIDA À AVALIAÇÃO DA
BANCA EXAMINADORA A SER AVALIADA PELO COLEGIADO DO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
DA UNIVERSIDADE FEDERAL DO PARÁ.

BANCA EXAMINADORA:

Prof. Dr. Eloi Luiz Favero
(ORIENTADOR – UFPA – Instituto de Ciências Exatas e Naturais)

Prof. Dr. Bianchi Seirique Meiguins
(UFPA – Instituto de Ciências Exatas e Naturais)

Prof. Dr. Manoel Ribeiro Filho
(UFPA – Membro PPGEE)

VISTO:

Prof. Dr. Sandro Ronaldo Bezerra Oliveira
(COORDENADOR DO PPGCC/ICEN/UFPA)

UFPA / ICEN / PPGCC

Dedico este trabalho a todos os professores e pessoas que se dedicam a ensinar.

AGRADECIMENTOS

Agradeço primeiramente a Deus por tudo o que ele me dá e por sempre colocar as coisas no meu caminho no momento certo. Agradeço por Ele ter me dado a oportunidade de progredir mais ou pouco em minha carreira e por me dar forças para concluir mais este passo na minha vida.

Agradeço ao Prof. Dr. Eloi Luiz Favero por ter me aceitado e acreditado em mim e no meu trabalho e pela atenção e paciência com que me orientou.

Agradeço à minha mãe Regina, que sempre ora por mim e sempre acredita em mim, por sempre me dar forças e ao meu pai Pedro por ser sempre sereno quando se trata da minha vida profissional, pelos sábios conselhos e por sempre me ajudar com o que ele pode. Agradeço a eles principalmente pelo fato de sempre fazerem de tudo, mesmo quando a luta é desigual, para me ver lá em cima.

Agradeço à minha irmã, Débora, por sempre me tirar do sério e fazer meus dias serem mais engraçados quando as coisas ficam tensas demais, seja na minha vida acadêmica, profissional ou pessoal.

Agradeço ao meu marido Nelson Magalhães por ter sido, acima de tudo, paciente ao longo dessa jornada, pelo incentivo a todo o momento e por me ajudar e apoiar sempre que pôde.

Um agradecimento especial à minha amiga Mireille, companheira nessa jornada, por toda a ajuda, por todas as conversas e trocas de idéias.

Agradeço a todos os colegas e pessoas que de alguma forma me apoiaram, me ouviram quando precisei ser ouvida ou simplesmente disseram o que eu precisava ouvir, mesmo sem saberem o que eu precisava ouvir.

SUMÁRIO

RESUMO	16
1. Capítulo Introdução	18
2. Capítulo Mapas Conceituais	23
2.1. A Teoria da Aprendizagem Significativa	24
2.2. Diferenciação Progressiva e Reconciliação Integrativa	25
2.3. Mapas Conceituais	26
2.4. Mapeamento de conceitos	27
2.5. Estrutura de MCs	28
2.6. Mapas Conceituais na Avaliação do Conhecimento	28
2.6.1. Avaliação Automática de Mapas Conceituais	29
3. Capítulo Trabalhos Relacionados	32
3.1. Análise dos trabalhos relacionados	33
3.2. Ferramenta de Hipermídia para Avaliação de Questões Interdisciplinares	33
3.3. COMPASS: An Adaptive Web-Based Concept Map Assessment Tool (Ferramenta de Avaliação Adaptativa de Mapas Conceituais em Ambientes Web)	36
3.4. Avaliação da Aprendizagem: Uma Abordagem Qualitativa Baseada em Mapas Conceituais, Ontologias e Algoritmos Genéticos	44
3.5. Concept Map Based Knowledge Assessment System	47
3.6. Considerações Finais	51
4. Capítulo Ferramenta de Avaliação de Mapas Conceituais para Ambientes de EAD 53	
4.1. Introdução à Ferramenta de Avaliação de Mapas Conceituais	54
4.2. Processo de Extração de Informação dos Mapas Conceituais	54
4.2.1. Normalização das informações	55
4.3. Processo de Avaliação Quantitativa	56
4.4. Processo de Avaliação Qualitativa	59
4.5. Etapas de Avaliação	62
4.5.1. Etapas da avaliação qualitativa	64
4.5.2. Etapa de avaliação quantitativa	64
4.6. Arquitetura do sistema	64
4.7. Considerações	65
5. Implementação	67
5.1. Introdução	68
5.2. Módulo de Extração de Informação	68
5.3. Módulo N-Gramas	71
5.4. Módulo KNN	73
5.5. Módulo de Entrada/Saída	78
5.6. Considerações	79
6. Experimentos Realizados e Resultados Obtidos	80
6.1. Introdução	81
6.2. Experimentos Realizados	81
6.3. Experimentos da Avaliação Quantitativa	81
7. Considerações Finais	86
7.1. Conclusões	87
7.2. Resultados Alcançados	88
7.3. Trabalhos Futuros	88
7.4. Publicações Relacionadas com a Dissertação	89
8. Referências	90

Anexo A – Modelagem da Extração de Informação	94
Anexo B – Modelagem N-Gramas	95
Anexo C – Modelagem do KNN	96
Anexo D – Modelagem do Sistema	97
Anexo E – Fluxo KNN	98
Anexo F – Métodos compararLigacoesString()	99
Anexo G – Métodos compararLigacoesArray()	100
Anexo H – Métodos listarConceitos() e de listagem de ligacoes	101
Anexo I – Método compararLigacoesRelatorio().....	102
Anexo J – Classe KnnExecucao	103
Anexo K – Classe KnnExecucaoCross.....	104
Anexo L – Método gerarModelo()	106
Anexo M – Métodos recuperarModelo() e calcularMedia()	107

LISTA DE FIGURAS

Figura 2.5-1: Mapa Conceitual com a definição do conhecimento a respeito da Aprendizagem Significativa (Adaptação de Novak, 1998).....	28
Figura 3.2-1. Medidas para avaliar a estrutura dos Mapas Conceituais (Schaal, 2008). 36	36
Figura 3.3-1. Modelo de MC desenvolvido por um professor na ferramenta COMPASS (Anohina, Grundspekis, 2006).....	37
Figura 3.3-2. Tela exibida para o estudante, apresentando a tarefa de completar o MC com os conceitos existentes na primeira etapa de aprendizagem (Anohina, Grundspekis, 2006).....	38
Figura 3.3-3. Tela com o MC desenvolvido pelo aluno para a primeira etapa de aprendizagem (Anohina, Grundspekis, 2006).	38
Figura 3.3-4. Relatório de diagnóstico exibido após a finalização da tarefa (Anohina, Grundspekis, 2006).....	40
Figura 3.3-5. Construção de um MC complementar ao anterior como segunda etapa da tarefa.	40
Figura 3.3-6. Arquitetura do COMPASS (Gouli et.al., 2004).....	42
Figura 3.4-1. Arquitetura do CMTTool's.	45
Figura 3.5-1. Arquitetura baseada em agentes do sistema de avaliação do conhecimento baseado em MCs.....	48
Figura 4.2.1-1. Exemplo de normalização realizado durante processamento de extração de informações.....	56
Figura 4.3-1 Fluxo do processo de avaliação quantitativa (modificar os MCs para cima e a nota do MC para baixo)	57
Figura 4.3-2. MC Modelo de resposta (à esquerda) e exemplo de MC de aluno (à direita).....	58
Figura 4.4-1. Fluxograma do processo de avaliação qualitativa.	60
Figura 4.4-2. MC de aluno a ser avaliado.	61
Figura 4.4-3. Exemplos de resultados do processamento do MC do aluno e dos modelos de referência para o sistema.....	61
Figura 4.4-4. Relatório-guia processado ao final da avaliação qualitativa, onde as proposições com “*” são as proposições consideradas corretas incluídas no relatório..	62
Figura 4.5-1. Fluxograma do processo de avaliação	63
Figura 4.6-1. Arquitetura da ferramenta de avaliação automática.	65
Figura 5.2-1. Métodos obterFraseMCPorString() e obterFraseMCPorArray() da classe Mapeamento.	70
Figura 5.2-2. Saída do processamento de extração de informação para um MC de aluno e MC modelo.	71
Figura 5.3-1. Interface base de implementação da análise de N-Gramas.....	72
Figura 5.4-1. Classe MapaConceitual e seus atributos componentes.....	74
Figura 5.4-2. Classe MapaConceitualCross, filha da classe MapaConceitual.	75
Figura 5.4-3. Interface IKnnExecucao.	75
Figura 5.4-4. Método calcularKnn() da classe KnnCalculo.	76
Figura 5.4-5. Método de cálculo da distância euclidiana para unigrama, bigrama e trigrama.....	77
Figura 6.3-1. Média de erros para as questões 64 e 115, variando-se o número de MC (1 até 4) para compor a reposta modelo e o valor do K (1 até 4) para KNN: menor erro 1,08 com 2 MC e K=3.	83
Figura 6.3-2. Notas do especialista <i>versus</i> notas preditas pelo sistema para a questão 64, utilizando 2MC modelo e classificador KNN com K=3.	84

Figura 6.3-3. Notas do especialista comparadas com as notas preditas pelo sistema para a questão 115, utilizando 2 MC modelo e classificador KNN com $K=3$ 84

LISTA DE TABELAS

Tabela 3.2-1: Matriz inter-relacional de um Mapa Conceitual	34
Tabela 3.4-1 Símbolos utilizados no CMTool's.....	45
Tabela 3.6-1. Tabela comparativa entre sistemas de avaliação automática de MCs.....	52
Tabela 5.3-1. Amostra do relatório de saída do processamento de N-gramas sobre os MCs dos alunos.	58
Tabela 5.4-1. Amostra do relatório de saída com as notas preditas para cada MC.....	78
Tabela 6.4-1. Média dos erros da comparação nota predita versus nota do avaliador para K=1 e K=2.	82
Tabela 6.4-2. Média de erros da comparação nota predita versus nota do avaliador para K=3 e K=4.	82

LISTA DE SIGLAS

AG	<i>Algoritmo Genético</i>
AVA	<i>Ambientes Virtuais de Aprendizagem</i>
EAD	<i>Ensino a Distancia</i>
KNN	<i>K Nearest Neighbor</i>
MC	<i>Mapa Conceitual</i>

RESUMO

A avaliação automática de questões abertas é uma característica almejada em ambientes de Ensino a Distância, pois permite ao professor avaliar de forma mais adequada o desempenho do estudante. Hoje, a avaliação é utilizada apenas questões objetivas (questões fechadas). Pesquisas vêm sendo feitas com o intuito de criar ferramentas de avaliação automática para questões abertas. Neste contexto, dentre as questões abertas, a abordagem de Mapas Conceituais caracteriza-se como uma metodologia voltada para a avaliação que permite uma visão mais detalhada do aprendizado do aluno. Este trabalho apresenta uma proposta de avaliação automática quantitativa e qualitativa de Mapas Conceituais baseada em técnicas de Inteligência Artificial (n-gramas e KNN). Os resultados desta proposta atingem uma acurácia de aproximadamente 90% de acertos em relação a avaliadores humanos para a avaliação quantitativa.

PALAVRAS-CHAVE: Avaliação automática, Mapas Conceituais, n-gramas, KNN.

ABSTRACT

Virtual Learning Environment requires automated assessment for opened questions once it allows a better way to evaluate students' performance. Therefore, this evaluation is usually use on objective questions (closed questions). Researches have been made in order to create automated evaluation for opened questions, but the main focus of these tools lays in short essays. Through opened questions, Conceptual Maps has a role as an assessment methodology that allows a detailed vision about students' learning, where proposals to do automated assessment have been researched. This paper describes a proposal that uses Artificial Intelligence Techniques (n-grams and Knn) to assess students' Conceptual Maps in a quantitative way and qualitative way. The results achieved an average of 90% rights related to humans experts for quantitative assessment, while considering qualitative assessment it's created guide reports to students in order to help them with the Conceptual Map construction.

KEYWORDS: Automatic evaluation, Concept Maps, n-grams, Knn.

1. Capítulo

Introdução

Este capítulo apresenta a introdução do trabalho, descrevendo o objetivo, a proposta e a estrutura da dissertação.

1.1 Introdução

A avaliação discente no contexto educacional é vista como um processo que tem como finalidade verificar o aprendizado de um aluno. O processo de avaliação geralmente é realizado através da aplicação de testes, exercícios e trabalhos com o objetivo de quantificar o desempenho do aluno. Entretanto, devido à sobrecarga que esta tarefa impõe, torna-se difícil para o docente acompanhar de maneira efetiva e gradual o processo de aprendizado do estudante. Essa preocupação é maior principalmente no que concerne à educação a distância (EaD) em larga escala. EaD é um paradigma de educação na qual o ensino é mediado através de meios de comunicação e tecnologias com a finalidade de auxiliar no processo de ensino-aprendizagem, uma vez que docente e discente encontram-se distantes seja em dimensões espaciais, temporais ou ambas (Brasil/Mec, 2000). Nesse sentido, a complexidade no processo de avaliação e o esforço necessário para realizá-la aumentam.

O ensino a distância é um paradigma em expansão dentro da educação, uma vez que o corpo discente que se encontra ambientado no ensino à distância tem mostrado uma taxa de crescimento (cerca de 20% ao ano), bem superior à modalidade presencial (Moodle 2008), reforçando a necessidade de pesquisar metodologias de avaliação automática para ambientes de EaD. Ambientes Virtuais de Aprendizagem (AVA), que são softwares que se utilizam da comunicação através da internet para auxiliar no processo de EaD, tais como Teleduc (Teleduc 2008) e Moodle (Moodle 2008), têm apoiado o ensino em EaD. Assim, com a finalidade de auxiliar no ensino AVAs oferecem serviços como correção automática de questões fechadas com o intuito de amenizar estes problemas de avaliação.

Todavia, a avaliação automática de questões fechadas não fornece uma análise efetiva sobre o processo de aprendizado do aluno, pois demonstra apenas em parte o que o estudante entendeu a respeito de um assunto e como construiu este conhecimento. As questões abertas demonstram de maneira mais clara o raciocínio do aluno (Novak 1984).

Dentre as questões abertas encontram-se os Mapas Conceituais (MCs), cuja principal característica reside na habilidade de expressar o conhecimento individual de cada pessoa acerca de um assunto (Cañas 1999). Esta característica torna os MCs uma

ferramenta importante em AVAs, pois possibilita o acompanhamento gradual da construção do conhecimento, mostrando ao professor o desenvolvimento e dificuldades do aluno acerca do tópico ministrado.

O desafio encontra-se no desenvolvimento de ferramentas que automatizem o processo de avaliação para questões abertas, sejam elas pequenas questões discursivas, ensaios ou MCs. Estudos neste sentido ocorrem desde meados da década de 60 quando o sistema pioneiro PEG (*Project Essay Grader*) foi desenvolvido para avaliar pequenas questões discursivas (Page 1967). Entretanto, com o surgimento de novas técnicas como PLN (Processamento de Linguagem Natural) e EI (Extração da Informação) em meados da década de 90, retomaram-se as pesquisas e novas ferramentas foram desenvolvidas, a exemplos dos sistemas IEA e E-Rater, (Hearst 2000), que atingem uma acurácia de acerto em torno de 0.83 em relação a avaliadores humanos em experimentos controlados. A avaliação automática de questões fornece algumas vantagens em relação à avaliação manual, tais como:

- Apresenta menor sobrecarga no que concerne à avaliação para o docente, onde este poderá dedicar-se a uma avaliação mais qualitativa e pontual de acordo com a necessidade de cada aluno;
- Oferece maior visibilidade ao docente do desempenho dos alunos através de relatórios avaliativos, permitindo ao professor identificar problemas, como a deficiência de um aluno em relação a um tópico ministrado de forma mais rápida e tomar decisões para solucioná-las de forma mais eficiente;
- Possui retorno imediato dos resultados para os estudantes, característica importante principalmente em ambientes EAD onde o professor não está sempre disponível ao aluno.

1.2 Motivação

A avaliação, de maneira geral, é uma tarefa dispendiosa que custa tempo e dedicação por parte do docente. As questões abertas são as que exigem maior tempo e dedicação do docente, pois requerem uma análise detalhada a respeito do conhecimento do aluno. Dentre as questões abertas, os MCs são uma vertente cuja estrutura evidencia tanto os aspectos apreendidos quanto as dificuldades do aluno.

Desta maneira, a avaliação de MCs é bastante efetiva no acompanhamento do aprendizado do aluno. Entretanto, permanece o esforço realizado pelo docente que não consegue dar a atenção adequada a todos os aspectos do conhecimento que não foram bem adquiridos pelo aluno. Conseqüentemente o docente tem dificuldade de acompanhar de maneira efetiva o processo de aprendizado do aluno. A solução para este problema é a automatização do processo de avaliação. A avaliação automática auxilia o docente na tarefa de acompanhamento de aprendizado dos estudantes, além de liberá-lo para a realização de outras tarefas docentes, visando à melhoria do ensino.

1.3 Objetivo Geral

Desenvolver o projeto de um modelo que automatize o processo de avaliação de MCs, podendo ser integrado a um ambiente Web de ensino a distancia (EAD). Esta proposta é centrada em técnicas de inteligência artificial, utilizando o algoritmo de classificação KNN combinando com a técnica de similaridade via N-gramas para realizar a avaliação. A acurácia alcançada pelo modelo deve encontrar-se em torno de 80% a 90% em média de acertos.

1.4 Objetivos específicos

Estudar o modelo pedagógico de avaliação através de MCs (Cañas, 1999), (Rocha, 2004), (Novak, 2002), (Ausubel, 1980), (Moreira e Buchweitz, 1993). Estudar os temas de avaliação automática de questões abertas para formar uma base para proposição de um modelo para avaliação automática de MCs. Focar os aspectos qualitativos (conceitos e listagem de proposições) e também quantitativos, através do retorno de um conceito numérico, cuja escala varia de 0 a 10. Implementar o modelo de avaliação. Tornar a proposta acessível para ambientes de EAD. Buscar uma acurácia em torno de 80% a 90% se comparada com especialistas humanos, como encontrado na literatura especificada.

1.5 Estrutura do trabalho

Este trabalho propõe uma ferramenta que realiza uma avaliação automática de MCs no contexto de uma plataforma de ensino a distância. O foco da avaliação encontra-se na atribuição de escores às atividades desenvolvidas pelos alunos. Além da atribuição de escore, a ferramenta também tem a finalidade de auxiliar ao aluno na

obtenção do conhecimento e conseqüentemente no desenvolvimento das tarefas através de relatórios-guia gerados pela ferramenta.

Além desta introdução, esta dissertação encontra-se dividida em mais seis capítulos:

- O capítulo 2 explana sobre os conceitos acerca dos Mapas Conceituais;
- O capítulo 3 mostra trabalhos relacionados à pesquisa de avaliação automática de MCs;
- O capítulo 4 aborda a ferramenta proposta neste trabalho, descrevendo seu processo de avaliação de MCs;
- O capítulo 5 descreve a implementação da ferramenta de acordo com suas funcionalidades;
- O capítulo 6 descreve os experimentos realizados e os resultados provenientes destes experimentos de acordo com as avaliações aplicadas (quantitativa e qualitativa);
- O capítulo 7 discorre sobre as conclusões obtidas no decorrer dos experimentos realizados sobre a ferramenta.

2. Capítulo

Mapas Conceituais

Este capítulo explana sobre Mapas Conceituais como ferramenta que auxilia o processo de aprendizagem. Neste capítulo são abordados os principais conceitos relacionados aos Mapas Conceituais, bem como as variadas formas de realizar uma avaliação com eles. Além disso, aborda-se também a relação da utilização de Mapas Conceituais com a capacidade de aprendizagem do indivíduo ao fazer associações e sua relação com a Teoria da Aprendizagem Significativa.

2.1. A Teoria da Aprendizagem Significativa

A Teoria da Aprendizagem Significativa foi proposta por David Ausubel (2000) em meados da década de 60. Esta teoria era uma proposta alternativa ao paradigma educacional dominante da época denominado aprendizagem habitual, que ditava haver apenas uma resposta correta para uma questão. Em contrapartida, a teoria de Ausubel (1980) determinava que o processo de aprendizagem dependia tanto do conhecimento armazenado na mente de um indivíduo como da sua capacidade de processamento de informação.

Além disso, a aprendizagem habitual visa à aquisição de novos conhecimentos através da memorização, onde novos conceitos são armazenados arbitrariamente sem serem relacionados a nenhum conhecimento pré-existente do indivíduo. Por outro lado, o princípio da teoria de Ausubel reside na idéia que o indivíduo deve escolher relacionar novos conhecimentos apreendidos a conceitos relevantes presentes em sua estrutura cognitiva (Novak, 1984).

A Teoria da Aprendizagem Significativa determina que a aprendizagem de um novo conhecimento deve estar relacionada a um conhecimento relevante e prévio do indivíduo. O relacionamento entre conceitos novos e antigos é realizado através de uma ancoragem, sendo esta ligação o núcleo principal da teoria de Ausubel. A ancoragem ocorre quando um conceito mais específico e menos inclusivo é ligado a um conceito já existente e mais geral chamado subsunçor. A ancoragem de conceitos causa uma pequena mudança tanto no significado do conceito subsunçor como na informação já armazenada na estrutura cognitiva do indivíduo (Novak, 1998).

Assim, a aprendizagem significativa caracteriza-se através da ligação entre conceitos novos e conceitos pertencentes a conhecimentos pré-existentes na estrutura cognitiva de um indivíduo. Esta forma de aprendizagem oferece grandes vantagens, dentre as quais podemos citar (Novak, 1984):

- O conhecimento adquirido é retido por muito mais tempo, ou seja, o estudante não esquece o conhecimento apreendido dentro de semanas;
- A informação apreendida de forma significativa pode ser aplicada em uma grande variedade de novos problemas ou contextos; a transferência de

conhecimento é alta, sendo esta transferência importante para o pensamento criativo.

Por outro lado, a qualidade da aprendizagem é relativa à riqueza conceitual existente no material a ser estudado. Deste modo, a aprendizagem significativa necessita de três requisitos básicos (Novak, 1984):

- Conhecimento prévio relevante: O aprendiz deve saber alguma informação prévia relativa à nova informação a ser apreendida;
- Material rico em significados: O conhecimento a ser apreendido deve ser relevante a outro conhecimento e deve possuir conceitos significativos e proposições;
- Preferência pela aprendizagem significativa: O aluno deve escolher a aprendizagem significativa, ou seja, deve-se escolher consciente e deliberadamente relacionar conhecimentos novos aos antigos.

2.2. Diferenciação Progressiva e Reconciliação Integrativa

O refinamento da estrutura cognitiva do indivíduo causada pela aquisição de novos conceitos, bem como as conseqüentes mudanças nos significados dos conceitos dá maior precisão e especificidade aos conceitos (Novak 1998). Este fenômeno denomina-se *Diferenciação Progressiva* da estrutura cognitiva, sendo esta uma das principais características resultantes da teoria da aprendizagem significativa.

A diferenciação progressiva leva não somente a uma adição quantitativa de conceitos a estrutura conceitual, mas também resulta na mudança qualitativa em que o sentido de cada conceito existente na estrutura cognitiva é modificado de alguma forma, seja direta ou indiretamente. Além da diferenciação progressiva, outra característica da Aprendizagem Significativa é a *Reconciliação Integrativa* que tem por finalidade a detecção de novas inter-relações entre conceitos existentes na estrutura cognitiva (Novak 1998). A reconciliação integrativa tem como base a distinção de novas relações que não se encontravam listadas anteriormente nos MCs, havendo dois modelos de reconciliação integrativa: *Aprendizagem combinatorial* e *reconciliação integrativa superordinate*.

A aprendizagem combinatorial é caracterizada quando o aprendiz percebe a necessidade de uma relação entre conceitos que se encontram em diferentes

ramificações do MC. A reconciliação integrativa superordinate é caracterizada através da identificação de um conceito mais inclusivo, não identificado previamente no MC, e que deve se relacionar com conceitos já existentes no MC.

A ancoragem do conhecimento através de Reconciliação Integrativa e Diferenciação Progressiva é uma forma de avaliação sobre MCs, uma vez que contempla a análise da estrutura do MC desenvolvido por um indivíduo. Esta avaliação ocorre de acordo com a estrutura desenvolvida do MC, onde se verifica a forma como o conhecimento foi adquirido e quais os pontos falhos neste conhecimento.

2.3. Mapas Conceituais

Os Mapas Conceituais (MCs) foram desenvolvidos por Joseph Novak (1998) com o objetivo de colocar em prática a Teoria da Aprendizagem Significativa. Os MCs visam externalizar o conhecimento contido na estrutura cognitiva de um indivíduo, refletindo seu nível de conhecimento a respeito de determinado assunto (Ruiz-Primo et.al. 2001). A externalização do conhecimento é visível através da representação de relações significativas entre conceitos na forma de proposições. Proposições são unidades semânticas constituídas de conceitos interligados através de links (palavras de ligação), que representam a natureza da relação entre estes conceitos.

A externalização do conhecimento é individual e é uma das principais qualidades dos MCs, pois evidencia o conhecimento do indivíduo a respeito de determinado assunto. Assim, a estrutura de um MC sobre um mesmo contexto é necessariamente diferente para cada indivíduo. Desta maneira é inviável referenciar um único MC como solução a um problema, uma vez que há diferentes representações para o mesmo assunto abordado (Cañas, 2004).

Além disso, a externalização do conhecimento possibilita uma melhor visualização no que concerne às dificuldades de um indivíduo, bem como seus conhecimentos superficiais de acordo com o contexto. Esta característica é importante principalmente quando MCs são utilizados na educação para avaliar a aprendizagem, onde os professores podem verificar as dificuldades dos alunos e tentar supri-las.

O desenvolvimento de um MC considera as principais características e princípios da teoria de Ausubel, bem como sua idéia central, que é a idéia de ancoragem de conhecimentos. A ancoragem possibilita refletir o caráter evolutivo do conhecimento

de acordo com os conceitos que são apreendidos pelo indivíduo. Assim, o MC assume um aspecto hierárquico, onde os conceitos mais gerais são os primeiros a ser apreendidos e, portanto, assumem as posições mais altas na descrição da estrutura do MC. Os conceitos mais específicos são ligados posteriormente aos conceitos mais gerais e localizam-se na base da estrutura do MC.

2.4. Mapeamento de conceitos

O mapeamento de conceitos é uma tarefa flexível, sendo utilizada para diversos fins, tais como: metodologia de avaliação, instrumento de análise de currículo, entre outros (Moreira e Buchweitz, 1993). Além disso, a estrutura de um MC pode abranger tanto um conhecimento de contexto macro como de um contexto detalhado. Essa magnitude de visualização acerca da estrutura do MC depende exclusivamente do conhecimento que se deseja transpor através do MC. A diferença de magnitude encontra-se no grau de generalidade e inclusividade dos conceitos existentes no MC.

Assim como a reconciliação integrativa e a diferenciação progressiva, a generalidade e a inclusividade são consideradas formas de avaliação de MC, pois realizam uma análise através da dimensão vertical de um MC, apresentando a idéia de hierarquia dentro do MC. Na generalidade é avaliada a abrangência de um conceito, onde os conceitos mais abrangentes encontram-se no topo do MC, enquanto conceitos mais específicos encontram-se na base do MC. A idéia de inclusividade é proveniente do grau em que um conceito engloba outros conceitos. Os conceitos mais gerais são os mais inclusivos, situados no topo do MC, enquanto que os conceitos mais específicos podem ou não englobar conceitos, situando-se na base do MC (Buchweitz, 1984).

Os conceitos de generalidade e inclusividade são diretamente proporcionais, uma vez que quanto mais abrangente é um conceito maior sua inclusividade. Deste modo, conceitos com aproximadamente o mesmo nível de generalidade e inclusividade aparecem na mesma posição vertical de um MC.

Um MC pode expressar uma visão geral a respeito de um determinado assunto, utilizando apenas conceitos gerais e inclusivos para expressar, por exemplo, o planejamento de um curso. Um MC que possui visão mais detalhada utiliza conceitos mais específicos e pouco inclusivos para representar, por exemplo, a seleção de materiais instrucionais para utilizar durante o curso.

2.5. Estrutura de MCs

A estrutura de um MC é formada por nós e arcos de um grafo. Nós são conceitos definidos por Novak (1998) como uma regularidade percebida em eventos ou objetos, ou registros de eventos ou objetos, assinalados por um rótulo. Enquanto os arcos são unidades que realizam a conexão entre os conceitos.

A representação gráfica de uma estrutura de MC é feita através de um grafo, onde os nós são representados através de retângulos ou elipses, enquanto os arcos são representados através de linhas rotuladas com a natureza da ligação entre os conceitos como apresentado na Figura 2.5-1.

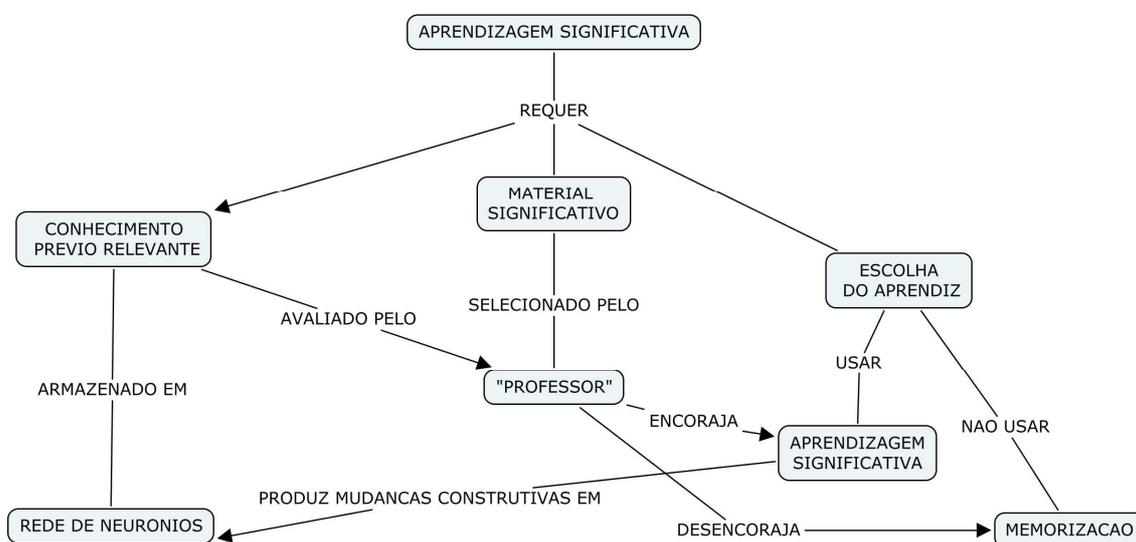


Figura 2.5-1: Mapa Conceitual com a definição do conhecimento a respeito da Aprendizagem Significativa (Adaptação de Novak, 1998).

A estrutura de um MC tende a ilustrar o conhecimento através de caminhos que assumem significados como listas de proposições, onde cada proposição corresponde a uma tripla conceito-link-conceito. Na finalização da tarefa, o MC provê um sumário esquemático sobre o que foi aprendido.

2.6. Mapas Conceituais na Avaliação do Conhecimento

A utilização de MCs como ferramenta de avaliação pode ser vista como um procedimento para medir a estrutura do conhecimento declarativo do estudante (Ruiz-Primo, Shavelson, 1996).

Além disso, a variabilidade de tarefas de MCs permite lidar mais facilmente com a quantidade de informações que será provida ao aluno durante a realização de uma avaliação. Ruiz-Primo (2004) define dois tipos de questões envolvendo MCs, onde estas questões são classificadas de acordo com a quantidade de informações fornecidas ao aluno. Questões que provêm informações durante a realização da tarefa, como o complemento de um MC semi-pronto com proposições, são definidas como altamente dirigidas (*high directness*). Por outro lado, questões que visam verificar o conhecimento adquirido pelo aluno, como a construção completa de um MC, são denominadas fracamente dirigidas (*low directness*).

2.6.1. Avaliação Automática de Mapas Conceituais

Os MCs têm como principal característica a habilidade de expressar o conhecimento individual de cada pessoa acerca de um assunto (Cañas, 1999). Esta característica torna os MCs uma ferramenta importante, principalmente em Ambientes Virtuais de Aprendizagem (AVAs), pois possibilita o acompanhamento gradual da construção do conhecimento, mostrando ao professor o desenvolvimento e as dificuldades do aluno acerca do tópico ministrado.

Entretanto, a tarefa de avaliação é desgastante à medida que aumenta o número de alunos. Este cenário torna-se mais complexo quando a avaliação é realizada em um AVA, uma vez que este geralmente encontra-se atrelado a ambientes EAD, onde aluno e professor encontram-se distantes espacial e temporalmente. Uma solução de auxílio no desafio de realizar uma avaliação em AVA é a automatização do processo de avaliação de MCs.

A importância de automatizar a avaliação de MCs reside em criar a oportunidade ao docente de acompanhar de modo mais efetivo o desenvolvimento do conhecimento dos estudantes. Inúmeras propostas de avaliação automática para MCs foram idealizadas, utilizando diferentes métricas para realizar avaliações de acordo com seus objetivos.

Geralmente, a avaliação automática tem como metodologias dois tipos de avaliação: a qualitativa e a quantitativa. A avaliação qualitativa visa a acurácia de cada proposição presente no MC do aluno. A avaliação quantitativa visa avaliar os

componentes estruturais válidos de acordo com um MC de referência desenvolvido por um especialista (Gouli et al., 2005).

Dentre os trabalhos de avaliação automática, o método de avaliação de Novak e Gowin (1984) é frequentemente utilizado como base para calcular o escore de um MC. Este método tem como base pontuar os MCs através das seguintes premissas:

- Proposições: Frases de ligações utilizadas para ligar conceitos, consideradas válidas e significativas;
- Hierarquia: Grau de subordinação de um conceito em relação a outro no MC;
- Ligações Cruzadas: Ligações significativas e válidas entre um segmento da hierarquia conceitual e outro segmento;
- Exemplos: Acontecimentos e objetos que sejam exemplos válidos do que significam os termos conceituais.

Em 1999, McClure et al. comparou métodos holísticos, relacionais e estruturais sem e com o uso de MCs de referência, alcançando maior nível de confiabilidade na avaliação que considera um MC de referência, atingindo uma acurácia de 76%.

Ruiz-Primo (2004) criou uma abordagem que reflete o conhecimento de um estudante sobre determinado tópico comparando seu MC a estrutura de um MC de referência desenvolvido por um especialista. Além disso, Ruiz-Primo distinguiu as tarefas de MCs em duas categorias (altamente dirigida e fracamente dirigida) e ponderou três metodologias de cálculo de escore: (i) acurácia de proposição; (ii) convergência com um MC de referência e (iii) saliência, que é a proporção de proposições válidas fora do escopo de proposições no MC do estudante (Ruiz-Primo, 2000).

O ponto em comum destas pesquisas reside no fato de que em todas elas é indicado o uso de MC de referência na avaliação, havendo diversas características que podem ser avaliadas no processo de criação de MCs:

- Considerando o processo como um todo, pode-se coletar a seqüência de operações de ancoragem (diferenciação progressiva e reconciliação integrativa), realizando uma avaliação quantitativa;

- Considerando a generalidade e a inclusividade, pode-se obter uma avaliação qualitativa ao verificar se um conceito foi adequadamente ancorado no MC.
- Considerando a representação de proposições é possível verificar quanto o estudante representa a forma correta/incorreta no MC: (i) através de uma avaliação qualitativa (representada por listagem de proposições e conceitos); (ii) através de uma avaliação quantitativa segundo um valor atribuído às proposições de acordo com seu grau de pertinência em relação ao assunto abordado.

Fundamentando-se nestas idéias, o capítulo 3 apresenta algumas propostas de avaliação automática de MCs, mostrando as técnicas e características utilizadas na avaliação, bem como cenário especificado para cada ambiente.

3. Capítulo

Trabalhos Relacionados

Este capítulo apresenta alguns trabalhos de avaliação automática de MCs existentes, descrevendo a técnica utilizada, a arquitetura do avaliador e os resultados dos experimentos realizados na validação das propostas.

3.1. Análise dos trabalhos relacionados

Os trabalhos relacionados têm como objetivo comum realizar uma avaliação automática de MCs, focando na utilização de agentes como auxiliares no processo de desenvolvimento de MCs e às vezes relacionando o resultado final a um escore. Estes escores são resultados de métricas utilizadas no processo de avaliação. Entretanto, as abordagens de geração de escores destas propostas não produzem escalas cujo fim seja a própria nota do aluno, por exemplo, uma escala de 0 a 10 ou uma escala classificatória como, por exemplo, Regular, Bom e Excelente.

As metodologias de avaliação aplicadas variam de acordo com a proposta, sendo possível classificar as tarefas de Mapas Conceituais em dois tipos (Ruiz Primo, 2004):

- Altamente dirigidas (*High directness*): envolvem tarefas como preenchimento de conceitos dentro de um dado MC ou o complemento de um MCs semi-pronto com proposições;
- Fracamente dirigidas (*Low directness*): envolve tarefas de construção de MCs baseado apenas no conhecimento do indivíduo a respeito do assunto.

3.2. Ferramenta de Hipermissão para Avaliação de Questões Interdisciplinares

Histórico & objetivo. O desenvolvimento do ambiente de aprendizagem em hipermissão foi criado por Schaal (2008) para avaliar quantitativamente MCs desenvolvidos por estudantes em um contexto colaborativo. Este ambiente tem como objetivo a aprendizagem interdisciplinar, onde o aluno deve ter como conhecimento prévio conceitos das disciplinas em questão, elaborando um contexto mais complexo de modo a explicar o novo conhecimento. A aquisição de um novo conhecimento geralmente requer a interconexão de conceitos científicos básicos interdisciplinares para um entendimento adequado.

O conhecimento interdisciplinar pode ser visto como a construção de uma estrutura de conhecimento bem fundamentada aplicando metodologia e linguagem para mais de uma disciplina a fim de examinar o tópico central. Esta estrutura de conhecimento consiste de conceitos interconectados e provê a base para o entendimento conceitual.

O ambiente de hipermídia é o mais adequado para este tipo de questão, pois auxilia na construção de conhecimento dentro de domínios mais complexos. Entretanto, o material hipermídia deve estar de acordo com o nível de conhecimento do estudante. O resultado da avaliação é um escore que é utilizado para medir o ganho de aprendizado interdisciplinar após a utilização do ambiente.

Técnica. Antes e após o processo de instrução os alunos são submetidos a testes, que consistem em desenvolver um MC de acordo com o assunto abordado. Os MCs são desenvolvidos utilizando o software MaNet® 1.6, que permite a construção MCs com ou sem dicas de conceitos ou relações, além de aplicar diferentes técnicas de atribuição de escore sobre as proposições desenvolvidas nos MCs. Neste experimento, os MCs dos alunos foram avaliados através de atributos estruturais que consideram:

- O “volume” de um Mapa Conceitual, indicado pelo numero total de relações utilizadas no MC do estudante;
- A “granularidade” que consiste da divisão dos MCs em sub-Mapas;
- A quantidade de proposições corretas em relação ao volume.

Além dos atributos estruturais os mapas também são comparados a um MC de referência como outra forma de avaliação. Nessa avaliação valor de similaridade é calculado baseado em um coeficiente de correspondência, que é obtido de acordo com a matriz inter-relacional listada na Tabela 3.2-1:

Tabela 3.2-1: Matriz inter-relacional de um Mapa Conceitual

Mapa do estudante	-	Mapa referência	Label
Não conectado	e também	Não conectado	cnc (correto e não conectado)
Não conectado	porém	Conectado	mc (falta de conexão)
Conectado	porém	Não conectado	wc (conexão errônea)
Conectado	e também	Conectado	cc (conexão correta)

O coeficiente de correspondência é calculado pela seguinte fórmula:

$$\text{Correspondência} = \frac{\sum \text{cnc} + \sum \text{cc} - (\sum \text{mc} + \sum \text{wc})}{\text{Máxima quantidade de interconexões possíveis}}$$

Fórmula 3.2-1. Fórmula para cálculo do coeficiente de correspondência

Ao coeficiente de correspondência pode-se ainda acrescentar pesos de modo a aumentar o rigor durante a avaliação dos MCs, modificando a fórmula original apresentada na Figura 3.2-1, inserindo os pesos. Os critérios considerados para aumento do rigor são:

- Todo link considerado correto, para o assunto abordado, entre dois ou mais conceitos é considerado como válido;
- Links são contabilizados como corretos desde que seu rótulo esteja correto;
- Links são contabilizados como corretos se a posição em que se encontram e seu rótulo estão corretos.

Assim, os pesos são calculados da seguinte forma:

$$\text{Peso 1} = \frac{\sum \text{corretamente conectado no mapa}}{\sum \text{incorretamente conectados no mapa}} \quad \text{Peso 2} = \frac{1}{\text{Peso 1}}$$

Fórmula 3.2-2. Cálculo dos pesos.

Resultados. O trabalho de Schaal provê uma avaliação qualitativa de MCs em um contexto interdisciplinar e colaborativo. Esta avaliação é realizada de acordo com as proposições existentes nos MCs segundo o "volume" de um MC, que é caracterizado por sua quantidade de relações, "granularidade", que é a divisão de MC em sub-mapas não conectados e quantidade de proposições válidas em relação ao volume. Assim, um MC é considerado bem estruturado quando há uma alta inter-relação entre conceitos distintos (volume) e baixo grau de granularidade.

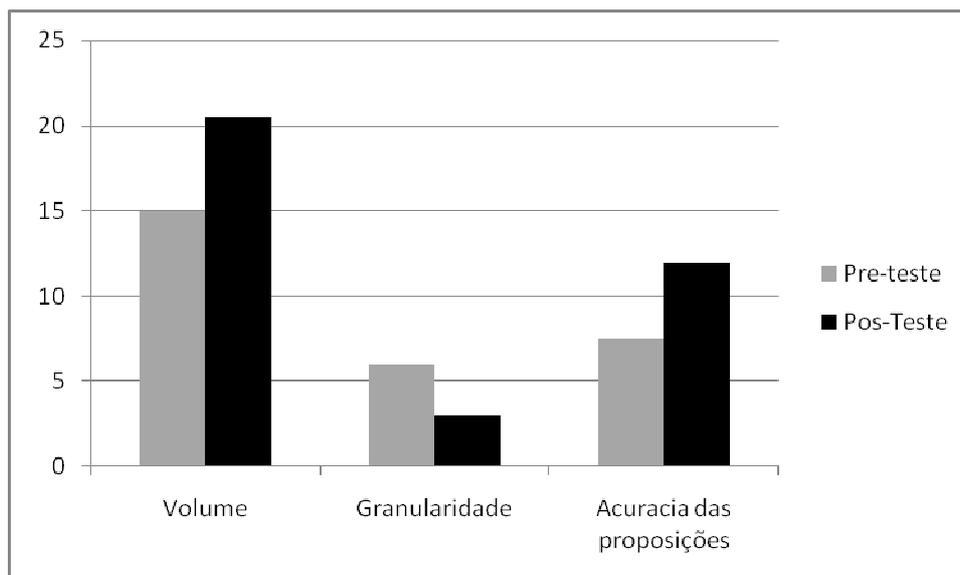


Figura 3.2-1. Medidas para avaliar a estrutura dos Mapas Conceituais (Schaal, 2008).

Os experimentos foram realizados sobre 106 alunos com idade média de 15 anos que trabalharam em grupos, cuja formação era de livre escolha, totalizando 53 grupos. Os resultados da avaliação qualitativa indicaram que os escores dos atributos estruturais de um MC de aluno são maiores ao serem trabalhados colaborativamente no ambiente hipermídia de ensino e após a devida instrução, como pode ser observado na Figura 3.2-1/ Figura 2.5-1. Além disso, foi observado que, após a instrução, o conhecimento do aluno se ajusta consideravelmente a estrutura de domínio do MC de referência.

3.3. COMPASS: An Adaptive Web-Based Concept Map Assessment Tool (Ferramenta de Avaliação Adaptativa de Mapas Conceituais em Ambientes Web)

Histórico & objetivo. COMPASS (COnccept MaP ASSEssment tool) é uma ferramenta de avaliação de MCs que foi desenvolvida por Gouli et. al. (2004). Esta ferramenta pertence à arquitetura do framework AssessToLearn (Gouli et. al, 2003) que utiliza MCs para várias funções de avaliação, e tem como objetivo avaliar o entendimento dos estudantes e realizar o suporte ao processo de aprendizagem, da seguinte maneira:

- Avaliação do entendimento: é realizada através da utilização de atividades de avaliação (através de tarefas de Mapas Conceituais específicas) e aplicando um esquema para estimativas quantitativa e qualitativa do conhecimento do estudante;

- Suporte ao processo de aprendizagem: É realizada através do provimento de diferentes componentes de *feedback* ajustados de acordo com as necessidades do estudante.

Técnica. COMPASS possui um esquema de avaliação embutido que considera tanto os conceitos presentes quanto os conceitos que faltam. Essa visualização de conceitos presentes e ausentes é proveniente da comparação entre o MC desenvolvido pelo aluno e o MC de referência, criado por um professor. A avaliação qualitativa é baseada na análise qualitativa de erros e tem como objetivo contribuir no diagnóstico qualitativo do conhecimento do aluno, identificando entendimentos errôneos.

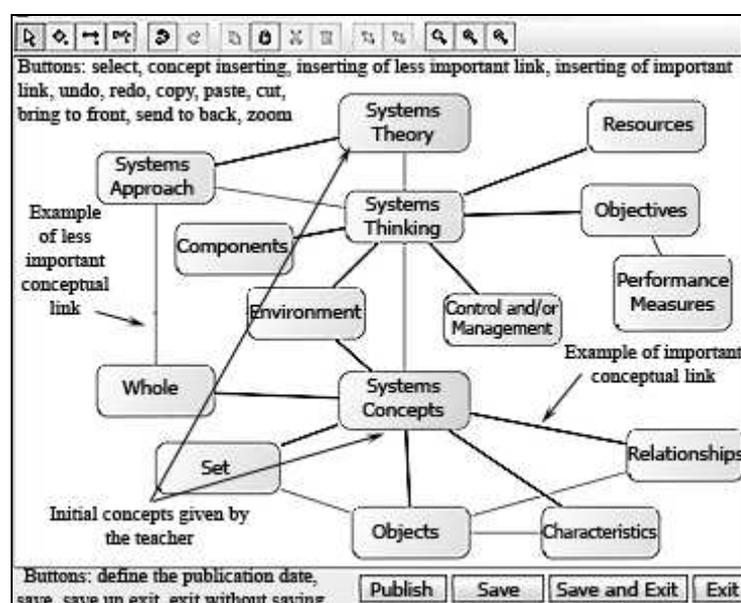


Figura 3.3-1. Modelo de MC desenvolvido por um professor na ferramenta COMPASS (Anohina, Grundspekis, 2006).

A Figura 3.3-1 apresenta a ferramenta COMPASS com um MC de referência, que será usado para avaliar os MCs desenvolvidos pelos estudantes em duas etapas de aprendizagem. Este MC tem 20 proposições dentre elas doze são consideradas importantes e oito são consideradas superficiais. O desenvolvimento do MC do aluno é realizado na tela apresentada na Figura 3.3-2.

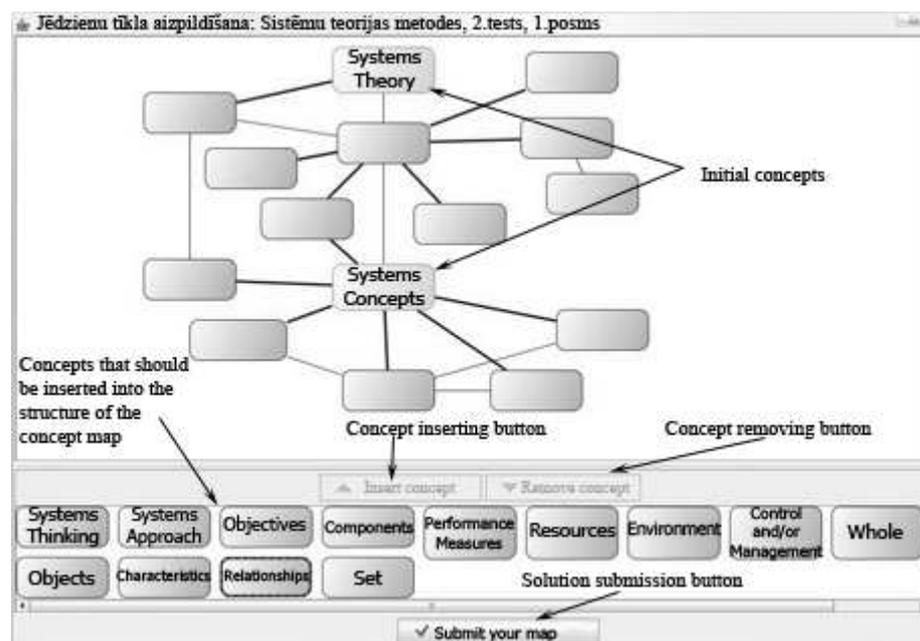


Figura 3.3-2. Tela exibida para o estudante, apresentando a tarefa de completar o MC com os conceitos existentes na primeira etapa de aprendizagem (Anohina, Grundspekis, 2006).

Na Figura 3.3-2 é apresentada a tela do estudante, onde ele desenvolve o raciocínio para o assunto em questão. A ferramenta baseia-se em questões *altamente dirigidas* para a construção dos MCs dos estudantes, uma vez que os conceitos pertinentes ao assunto são apresentados juntamente com outros que não são condizentes com o assunto, mas que estão participando como ruído. Assim, o estudante completa o MC apresentado a ele com os conceitos que lhe são oferecidos, cujo resultado é apresentado na Figura 3.3-3.

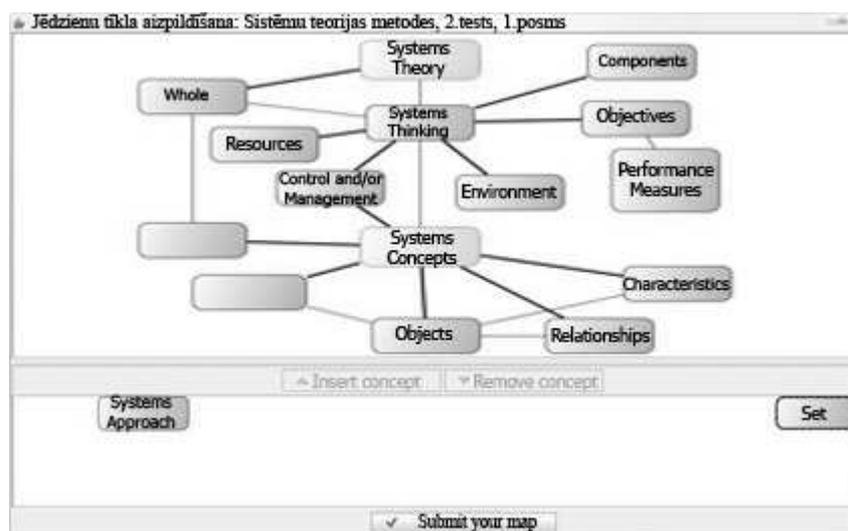


Figura 3.3-3. Tela com o MC desenvolvido pelo aluno para a primeira etapa de aprendizagem (Anohina, Grundspekis, 2006).

A análise quantitativa é baseada no peso atribuído a cada categoria de erro, assim como a cada conceito e proposição que aparece no mapa de referência. Os pesos são atribuídos pelo professor e refletem o grau de importância dos conceitos e proposições, bem como das categorias de erro. O processo de aprendizagem consiste dos seguintes passos:

- Os pesos dos conceitos, que existem tanto no MC de referência quanto no Mapa do aluno, e os pesos das proposições são adicionados ao escore total;
- As proposições/conceitos parcialmente corretos, onde seus pesos são ajustados de acordo com o peso da categoria de erro e adicionados ao escore total;
- O escore total do aluno, que é dividido pelo escore do Mapa de referência para produzir uma taxa como um índice de similaridade.

O resultado das avaliações qualitativa e quantitativa são utilizados para prover um *feedback* personalizado de acordo com os erros identificados, visando estimular o aluno a refletir a respeito de seu conhecimento sobre o assunto abordado (Gouli et.al., 2005). Entretanto, existem vários componentes de *feedback* que são disponibilizados para os estudantes, alguns são ofertados durante a execução de uma tarefa, tendo como exemplo as listas de conceitos (ver Figura 3.3-2), onde alguns dos conceitos são relevantes e outros agem como ruídos, pois são considerados errôneos ou supérfluos para a construção do MC.

Dentre estes componentes de *feedback* o mais comum é o relatório de diagnóstico apresentado na Figura 3.3-4, que é gerado ao final de cada tarefa realizada pelo aluno. Este relatório mostra ao estudante os conceitos que foram erroneamente relacionados, bem como uma lista dos conceitos que não foram contemplados no MC e que estavam disponíveis durante a tarefa. Além disso, o relatório também provê ao estudante o escore obtido para a tarefa desenvolvida e máximo ponto de escore que ele poderia ter obtido.

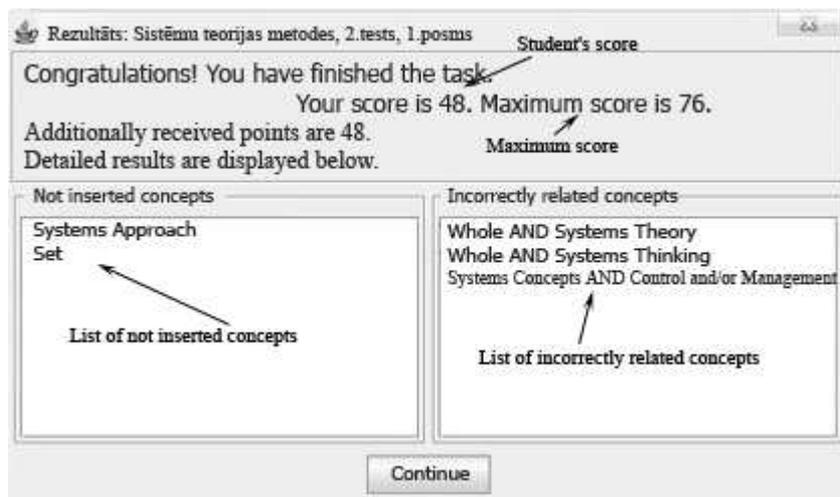


Figura 3.3-4. Relatório de diagnóstico exibido após a finalização da tarefa (Anohina, Grundspekis, 2006).

Terminada a tarefa da primeira etapa os resultados são salvos e a nova tarefa da segunda etapa é apresentada, sendo esta uma continuação da tarefa anterior. Retomando-se a Figura 3.3-1 que apresenta o MC desenvolvido pelo professor, percebe-se que apenas metade dos conceitos foi apresentada aos estudantes na primeira tarefa. Assim, a outra parte do MC contendo os conceitos restantes é apresentada em uma nova etapa como tarefa complementar, como pode ser observada na Figura 3.3-5.

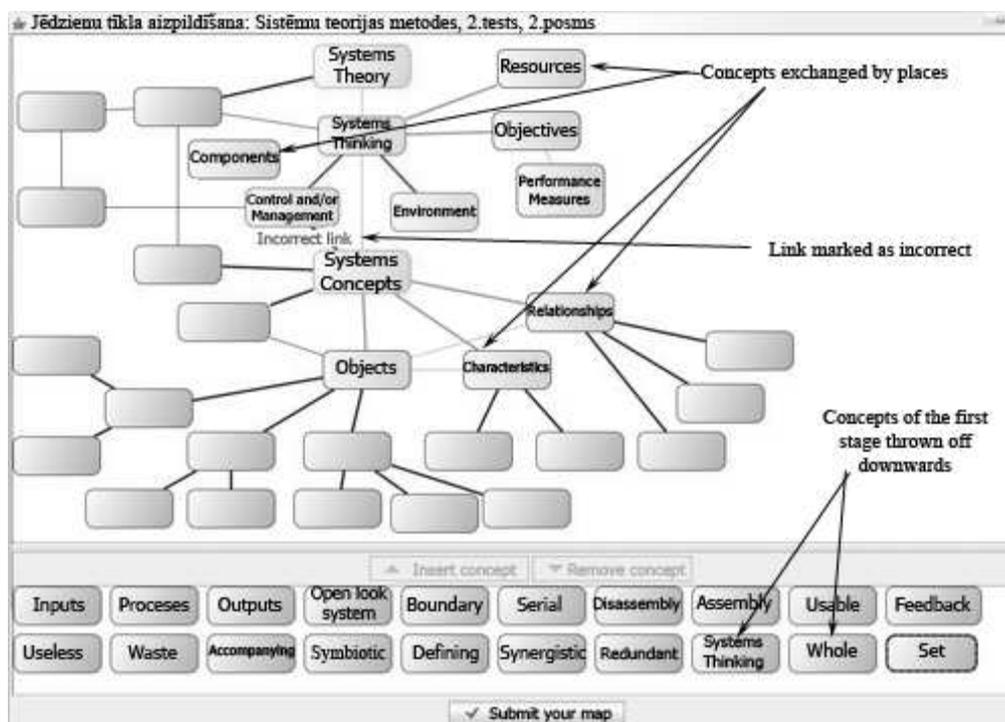


Figura 3.3-5. Construção de um MC complementar ao anterior como segunda etapa da tarefa.

A Figura 3.3-5 mostra a segunda tarefa, que consiste em utilizar o conhecimento construído anteriormente de modo a agregar novos conhecimentos. Os novos

conhecimentos serão avaliados da mesma forma relatada anteriormente e, assim, o estudante descreve seu domínio cognitivo de acordo com as tarefas que lhe são ofertadas.

Arquitetura. A arquitetura do COMPASS possui funcionalidades que abrangem desde interface de tarefas para os alunos até geração de componentes de *feedback* para auxiliar aos alunos no desenvolvimento das tarefas. Esta arquitetura encontra-se dividida em cinco módulos de acordo com as principais funcionalidades do sistema:

- Módulo de monitoramento e interação (IMM): é responsável pela coleta de dados relacionados ao comportamento do estudante. Além disso, é responsável pela ativação de outros módulos de acordo com as ações realizadas pelo estudante;
- Módulo de diagnóstico (DM): realiza a avaliação do MC do estudante baseado em um MC de referência;
- Módulo de geração de *feedback* adaptativo (AFGM): gera um feedback personalizado de acordo com o nível de conhecimento, preferências e comportamento do estudante;
- Módulo de apresentação (PM): apresenta as partes constituintes do MC que será disponibilizado para o aluno ao completar a tarefa.
- Módulo de armazenamento de dados: contém o Conhecimento de Domínio (Domain Knowledge - DK) e o Modelo do Estudante (Learner's Model - LM).

Os módulos da arquitetura do COMPASS podem ser visualizados na Figura 3.3-6.

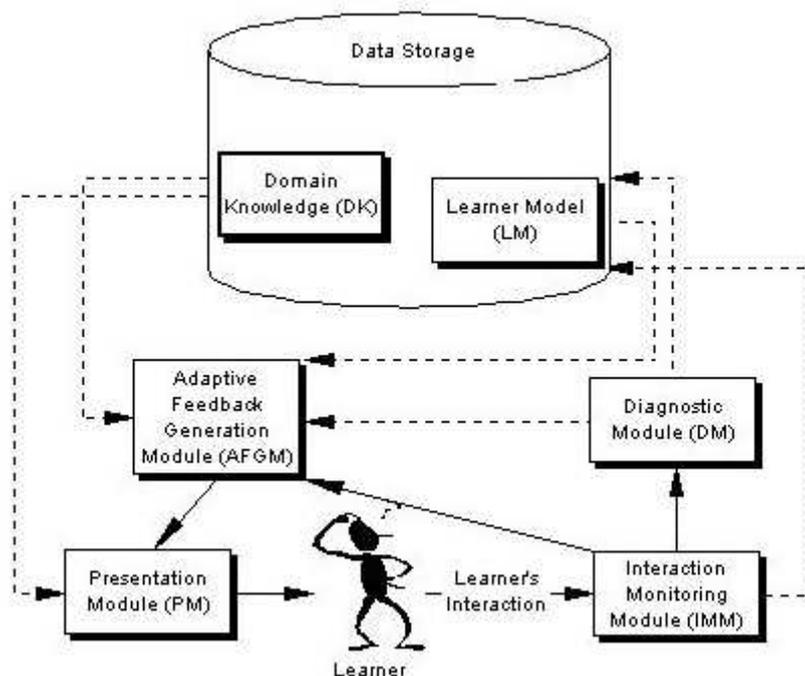


Figura 3.3-6. Arquitetura do COMPASS (Gouli et.al., 2004).

O conhecimento de domínio pertencente ao módulo de armazenamento de dados possui como função selecionar o tipo de avaliação a ser aplicada de acordo com o tipo de avaliação a ser selecionada pelo estudante. Cada tipo de avaliação é amplamente analisado para especificar resultados da avaliação através de várias atividades de avaliação, retornando um feedback ajustado de acordo com as necessidades do aluno. As atividades de avaliação seguem um processo de três passos, onde para cada passo existe uma atividade de avaliação que realiza funções específicas:

- Constatar o conhecimento prévio do estudante: a atividade de avaliação visa habilitar os alunos a ativar seus conhecimentos existentes, além de auxiliar os professores a: (i) Elencar o conhecimento prévio dos alunos; (ii) Identificar o nível do conhecimento inicial do aluno; (iii) Identificar conceitos desconhecidos e entendimentos incompletos ou errôneos do aluno;
- Promover a construção de conhecimento e identificar mudanças conceituais: a atividade de avaliação visa habilitar os professores a monitorar e avaliar as mudanças progressivas do estudante durante a instrução. Além disso, promove a construção do conhecimento e o desenvolvimento, aumentando os níveis de habilidades e levando o aluno a exercer pensamento crítico e reflexão, que o induz a avaliar seu próprio progresso;

- Avaliar a construção de conhecimento: a atividade de avaliação habilita o aluno a refinar o próprio conhecimento. No caso do professor, a avaliação habilita a capturar o crescimento do entendimento geral do aluno e a identificar como o conhecimento foi construído após a instrução.

Um *feedback* é gerado após a análise das atividades de avaliação. Este *feedback* considera características e necessidades individuais de cada aluno, sendo uma das fontes mais importantes para auxiliar os estudantes a reestruturar seus conhecimentos. Inicialmente, a tarefa do *feedback* é trabalhar a conhecimento do estudante a respeito dos conceitos relatados por ele durante a tarefa. Caso o aluno permaneça no erro, a ferramenta busca redirecionar a linha de raciocínio do aluno através de dicas para que o erro seja corrigido e a tarefa seja completada. Unidades de tutoria também são utilizadas como *feedback*, elas são relevantes aos conceitos e relações incluídos no mapa, permitindo ao estudante revisar o material educativo relevante aos atributos da resposta correta (Gouli et.al., 2004).

Resultados. A validação da ferramenta proposta foi feita através de experimentos realizados com 65 alunos selecionados aleatoriamente com idade média entre 13 e 14 anos. Estes alunos foram separados em dois grupos, onde um (com 32 alunos) faria uma avaliação tradicional, enquanto o outro (com 33 alunos) utilizaria o COMPASS como ferramenta de auxílio à aprendizagem.

Os experimentos tinham como objetivo avaliar o desempenho dos alunos ao trabalhar com a ferramenta COMPASS. Para alcançar este objetivo, o MC do aluno era comparado a dois MCs de referência desenvolvidos por especialistas. A comparação, realizada através do índice de similaridade de Goldshmidt (1991), revelou que a similaridade entre os MCs de referência e o MC do aluno alcançou uma média de 85%.

Além do índice de similaridade, o *feedback* oferecido para os alunos foi bem aceito, onde 94% dos estudantes o consideraram muito prático. A aceitação positiva do *feedback* reside em sua habilidade adaptativa de fornecer um retorno personalizado ao aluno através de seus variados tipos. Além disso, o *feedback* se encontra disponível a qualquer momento da execução da tarefa. A resposta do *feedback* em tempo de execução da tarefa para o aluno torna a aprendizagem mais simples, auxiliando o aluno no desenvolvimento cognitivo a respeito do assunto ministrado.

3.4. Avaliação da Aprendizagem: Uma Abordagem Qualitativa Baseada em Mapas Conceituais, Ontologias e Algoritmos Genéticos

Histórico e Objetivo. O CMTTool's é um sistema de avaliação de aprendizado desenvolvido por Rocha (2007), que considera a aprendizagem humana e sua avaliação como um problema de natureza adaptativa e evolucionária, utilizando MCs de modo a aplicar a teoria da aprendizagem significativa. O principal objetivo deste sistema é ajudar estudantes e professores a aplicar princípios construtivistas no decorrer do processo de aprendizagem escolar (Rocha et. al. 2004).

O CMTTool's proporciona uma avaliação qualitativa, comparando o MC do estudante com uma Ontologia criada a cada assunto abordado. O resultado é um relatório que evidencia se o conhecimento do estudante é compartilhado com a Ontologia, possibilitando a utilização de inferências do MC do estudante.

Técnica. A avaliação realizada no ambiente CMTTool depende de componentes fundamentais que trabalham interdependentes:

- Um domínio de ontologia que armazena conceitos, relações binárias, palavras de ligação, bem como a função de medir a distância semântica entre os conceitos, proposições e Mapas Conceituais;
- Um algoritmo genético (AG) que gera coleções de MCs (espaço de busca) de acordo com a Ontologia a fim de realizar a avaliação;
- Um avaliador que usa o espaço de busca gerado pelo AG e a Ontologia para detectar evidências do aprendizado no MC do aluno.

Ao receber um MC para analisar, o avaliador examina o repositório em busca de uma população que possa ser usada na análise atual. Se a população satisfaz a necessidade, ela é recuperada e usada, caso contrário o AG é ativado para ler uma ontologia a fim de gerar uma população de MCs. Esta ontologia também é lida pelo avaliador, que obtém o resultado das funções que calculam as distâncias semânticas. Após a realização da avaliação, a ontologia e a população de MCs são armazenadas em um repositório.

Arquitetura. O ambiente do CMTTool's envolve sete módulos interdependentes juntamente com um repositório. A Figura 3.4-1 mostra a arquitetura do sistema.

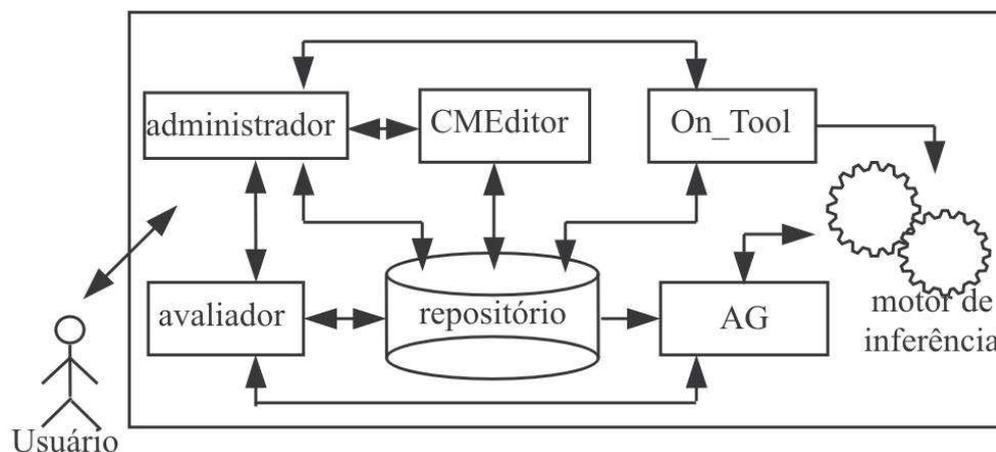


Figura 3.4-1. Arquitetura do CMTool's.

O módulo administrador é responsável pelo controle de acesso ao ambiente (registro e identificação de usuários). O sistema de registro do administrador atribui ao usuário o direito de acessar funcionalidades específicas do ambiente de acordo com seu perfil (perfil de professor, perfil de aluno).

O módulo CMEditor é utilizado principalmente pelos estudantes, pois implementa uma linguagem visual para construir MCs de acordo com a Teoria da Aprendizagem Significativa. A Tabela 3.4-1 mostra alguns dos símbolos que são utilizados na construção dos MCs no CMEditor.

Tabela 3.4-1 Símbolos utilizados no CMTool's

Símbolo	Semântica
	Usado para representar conceitos genéricos
	Usado para representar exemplos de conceitos
	Usado para representar hierarquias e diferenciação progressiva
	Usado para representar hierarquias e reconciliação integradora
	Usada para escrever proposições agrupadas (conjunção)

O módulo On_Tool realiza a construção e armazenamento no repositório da coleção da ontologia de domínio associado a tarefa de aprendizagem. As ontologias de domínio são construídas desenhando as entidades e atribuindo dimensões de relacionamento para as ligações entre elas. As dimensões de relacionamento são

selecionadas na taxonomia de frases de ligação e incorporadas na ontologia em construção.

O repositório contém informações importantes para o funcionamento do ambiente, dentre as quais se destaca a taxonomia de frases de ligação. Esta taxonomia implementa um modelo capaz de:

- Distinguir duas proposições diferentes, embora próximas;
- Atribuir valores de distância semântica proporcionais à diferença em significado entre duas proposições quaisquer que estejam sendo comparadas.

O módulo do motor de inferência tem como funcionalidade interpretar e combinar axiomas que representam as dimensões semânticas da taxonomia. O principal objetivo desta funcionalidade é validar proposições existentes nos MCs desenvolvidos de acordo com o conhecimento estabelecido na ontologia.

O módulo de AG possui um algoritmo genético centrado na ontologia denominado GAADT-CM. Este algoritmo possui axiomas que descrevem a formação de seus elementos básicos e operadores, utilizados para gerar populações de MC. As populações geradas são conjuntos de MCs semelhantes ao MC utilizado como entrada no algoritmo.

O módulo avaliador usa os resultados produzidos pelo AG e a ontologia para produzir uma avaliação completa do estudante ou do grupo de estudantes. Além disso, o avaliador também tem a função de gravar no repositório o resultado de cada avaliação realizada. Ao aplicar o avaliador ao MC as seguintes informações são mostradas:

- Proposições válidas;
- Hierarquias conceituais válidas;
- Exemplos;
- Relação de proximidade semântica entre o MC do aluno e os MCs da população gerados pelo AG.

Resultados. O CMTTool (Rocha, 2007) (Rocha, Favero, 2004) foi submetido a vários experimentos na Universidade Federal do Pará, o qual realizou com êxito as avaliações individuais sobre o entendimento dos alunos, de acordo com os princípios cognitivistas, em experimentos simulados. Até o presente momento o CMTTool foi

avaliado somente através de simulações que testaram a viabilidade do AG no processo de avaliação. Pesquisadores estão preparando uma versão operacional para trabalhar em ambientes de EAD.

3.5. Concept Map Based Knowledge Assessment System

Histórico e Objetivo. Esta proposta desenvolvida por Anohina (2007) tem como objetivo de emular a avaliação de um professor através de uma avaliação de conhecimento sistemática. Este sistema tem seu funcionamento baseado na interação entre professor e aluno e nos agentes que são responsáveis pela avaliação dos MCs desenvolvidos pelos estudantes. Ao final do processo de avaliação é retornando um *feedback* ao aluno, formulado de acordo com as necessidades do estudante, após verificados os erros existentes e os conceitos que faltam comparados ao MIC de modelo.

Técnica. O processo de avaliação do conhecimento funciona através da interação do sistema com seus usuários: docente e estudante. A idéia de avaliação consiste na divisão do curso de aprendizagem em vários níveis de aprendizagem, onde o MC é construído de acordo com cada nível, de tal forma que o MC do nível atual seja uma extensão do MC do nível anterior.

O professor cria MCs de referência para cada nível de aprendizagem do curso. O estudante desenvolve seu MC de acordo com o conhecimento exigido pelo nível em questão e ao final do desenvolvimento, recebe o modelo desenvolvido pelo professor como *feedback* imediato. Após a finalização do MC o aluno submete sua solução e o agente inteligente tenta adequar a solução a um dos cinco padrões de solução existentes em sua base de conhecimento. Em seguida à análise, é gerado um escore de comparação para o MC e este escore juntamente com o MC do aluno são salvos em uma base de conhecimento. Ao final o agente retorna um *feedback* ao aluno com as devidas correções do MC.

Arquitetura. O cenário de avaliação é possui como núcleo principal os agentes do sistema, que possuem tarefas específicas e interdependentes:

- Comunicação: é responsável por perceber as ações superficiais tomadas pelo usuário, bem como a visualização da estrutura do MC e seus elementos;

- Avaliação do conhecimento: realiza a comparação entre os mapas conceituais do aluno e do professor, calcula um escore para o estudante, coleta informações estatísticas e gera o *feedback* que é entregue ao agente de comunicação;
- Registro interativo: armazena em um banco de dados o MC do estudante enviado pelo agente de comunicação, bem como os resultados provenientes da comparação entre o MC do estudante e o MC de referência proveniente do agente de avaliação do conhecimento;
- Especialistas: formula uma tarefa de acordo com o estágio de conhecimento atual do aluno, baseando-se no mapa conceitual de referência e no MC desenvolvido pelo aluno no estágio anterior.

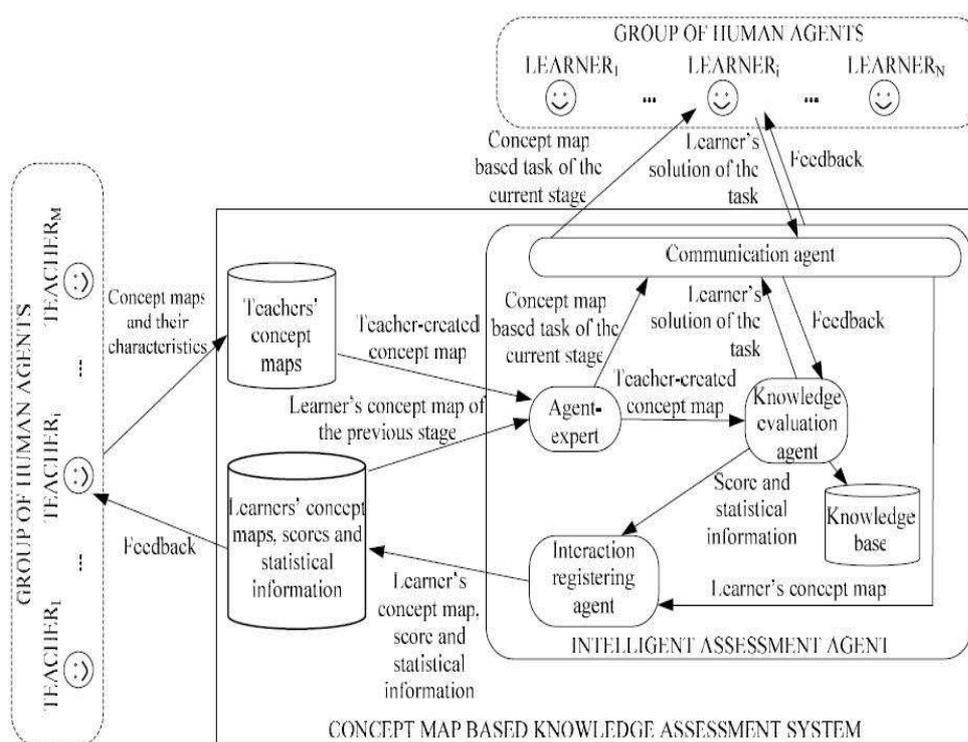


Figura 3.5-1. Arquitetura baseada em agentes do sistema de avaliação do conhecimento baseado em MCs.

O sistema de avaliação trabalha com aumento e diminuição do grau de dificuldade para as tarefas existentes. No aumento do grau de dificuldade não somente conceitos e links são considerados durante a avaliação, mas também se eles possuem relação com o assunto abordado e que posições assumem dentro do contexto do MC. O aumento do grau de dificuldade é avaliado através da inserção de conceitos adicionais

ou da oferta de outras questões ao estudante, onde o escore do estudante é calculado através da Equação 3.5-1.

$$P = \left(\sum_{i=1}^n p_i * c_i \right) * (1 - c_s * s - \sum_{i=1}^j (a + \Delta * (i - 1) / m) / m)$$

Equação 3.5-1. Equação de avaliação do Mapa Conceitual.

A Equação 3.5-1 é descrita de acordo com os seguintes elementos:

- P- Escore do estudante após a conclusão da tarefa;
- pi- Escore máximo de acordo com o tipo do i-ésimo relacionamento (5 pontos para cada relacionamento importante e 2 para cada relacionamento menos importante);
- ci- Coeficiente que corresponde ao grau de correção do i-ésimo relacionamento;
- n- Número de relacionamentos na estrutura do Mapa Conceitual;
- cs – Penalidade para cada vez que ocorre redução de dificuldade;
- s – Numero de vezes que houve redução de dificuldade;
- a – Penalidade pela inserção do primeiro conceito pelo sistema;
- j – Número total de conceitos inseridos pelo sistema;
- m – Número total de conceitos no Mapa Conceitual;
- Δ - Aumento de penalidade a cada inserção de conceito.

Assim como há a possibilidade de aumentar o nível de dificuldade de uma tarefa, o sistema também disponibiliza a capacidade de diminuir este nível. Entretanto ao realizar esta escolha, o sistema penaliza o estudante com a descontagem de pontos em seu escore.

Outra maneira de avaliar o estudante, nesta abordagem, é através de várias tarefas com diferentes níveis de dificuldade. Estas tarefas estão divididas entre complementação e construção de MCs e o escore do estudante avalia a solução da tarefa do estudante através da seguinte equação:

$$P = \sum_{i=1}^n l k_i * p_i * c_i$$

Equação 3.5-2. Equação de cálculo de escore sobre a solução da tarefa.

Os elementos constituintes da equação X.2 são:

- P- Escore do estudante para a dada tarefa;
- lki- Coeficiente do grau de dificuldade para a dada tarefa;
- pi- Escore máximo de acordo com o tipo do i-ésimo relacionamento (5 pontos para o relacionamento mais importante e 2 pontos para o menos importante);
- ci- Coeficiente que corresponde ao grau de correção do i-ésimo relacionamento;
- n- Número de relacionamentos no mapa conceitual de referência;

Resultados. A validação da proposta foi realizada através do uso de dois protótipos do sistema, onde cada um abordava uma das formas de mudança de grau de dificuldade: o primeiro com inserção de conceitos adicionais como aumento do grau de dificuldade e o segundo oferta diferentes tipos de questões como aumento do grau de dificuldade. Ambos os protótipos foram avaliados em cursos de aprendizagem, onde um questionário foi oferecido aos estudantes que participaram da avaliação. A finalidade deste questionário residia em analisar a opinião dos alunos acerca da metodologia de avaliação aplicada, bem como a possibilidade de mudança de grau de dificuldade.

Resultados para o protótipo de inserção de conceitos adicionais. Este protótipo foi avaliado por 44 alunos, sendo que destes 44 apenas 35 deles responderam ao questionário após usarem o sistema. Em média 60% dos que responderam ao

questionário consideraram a tarefa difícil e apenas 11% afirmaram ser muito difícil. Desta amostragem, somente um terço dos estudantes utilizou o método da redução de dificuldade, desses quais 80% dos estudantes que a utilizaram classificaram-na como útil, uma vez que facilitava o desempenho da tarefa.

Resultados para o protótipo de diferentes questões. Este protótipo que oferta diferentes tipos de questão como mudança do grau de dificuldade foi avaliado através da utilização de um MC de três estágios. Este protótipo foi aplicado a um único curso de aprendizagem com 30 alunos, sendo que apenas 28 completaram o questionário que lhes foi entregue. Neste caso os alunos explicitaram que este tipo de avaliação não era utilizado por outros cursos e, portanto, não era usual. Em média 43% dos estudantes tiveram dificuldade de realizar a tarefa, porém desta amostragem apenas 75% destes reduziram o grau de dificuldade.

Em ambos os protótipos os alunos que afirmaram não utilizar a redução do grau de dificuldade, o fizeram por não querer reduzir o escore da tarefa. Em média 92% dos estudantes verificaram que após a redução do grau de dificuldade a tarefa oferecida se tornava mais simples de solucionar. O grau de dificuldade foi aumentado por quase um terço dos estudantes após a conclusão satisfatória da tarefa no estágio anterior, sendo que deste grupo 89% verificou que a tarefa tornou-se mais difícil de ser realizada, principalmente se a tarefa envolvia a construção do MC.

3.6. Considerações Finais

A avaliação de MCs é realizada de maneiras distintas de acordo com o padrão estabelecido como meta de avaliação. A meta de avaliação considera os quais componentes dentro de uma construção de MC são determinantes para a geração de um escore compatível. Entretanto, embora a avaliação seja realizada sobre diferentes aspectos para cada ferramenta, algumas características são consideradas comuns na análise das características relativas à avaliação, tais como os tipos de avaliação abordados, as metodologias de avaliação utilizadas, entre outros. A Tabela 3.6-1 apresenta as ferramentas descritas ao longo deste capítulo e quais as principais características relacionadas à avaliação de MCs que são contempladas por elas.

Tabela 3.6-1. Tabela comparativa entre sistemas de avaliação automática de MCs.

		Ferramenta de Hiperfídia para Questões Interdisciplinares	Sistema de Avaliação Baseado em Conhecimento	COMPASS	CMTool
Características das ferramentas	Avaliação Quantitativa	OK	OK	OK	X
	Avaliação Qualitativa (<i>Feedback</i>)	X	OK	OK	OK
	Metodologia Altamente Dirigida	OK	X	OK	X
	Metodologia Fracamente Dirigida	OK	OK	X	OK
	Utilização de 1 Mapa de Referência	OK	OK	OK	OK
	Utilização de vários Mapas de Referência	X	OK ¹	OK	OK
	Múltiplas Submissões para a mesma tarefa	X	OK ²	OK	X
Métodos de Avaliação	Avaliação do processo de ancoragem	X	X	X	OK
	Avaliação de generalidade e inclusividade	X	X	X	OK
	Avaliação de conceitos ou proposições	OK	OK	OK	OK
Dados	Acurácia	-	-	85%	-
	Usabilidade	-	47%	94%	-

¹ É utilizado um MC de referência para cada etapa de aprendizagem da ferramenta.

² É aceita uma submissão para cada etapa de aprendizagem da ferramenta.

4. Capítulo

Ferramenta de Avaliação de Mapas Conceituais para Ambientes de EAD

Este capítulo explora a ferramenta proposta, detalhando o processo de avaliação e relatando os resultados alcançados de acordo com a pesquisa realizada.

4.1. Introdução à Ferramenta de Avaliação de Mapas Conceituais

O processo de análise utilizado pela ferramenta proposta tem como propósito realizar uma avaliação simples e direta a fim de auxiliar estudantes e docentes dentro de um ambiente AVA. Entretanto, esta ferramenta de avaliação é independente de um AVA específico, podendo ser integrada a um que ofereça tarefas que envolvam a utilização de MCs. A avaliação é realizada de duas maneiras:

- Quantitativa através de um escore dado ao MC do estudante;
- Qualitativa através de um “relatório-guia” fornecido ao estudante após o desenvolvimento de seu MC.

A metodologia de avaliação utilizada está focada em uma análise sobre o produto desenvolvido pelo aluno. Esta se encontra dividida em três etapas:

- Processo de extração de informações do MC;
- Processo de avaliação quantitativa através da análise de similaridade entre o MC do aluno e MC modelo;
- Processo de avaliação qualitativa através da criação do relatório-guia.

4.2. Processo de Extração de Informação dos Mapas Conceituais

A extração de informações ocorre através da leitura de um documento XML gerado a partir da estrutura do MC. A ferramenta lê o XML, que busca pelos elementos simples, que são os conceitos e links, para, em seguida, buscar pelas proposições formadas por estes conceitos e links. O processo de extração de informações é realizado em duas formas:

- Mapeamento: Os elementos da estrutura XML gerada para o MC são mapeados para dentro da ferramenta;
- Atribuição: Os elementos XML mapeados são selecionados e separados de acordo com suas respectivas características. Assim, de acordo com suas características, as informações presentes nesses elementos são tratadas para que possam ser analisada de forma adequada no processo de avaliação.

O aspecto mais importante do processo de extração de informação é o tratamento das informações, que separa conceitos e links para depois compor cada proposição separadamente. Ao final do tratamento das informações do MC, as

proposições compostas são armazenadas no sistema em um catálogo de proposições temporário. O catálogo de proposições é gerado individualmente para cada MC, como produto final do processo de extração de informação, com a finalidade de analisar separadamente cada um deles contra os MCs modelo.

4.2.1. Normalização das informações

Antes de serem colocadas para avaliação, as informações extraídas dos MCs são submetidas a uma normalização a fim de se tornarem aptos a serem avaliados. Esta normalização tem a finalidade de aumentar a acurácia no que concerne ao significado das palavras descritas através de dois processos:

- Stemming: Realiza a extração do radical das palavras;
- Dicionários de sinônimos: Busca pelo significado das palavras.

As palavras existentes nas proposições dos MCs do aluno e dos MCs de referência são varridas e comparadas, caso seja encontrada alguma palavra no MC do aluno que não se encontre no MC modelo, o dicionário é acionado. O dicionário tem como finalidade realizar uma pesquisa sobre uma dada palavra em uma base de dados e verificar se existe alguma equivalente no MC modelo. Assim, se houver uma palavra equivalente no dicionário e que esteja de acordo com aquela presente no MC modelo, ela será substituída no MC do aluno.

O processo de normalização de sinônimos gera como resultado um novo formato de texto com as mesmas informações do formato anterior, porém contendo as mudanças realizadas após sua aplicação. Sobre esse novo formato é aplicado o stemmer, onde as palavras são analisadas e seu radical extraído. Assim, o stemmer permite que palavras que possuem o mesmo radical e significado, e que, entretanto, estejam graficamente diferentes sejam consideradas como iguais. Durante este processo o resultado do stemming modifica uma última vez o formato de texto. Este formato final (Figura Figura 4.2.1-1) é denominado forma canônica e sobre esta forma será realizada a avaliação.

O formato de texto produzido obedece ao mesmo princípio utilizado no formato produzido após a extração de informação, ou seja, baseado em proposições. As proposições possuem formação de uma estrutura frasal simples através do formato

<SUJEITO><VERBO><OBJETO>, onde tanto sujeito como objeto são considerados conceitos e os verbos são considerados *links* dentro do contexto das proposições.

MODELO (Conceito-Link-Conceito)	MODELO NORMALIZADO
[chave primaria identifica registros]	[chav prim identific registr]
[chave primaria pode ser composta]	[chav prim pod ser compost]
[chave primaria possui valor único]	[chav prim possu val unic]
[valor único não deve ser nulo]	[val unic nao pod ser nul]

Figura 4.2.1-1. Exemplo de normalização realizado durante processamento de extração de informações.

4.3. Processo de Avaliação Quantitativa

A avaliação quantitativa tem como finalidade a geração de um escore para um MC desenvolvido por um estudante. A fim de atingir este propósito, é feita uma análise de similaridade utilizando a técnica de N-gramas sobre o MC do estudante. Esta análise de similaridade é realizada comparando as informações extraídas dos MCs dos estudantes com as informações dos MCs que servem como modelos de avaliação.

Os MCs modelo são aqueles considerados os mais próximos da resposta correta, sendo utilizados como MCs de referência na avaliação. Assim, a avaliação quantitativa adota uma abordagem de aprendizagem na qual não há uma única resposta correta para uma questão.

Os MCs modelo são formados a partir da seleção dentre os que mais se aproximam (mais corretos) e os que menos se assemelham ao MC de referência (menos corretos). Esta seleção é realizada de acordo com os seguintes passos:

- Os MCs dos estudantes passam por uma comparação preliminar com o MC de referência;
- Os MCs que mais se aproximam da MC de referência são utilizados para formar o modelo com os melhores MCs;
- Os MCs que mais se distanciam do MC de referência são utilizados para formar o modelo com os piores MCs.

A técnica de N-gramas (Kondrak, 2005), utilizada no sistema, calcula o índice de similaridade examinando as semelhanças entre conceitos e links existentes nos MCs. A análise de bi-gramas é realizada na ferramenta através das duplas de conceito-link ou

link-conceito, enquanto os arcos de proposição conceito-link-conceito ou link-conceito-link são a representação dos tri-gramas.

O resultado da análise de N-gramas é um conjunto de índices que representam os graus de similaridade entre os MCs desenvolvidos pelos alunos e os MCs modelo, ou seja, cada MC recebe um índice de similaridade. Estes índices são submetidos ao método de predição KNN (K-Nearest Neighbor), que a partir dos índices atribui um escore ao MC desenvolvido pelo aluno. O fluxo do processo de avaliação quantitativa pode ser visualizado de acordo com a Figura 4.3-1.

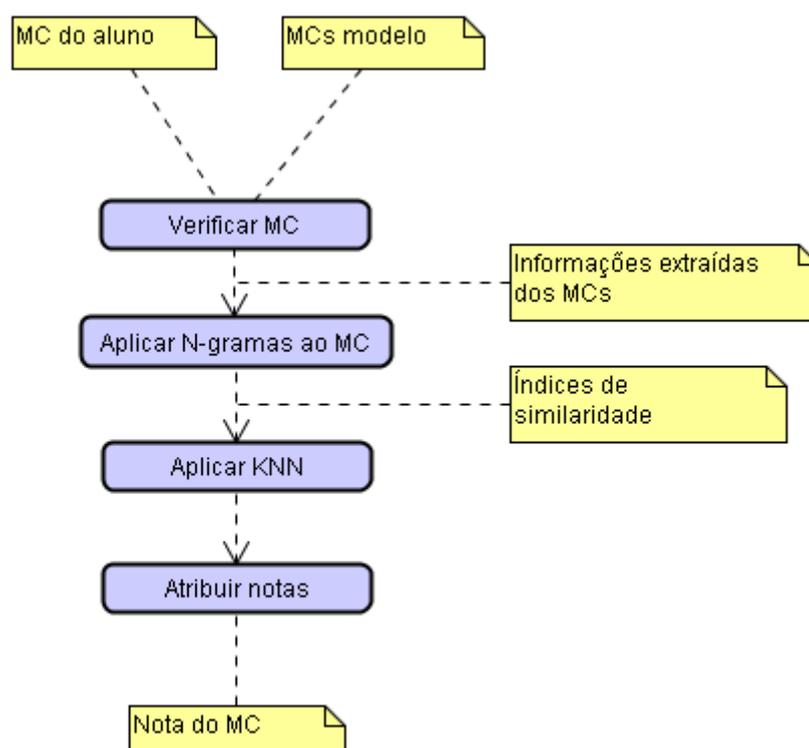


Figura 4.3-1 Fluxo do processo de avaliação quantitativa (modificar os MCs para cima e a nota do MC para baixo)

O processo de avaliação quantitativa pode ser mostrado através de um exemplo que ilustra as entradas e resultados gerados durante a avaliação. A Figura 4.3-2 mostra as entradas para o sistema.

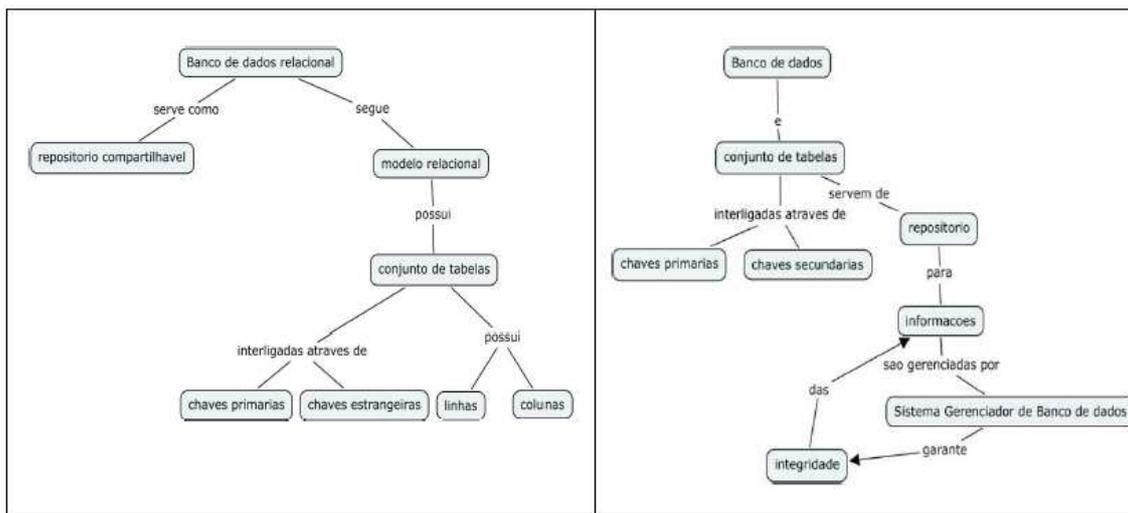


Figura 4.3-2. MC Modelo de resposta (à esquerda) e exemplo de MC de aluno (à direita).

Os MCs apresentados na Figura 4.3-2 representam as entradas no sistema. Os MCs modelo se encontram armazenados em um repositório e são utilizados a cada avaliação realizada. Ao serem submetidos, tanto o MC modelo como o MC do aluno tem suas informações extraídas e sobre elas é aplicada a técnica de N-gramas.

A técnica de N-gramas é utilizada de três formas a fim de verificar o grau de similaridade do MC do aluno com o MC do docente: uni-grama, bi-grama e tri-grama. Cada MC recebe estes três valores de similaridade, que são responsáveis pela avaliação e conseqüente atribuição de nota ao MC desenvolvido. Uma amostra dos valores de similaridade gerados pela técnica de N-gramas pode ser observada na Tabela 3.2-1.

Tabela 4.3-1. Amostra do relatório de saída do processamento de N-gramas sobre os MCs dos alunos.

MCs	UNIGRAMAS	BIGRAMAS	TRIGRAMAS
MC1	0,12963	0,036364	0
MC2	0,15534	0,095238	0,018692
MC3	0,10101	0,039604	0,019417
MC4	0,275229	0,198198	0,123894
MC5	0,103093	0,040404	0,019802
MC6	0,324324	0,19469	0,052174
MC7	0,080808	0,059406	0,019417
MC8	0,247619	0,205607	0,165138
MC9	0,336134	0,264463	0,211382
MC10	0,060606	0,039604	0
MC11	0,084211	0,082474	0,060606
MC12	0,336134	0,231405	0,146341
MC13	0,188679	0,074074	0,036364
MC14	0,150943	0,055556	0
MC15	0,061224	0,04	0,019608

A Tabela 4.3-1 apresenta dados de similaridade para uma determinada avaliação. Estes dados variam entre 0 e 1, onde os valores mais próximos de 1 indicam maior similaridade com os MCs modelo. Estes dados são repassados para o módulo KNN, que analisa cada linha como um mesmo MC. Cada variável da linha que representa o MC é analisada de acordo com seu grau de influência sobre o resultado para então atribuir uma nota.

4.4. Processo de Avaliação Qualitativa

A avaliação qualitativa é responsável pelo auxílio ao estudante com a finalidade de levá-lo à melhor solução para o problema que é apresentado através de um relatório-guia. O processo de avaliação qualitativa também depende de MCs modelos. Entretanto, no processo de avaliação qualitativa são trabalhados dois tipos de modelo de MCs: os considerados mais corretos e os considerados mais menos corretos.

A idéia é utilizar ambos os modelos para realizar a avaliação qualitativa. Esta avaliação consiste em contrastar o MC do estudante com os MCs modelo selecionados e gerar um relatório-guia com as proposições que faltam no MC do aluno. Referindo-se aos MCs considerados mais corretos, as proposições são importantes no relatório para auxiliar ao aluno na descoberta ou fixação do conhecimento. Os MCs considerados como mais menos corretos são comparados com os MCs mais corretos e as proposições que forem consideradas incorretas ou irrelevantes preenchem o relatório-guia, servindo como ruído. Assim, ao executar uma tarefa, o estudante se depara com certo grau de dificuldade, levando-o a buscar pela solução ideal.

O relatório-guia é composto por todas as informações que não foram consideradas no MC do estudante e que estão presentes nos MCs modelo. Estas informações encontram-se disponíveis de maneira aleatória no relatório-guia. Assim, esta abordagem permite que o aluno trabalhe sobre suas dificuldades de modo a evoluir seu aprendizado e seu conhecimento a respeito do assunto abordado. Nesse sentido, a ferramenta proposta aceita múltiplas submissões de respostas, permitindo ao aluno utilizar-se das informações existentes no relatório-guia para refinar sua solução.

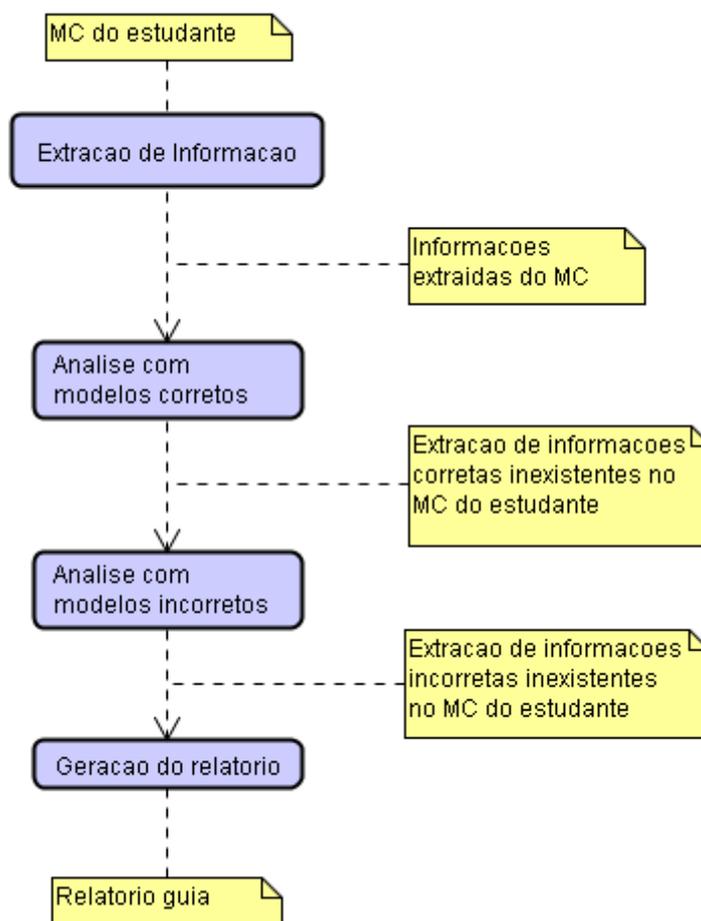


Figura 4.4-1. Fluxograma do processo de avaliação qualitativa.

O processo de avaliação qualitativa (Figura 4.4-1) ocorre após o término de cada tarefa realizada pelo estudante, que submete sua solução como entrada no sistema. Na primeira etapa deste processo são analisadas as proposições dos MCs modelo corretos, onde as proposições não consideradas pelo estudante são armazenadas. Em seguida, são analisados os MCs menos corretos, que passam pelo mesmo procedimento utilizado para os MCs corretos e suas proposições são armazenadas juntamente com às dos MCs corretos. Assim, o relatório-guia é preenchido de maneira aleatória com as proposições que foram armazenadas.

A avaliação qualitativa pode ser observada através de um exemplo, que represente seu processo através da apresentação dos dados gerados durante este processo. A Figura 4.4-2 representa a entrada de um MC de aluno para a resolução de um exercício que sofrerá a avaliação qualitativa.



Figura 4.4-2. MC de aluno a ser avaliado.

Ao receber o MC da figura 2, o sistema consulta o repositório de MCs (corretos e errôneos) e processa suas informações, em seguida as informações do MC submetidos são analisadas e processadas gerando para o sistema o resultado mostrado na figura 4.4-3.

<p>MODELO ALUNO</p> <p>[agrupamento de atributos possuem valores], [chave primaria deve possuir conteudo reduzido], [chave primaria e agrupamento de atributos], [chave primaria e identificador exclusivo], [chave primaria e atributo], [chave primaria deve possuir valor constante], [valores identificam tuplas]</p>	<p>MODELO ALUNO NORMALIZADO</p> <p>[agrup de atribut possu val] [chav prim dev possu conteud reduz] [chav prim e agrup de atribut] [chav prim e identific exclus] [chav prim e atribut] [chav prim dev possu val const] [val identific tupl]</p>
<p>MODELO REFERÊNCIA</p> <p>[chave primaria e coluna], [chave primaria identifica cada registro], [chave primaria informa valores], [chave primaria pode ser composta], [chave primaria possui valor único], [coluna possui clausula not null], [valor deve ser único], [valor nao pode ser nulo],</p>	<p>MODELO REFERÊNCIA NORMALIZADO</p> <p>[chav prim e colon] [chav prim identific cad registr] [chav prim inform val] [chav prim pod ser compost] [chav prim possu val unic] [colun possu clausul not null] [val dev ser unic] [val nao pod ser nul]</p>

Figura 4.4-3. Exemplos de resultados do processamento do MC do aluno e dos modelos de referência para o sistema.

A figura 4.4-3 mostra a extração das proposições tanto de um MC de estudante quanto do MC modelo de referência, que é composto pelas proposições dos MCs considerados corretos. O sistema realiza a análise de N-gramas sobre estas proposições e verifica quais as triplas que não são contempladas pelo MC do aluno; em seguida o sistema extrai as proposições dos modelos considerados errôneos. Assim, o relatório-guia é preenchido tanto com as proposições não consideradas pelo estudante como pelas proposições consideradas errôneas, gerando um *feedback* ao estudante como apresentado na figura 4.4-4.

<p>RELATÓRIO (Proposições Inexistentes)</p> <p>[chave primaria e coluna], [chave primaria identifica registro]* [chave primaria informa valores], [chave primaria pode ser composta]* [coluna possui clausula not null], [chave primaria possui valor], [registros devem ter valor unico]*, [unico para cada registro]* [valor deve ser unico]* [valor nao pode ser nulo]*</p>

Figura 4.4-4. Relatório-guia processado ao final da avaliação qualitativa, onde as proposições com “*” são as proposições consideradas corretas incluídas no relatório.

4.5. Etapas de Avaliação

O processo de avaliação consiste em etapas distintas, cada uma com seu próprio sub-processo, como pode ser observado no fluxo da Figura 4.5-1. Este processo envolve ambos os níveis de análise sobre os MCs dos estudantes, ou seja, abrangem tanto a avaliação quantitativa, representada pela etapa com a responsabilidade de atribuição de escores, como a qualitativa, que é caracterizada pela etapa cuja finalidade é gerar o relatório de diagnóstico.

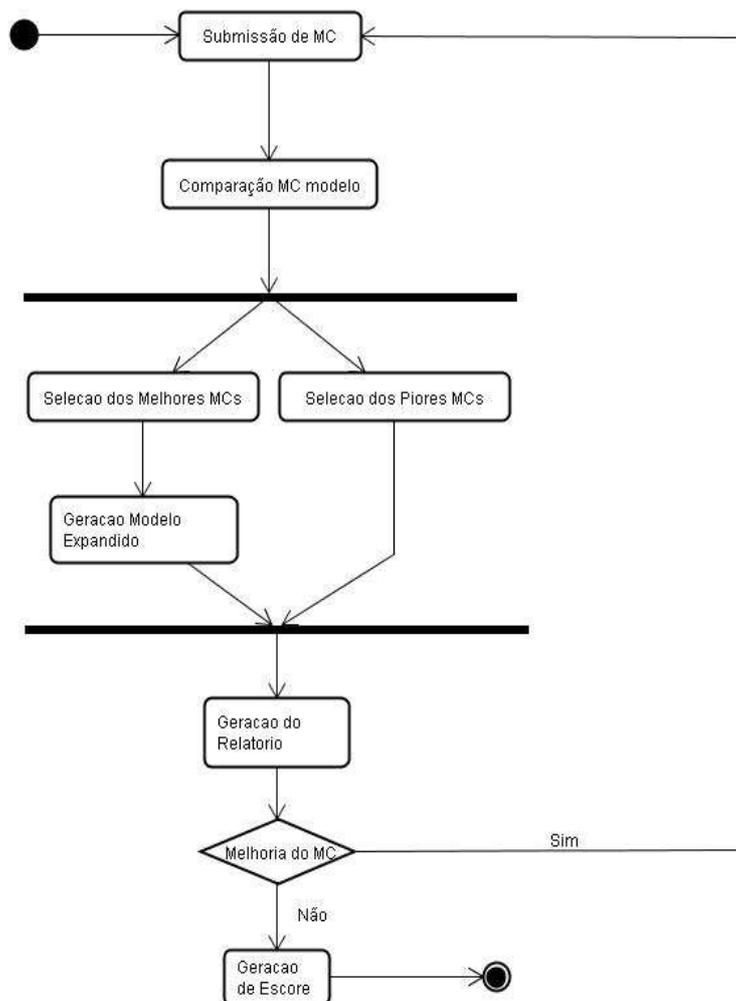


Figura 4.5-1. Fluxograma do processo de avaliação

A Figura 4.5-1 apresenta de maneira geral todo o processo de avaliação dos MCs de acordo com as etapas que descrevem desde a submissão dos MCs, passando pela seleção das melhores e das piores soluções e pela avaliação através do modelo expandido, culminando na geração do relatório e na geração do escore.

O fluxograma mostra ainda que as etapas de geração do relatório-guia e geração do escore são as últimas no processo de avaliação. Elas são precedidas por outras etapas, que tem a responsabilidade de receber os dados e tratá-los de acordo com as seguintes regras especificadas para cada uma:

- Primeira etapa: Os MCs desenvolvidos pelos alunos são comparados a um MC modelo de referência criado pelo docente ou especialista. Ao final deste sub-processo são selecionados, de acordo com os escores, os N melhores e os N piores MCs. Este N não é definido como uma constante, variando de a 4 MCs;

- Segunda etapa: Um modelo de resposta com **vocabulário expandido** é criado a partir do mapa modelo de referência em conjunto com os 2 a 4 MCs escolhidos no processo anterior como os melhores MCs.

4.5.1. Etapas da avaliação qualitativa

A avaliação qualitativa representa o processo de geração do relatório-guia. Ela corresponde à terceira etapa da avaliação de MCs, possuindo três sub-etapas de análise:

- a) O MC do aluno é comparado contra o modelo com vocabulário expandido, identificando-se todos os bigramas e trigramas que estão num modelo e não no outro e armazenando as informações que não se encontram no MC do aluno para serem colocadas no relatório-guia;
- b) Os MCs selecionados como menos corretos no primeiro processo são analisados e suas informações são extraídas a fim de que suas informações sejam colocadas no relatório-guia;
- c) As informações obtidas nas fases anteriores são mescladas e colocadas nos relatórios de diagnóstico.

4.5.2. Etapa de avaliação quantitativa

A etapa de avaliação quantitativa corresponde à quarta etapa do processo de avaliação. Nesta etapa, uma nota é gerada utilizando o método de classificação KNN (K vizinhos mais próximos). Cada MC de aluno é comparado ao modelo de vocabulário expandido registrando-se as métricas de bi-gramas e tri-gramas. A partir das medidas, os K-vizinhos mais próximos são selecionados para produzir a nota.

4.6. Arquitetura do sistema

A arquitetura da ferramenta proposta é composta por quatro módulos, onde cada módulo é interdependente do módulo central que se refere à avaliação propriamente dita. O detalhamento da arquitetura apresentada na Figura 4.6-1 é explicado no capítulo 5, que descreve as principais características de cada módulo, bem como detalhes de implementação.

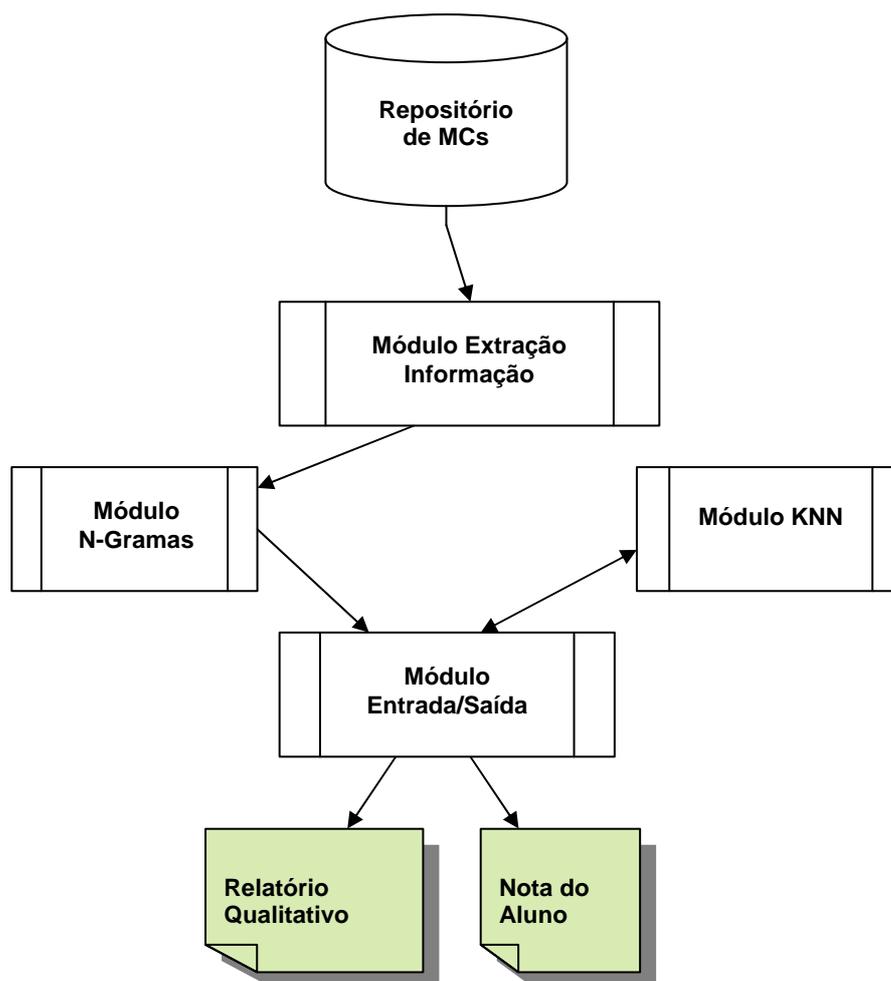


Figura 4.6-1. Arquitetura da ferramenta de avaliação automática.

4.7. Considerações

A idéia da ferramenta proposta é realizar uma avaliação baseada no aprendizado do estudante, ou seja, a ferramenta considera a evolução de conhecimento do aluno. Esta meta é alcançada através da possibilidade de múltiplas submissões para a mesma questão.

O modelo de avaliação foi idealizado em etapas, onde cada etapa representa uma funcionalidade, cujo resultado depende do resultado da etapa anterior, trabalhando em cadeia. Esse modelo é refletido na implementação, que separa as etapas em módulos, onde cada módulo é responsável por uma parte da tarefa de avaliação.

A ferramenta tem ainda a vantagem de ser adaptável a qualquer ambiente de EAD que possua Mapas Conceituais como modelo de avaliação, desde que exista um

modelo XML correspondente ao que foi desenvolvido pelo aluno. A etapa que corresponde à extração de informação foi criada para reconhecer qualquer tipo de XML que descreva os principais componentes de um MC. Este evento pode ser melhor observado no capítulo 5 que relata de forma detalhada a implementação das principais funcionalidades da ferramenta.

5. Implementação

O capítulo 5 explora o desenvolvimento da ferramenta proposta, detalhando as principais idéias acerca de sua construção. Neste capítulo são detalhados os componentes da arquitetura, cujos códigos são mostrados para ilustrar a funcionalidade de cada módulo.

5.1. Introdução

Este capítulo explana sobre o processo de construção da ferramenta proposta para avaliação. Como descrito anteriormente, a ferramenta possui quatro módulos, onde o primeiro deles é responsável pela extração de informações, que serão submetidas à avaliação. O segundo módulo envolve a execução da análise de similaridade através de N-Gramas. O terceiro módulo é o módulo responsável pelo cálculo do KNN. O quarto módulo é responsável pela entrada e saída de informações relativas à avaliação tais como: relatório de similaridade a partir dos resultados da análise de n-gramas, saída do relatório-guia para os estudantes.

Neste capítulo são descritos apenas os três primeiros módulos, pois são estes que sintetizam a idéia proposta. Para cada módulo são explicadas as descrições e os objetivos de cada classe e método utilizado e qual seu objetivo para aquele módulo. As classes e métodos mostrados neste capítulo têm como objetivo apresentar partes cruciais do desenvolvimento, a fim de mostrar a lógica do processo adotado para realizar a avaliação.

5.2. Módulo de Extração de Informação

O principal objetivo deste módulo é extrair as informações existentes em um MC. O processo de extração de informação é realizado através da geração de um modelo XML que representa efetivamente a estrutura de um MC desenvolvido pelo professor ou pelo aluno. Os modelos XML são salvos em repositórios. Este repositório utilizado para armazenar os modelos de referência é distinto do repositório destinado aos modelos desenvolvidos pelos alunos. Uma estruturas XML de um modelo é lida de seus repositórios e passa por uma análise que busca pelos conceitos e palavras de ligação existentes. Ao encontrar as informações inerentes ao conhecimento existente na estrutura XML, ela é repassada para o Módulo de N-Gramas.

A parte primordial do módulo de Extração de Informação é a classe Mapeamento, que realiza desde o reconhecimento do arquivo XML à geração do relatório-guia, relacionando-se desta maneira com o módulo de Entrada/Saída. A partir da classe Mapeamento é que se inicia todo o processo de avaliação, pois ela realiza o mapeamento dos elementos do arquivo XML e os transforma em objetos reconhecíveis

pelo sistema de modo que esses objetos auxiliem no processo de avaliação. Estes objetos são representados por duas outras classes denominadas Forma e Ligação, onde Forma representa tantos os conceitos como os links existentes no MC, enquanto Ligação representa as proposições formadas por estes conceitos e links.

A fim de realizar o mapeamento do arquivo XML e a partir deste gerar uma estrutura passível de avaliação, a classe Mapeamento dispõe de dois métodos: `compararLigacoesString()` e `compararLigacoesArray()` (anexo F), que varrem toda a estrutura XML e ao final retornam todas as proposições existentes no MC do aluno em formato de texto. A diferença entre os métodos reside na utilização do dicionário de sinônimos, onde o segundo método foi projetado para possibilitar substituição de palavras, caso seja verificado que no MC modelo a palavra possui uma equivalente.

Os métodos `compararLigacoesString()` e `compararLigacoesArray()` têm como objetivo principal montar as proposições de acordo com os objetos Forma e Ligacoes, catalogados durante a varredura do documento XML analisado. Essa varredura é realizada ao chamar dois métodos existentes `listarConceitos()` e `listarLigacoes()` (anexo G). Estes métodos realizam uma busca: o primeiro por um conceito e o segundo por uma palavra de ligação. A entrada é o id de uma palavra que pode vir a ser um conceito ou *link* e o retorno é um objeto do tipo Forma, cujo id seja relativo ao parâmetro utilizado. Um objeto do tipo Forma serve para representar tanto um conceito como um *link*.

O método `compararLigacaoString()` é utilizado por outros dois métodos: `obterFraseMCPorString()` e `obterFraseMCPorArray()`. Estes métodos são utilizados dentro da classe de Mapeamento para realizar a conexão com o arquivo XML que será lido e mapeado em seguida dentro do método `compararLigacaoString()`, a diferença entre os métodos é proveniente da utilização do dicionário de sinônimos que necessita de uma pesquisa acerca de cada palavra existente na estrutura desenvolvida pelo aluno. A Figura 5.2-1 detalha a construção destes dois métodos.

```

/**
 * Método obterFraseMCPorArray
 */
public String obterFraseMCPorString(File arquivo) {
    NodeList children = conexaoArquivo.getElementos(arquivo);
    StringBuilder builder = compararLigacoesString(children);
    return builder.toString();
}

/**
 * Método obterFraseMCPorArray
 */
public ArrayList<String> obterFraseMCPorArray(File arquivo) {
    ArrayList<String> frase = new ArrayList<String>();
    NodeList children = conexaoArquivo.getElementos(arquivo);
    frase = compararLigacoesArray(children);
    return frase;
}

```

Figura 5.2-1. Métodos obterFraseMCPorString() e obterFraseMCPorArray() da classe Mapeamento.

Os métodos apresentados na Figura 5.2-1 recebem como entrada o arquivo XML a ser mapeado. Em seguida, utilizando o método compararLigacoesString(), obtém a frase completa do MC presente no arquivo desenvolvido pelo aluno, retornando ao final um texto com todas as proposições desenvolvidas no MC.

Outra funcionalidade da classe Mapeamento reside na geração do relatório-guia que ocorre a cada avaliação realizada sobre os MCs dos estudantes. O método responsável é o compararLigacoesRelatorio() (anexo H). Este método tem a função de gerar um relatório para servir como base de comparação. Esta comparação tem como finalidade verificar quais conceitos estão em falta no MC do estudante. O resultado do mapeamento dos MCs, tanto dos modelos de referência quanto dos desenvolvidos pelos estudantes, serve como entrada para o módulo N-Grama.

O resultado do processamento de extração de informação é o arranjo de todas as proposições descritas nos MCs submetidos. Este resultado é saída do módulo Extração de Informação e entrada para o módulo N-gramas, onde são enviadas as proposições extraídas tanto do MC do estudante como dos MCs modelo.

<p>MODELO ALUNO (Conceito-Link-Conceito)</p> <p>[chave primaria e identificador exclusivo], [chave primaria e agrupamento de atributos],[valores identificam tuplas], [chave primaria e atributo], [chave primaria deve possuir valor constante], [identificador exclusivo de cada registro], [agrupamento de atributos possuem valores], [cada registro na tabela], [chave primaria deve possuir conteudo reduzido]</p> <p>MODELO REFERÊNCIA (Conceito-Link-Conceito)</p> <p>[valor deve ser único], [chave primaria possui valor unico para cada registro], [chave primaria pode ser composta], [chave primaria identifica cada registro], [valor nao pode ser nulo], [coluna possui clausula not null], [chave primaria informa, valores], [valores não podem ser nulos], [chave primaria e coluna], [registro dentro de uma tabela], [chave primária identifica registro]</p>

Figura 5.2-2. Saída do processamento de extração de informação para um MC de aluno e MC modelo.

5.3. Módulo N-Gramas

O Módulo de N-Gramas é responsável pela verificação da similaridade existente entre as informações presentes nos MCs dos estudantes e MCs modelo de referência, que são provenientes do módulo de Extração de Informação. A verificação de similaridade é realizada juntamente com a utilização de um dicionário de sinônimos que tem como finalidade tornar flexível a linguagem utilizada pelo aluno, de modo que esta não venha a afetar a análise. A verificação de similaridade divide-se em duas etapas:

- Primeira etapa: As informações existentes no MC do aluno são comparadas contra as informações de apenas um modelo de MC;
- Segunda etapa: As informações existentes no MC do aluno são comparadas contra as informações de dois ou mais modelos de MC;

Em ambos os casos o princípio é o mesmo: analisar o grau de similaridade. Entretanto, na primeira etapa a finalidade é encontrar MCs cujas informações sejam as mais próximas das desejadas na tarefa promovida pelo professor. Ao encontrar estes MCs, eles são movidos para o repositório dos modelos a fim de formarem, junto com o MC modelo inicial, um vocabulário expandido.

Nesta etapa também são verificados os MCs cujas informações se encontram o mais distante das desejadas na tarefa desenvolvida pelo professor. Assim, os MCs que se encontram nesta situação são coletados para o repositório de MCs modelo, porém são

reconhecidos como modelos “errôneos”. As informações existentes nestes modelos são utilizadas para gerar o relatório de auxílio ao aluno.

Na segunda etapa, as informações do MC do aluno são comparadas contra o vocabulário expandido gerado na primeira etapa e o resultado gerado é o índice/grau de similaridade entre estas informações. Os índices de similaridade gerados para os MCs de cada aluno são repassados para o Módulo KNN.

O cálculo do índice de similaridade é realizado fazendo-se uma comparação entre o resultado do mapeamento do MC do aluno e o resultado do mapeamento dos MCs modelo de referência. As classes que compõem este módulo são.

A interface IAutomatico (Figura 5.3-1) é responsável pela intersecção entre o resultado do mapeamento dos MCs, que é realizado na classe Mapeamento, e a execução da similaridade sobre este resultado. A execução do cálculo de similaridade é realizada na classe NGrama, que é a principal classe do módulo de N-Gramas, pois nela encontram-se todas as regras necessárias à realização do cálculo de similaridade via n-gramas, bem como os modelos Dice e Sobreposição responsáveis por este cálculo.

```
/**
 * Interface IAutomatico
 */
public interface IAutomatico {

    public double analisarNGramaMc(File arquivoAluno, File
arquivoModelo, int gramas);

    public double analisarNGramaMc(File arquivoAluno, File[]
arquivoModelo, int gramas);
}}
```

Figura 5.3-1. Interface base de implementação da análise de N-Gramas.

A interface IAutomático possui dois métodos similares, porém com diferentes enfoques. Ambos os métodos buscam atribuir um valor de similaridade para um MC desenvolvido pelo aluno. Contudo, o modo como são realizados é distinto, pois enquanto o primeiro método compara um MC do aluno com um MC modelo em uma comparação um-para-um, o segundo método utiliza uma comparação um-para-muitos, isto é, entre um MC do aluno e um conjunto de MCs modelo.

Após este processamento, o resultado da análise de similaridade é repassado e com ele é gerado como saída um relatório contendo os escores de similaridade de todos os MCs submetidos ao sistema. Este relatório é repassado ao módulo KNN que o lê e prediz a nota de acordo com o grau de similaridade atribuído ao MC do aluno. A geração e importação deste relatório para o módulo KNN é feita através de classes pertencentes ao módulo de Entrada/Saída, que implementam a interface `IExportacao`, cuja principal funcionalidade reside em importar e exportar dados em formato de relatório no sistema.

5.4. Módulo KNN

O módulo KNN é responsável pela atribuição de notas para os MCs desenvolvidos pelos alunos. O relatório contendo os índices de N-Gramas é analisado e notas são atribuídas a cada MC analisado. Esta análise KNN sobre os índices de N-Gramas pode ser realizada de duas maneiras.

- **Treinamento/Teste:** É selecionada uma porcentagem dos dados para treinamento e outra para testes. Assim, os dados de Treinamento servem como base para realizar uma predição a fim de atribuir notas para os dados de Teste.
- **Cross Validation:** É selecionado um número que será a quantidade de partes em que o número total de elementos será dividido. Após essa divisão, é realizado um rodízio, onde cada parte, em sua respectiva vez, será considerada a porcentagem de treinamento e o restante dos elementos será considerado porcentagem de teste.

Ambas as formas foram implementadas e utilizadas para avaliar os MCs dos estudantes, obtendo médias próximas às obtidas através de avaliadores humanos. Entretanto, percebeu-se que os resultados provenientes da utilização de Cross Validation eram mais satisfatórios. Portanto, o Cross Validation foi adotado como método de KNN para este sistema de avaliação.

O módulo KNN é composto pelas classes `MapaConceitual`, `MapaConceitualCross`, `KnnCalculo` e pelas classes `KnnExecucao` e `KnnExecucaoCross` que implementam a interface `IKnnExecucao`. Uma das principais características deste

módulo é que o cálculo do KNN foi adaptado para realizar tanto a predição simples, que apenas separa os dados de treinamento e teste, como a predição com Cross Validation.

A relação das classes MapaConceitual e MapaConceitualCross está diretamente ligada a execução da classe responsável pelo cálculo de similaridade via n-gramas, que ao realizar suas operações atribui os respectivos valores de n-gramas no objeto, e às classes referentes ao KNN, que a partir dos atributos referentes a n-grama calculam a distância e a nota predita de acordo com os cálculos pertencentes ao módulo Knn.

```

**
* Classe MapaConceitual
*/
public class MapaConceitual {

    protected double unigrama;
    protected double bigrama;
    protected double trigrama;
    protected double notaAvaliador;
    protected double notaPredita;
    protected double distancia;

    public MapaConceitual() {
        this.notaPredita = 0;
    }

    ...

    // Métodos gets e sets
}

```

Figura 5.4-1. Classe MapaConceitual e seus atributos componentes.

As classes que representam os MCs (MapaConceitual e MapaConceitualCross) no módulo KNN são representadas pelos atributos de MC necessários à realização da predição da nota a ser atribuída, que também é um atributo do MC. Assim, os valores obtidos através da análise de n-gramas estão presentes, bem como a nota do avaliador, a distância obtida através do cálculo do KNN e a nota a ser predita baseada nesta distância.

A diferença entre ambas as classes que representam os MCs encontra-se na existência de um atributo que serve para verificar o status do MC, ou seja, se ele foi analisado pelo sistema. Este atributo se faz necessário no método cross-validation, que realiza uma espécie de rodízio sobre os dados de entrada com a finalidade de realizar a predição dos dados, porém estes dados não podem ser repetidos. A classe MapaConceitualCross pode ser visualizada na Figura 5.4-2.

```

/**
 * Classe MapaConceitualCross
 */
public class MapaConceitualCross extends MapaConceitual {

    private boolean analisado;

    /**
     * IConstructor que inicializa o MC como não analisado.
     */
    public MapaConceitualCross() {
        // TODO Auto-generated constructor stub
        super();
        analisado = false;
    }

    /**
     * Verifica o status do Mapa Conceitual.
     */
    public boolean isAnalisado() {
        return analisado;
    }

    /**
     * Seta o status de análise no Mapa Conceitual.
     */
    public void setAnalisado(boolean analisado) {
        this.analisado = analisado;
    }
}

```

Figura 5.4-2. Classe MapaConceitualCross, filha da classe MapaConceitual.

O cálculo de predição de notas é realizado através da interface IKnnExecucao (Figura 5.4-3), cuja primitiva é realizar uma separação em dados de treinamento e teste para em seguida realizar a atribuição de escores. Esta tarefa é implementada na classe KnnExecucao.

```

**
 * Interface IKnnExecucao
 */
public interface IKnnExecucao {
    /**
     * Método que separa as listas entre Mapas de teste e treinamento
     */
    public void separarLista();
    /**
     * Método que executa o algoritmo de calculo do KNN
     */
    public void executarAvaliacao(int k, int numeroVariaveis, double
    porcentagem, int folders, String caminhoArquivo);
}

```

Figura 5.4-3. Interface IKnnExecucao.

A interface `IKnnExecucao` separa os dados em treinamento e teste através do método `separarLista()` para, em seguida, realizar o cálculo do Knn sobre os dados, utilizando o método `executarAvaliacao()`. Este método é implementado nas classes `KnnExecucao` (anexo I), que utiliza a predição simples e a correspondente com o método `Cross Validation`, `KnnExecucaoCross` (anexo J).

A implementação método `executarAvaliacao()` utiliza ainda o método `calcularKnn()` listado na Figura 5.4-4, que tem o objetivo de atribuir as notas preditas aos MCs de teste. Este método pertence à classe `KnnCalculo`, sendo esta a mais importante do módulo `Knn`, pois é responsável pelo cálculo Knn em si. Nesta classe, a avaliação é realizada sobre os objetos `MapaConceitual`, onde seus dados são manipulados com a finalidade de predizer a nota a ser atribuída para um MC desenvolvido.

```
/**
 * Método que atribui notas preditas aos mapas de teste
 */
public void calcularKNN(List<T> treinamento, List<T> teste, int k, int
numeroVariaveis) {
    List<T> modelo = new ArrayList<T>();
    for (T testeItem : teste) {
        modelo = gerarModelo(treinamento, testeItem, k,
numeroVariaveis);
        double nota = calcularMedia(modelo, k);
        testeItem.setNotaPredita(nota);
    }
    this.mapasTeste = teste;
}
```

Figura 5.4-4. Método calcularKnn() da classe KnnCalculo.

No método `calcularKnn()` são sintetizadas funções necessárias à realização da predição de notas. Uma destas funções é implementada através do método `gerarModelo()` (anexo K), cuja funcionalidade é selecionar `K` respostas, cujas características sejam as mais próximas ao objeto de teste em avaliação. Para alcançar este objetivo são verificadas as listas de treinamento e o objeto de teste em questão. O número de variáveis a serem avaliadas e o `k` número de vizinhos a ser utilizado no cálculo também são verificados.

A necessidade de verificar a quantidade de variáveis que serão utilizadas está relacionada ao grau de influência de cada uma para a predição de dados, pois o número de variáveis é proporcional ao número de unidades dimensionais. Portanto, cada nova

variável é um novo fator de influência para a predição. Esta predição é realizada de acordo com o cálculo da distância Euclidiana (Equação 5.4-1), que tem como finalidade calcular a distância entre dois pontos.

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}.$$

Equação 5.4-1. Cálculo da Distância Euclidiana.

No sistema desenvolvido foram consideradas três variáveis para o cálculo da distância Euclidiana: Unigramas, Bigramas e Trigramas. Os cálculos existentes no sistema permitem contemplar diferentes combinações de variáveis para realizar a predição:

- Cálculo unidimensional: considera apenas uma variável. Este cálculo é representado pelos métodos analisarUnigrama(), analisarBigrama() e analisarTrigrama());
- Cálculo bidimensional: considera duas variáveis e é representado pelos métodos analisarUniBigrama(), analisarUniTrigrama() e analisarBiTrigrama());
- Cálculo tridimensional: considera todas as variáveis, sendo representado pelo método analisarUniBiTrigrama().

```
/**
 * Calcula a distância euclidiana de unigramas, bigramas e trigramas
 */
private void analisarUniBiTrigrama(List<T> treinamento, T
mapaConceitual) {
    for (T mc : treinamento) {
        double unigrama = Math.pow(
            mapaConceitual.getUnigrama() - mc.getUnigrama(), 2);
        double bigrama = Math.pow(
            mapaConceitual.getBigrama() - mc.getBigrama(), 2);
        double trigrama = Math.pow(
            mapaConceitual.getTrigrama() - mc.getTrigrama(), 2);
        mc.setDistancia(Math.sqrt(unigrama + bigrama + trigrama));
    }
}
```

Figura 5.4-5. Método de cálculo da distância euclidiana para unigrama, bigrama e trigrama.

A idéia da distância para este trabalho é verificar a distância entre as variáveis existentes, que pertencem ao conjunto de treinamento e comparar com as variáveis pertencentes aos objetos do conjunto de teste. A distância calculada é atribuída aos objetos MapaConceitual da lista de treinamento e, a partir desta distância, são selecionados os K objetos da lista de treinamento, cuja distância com o objeto de teste seja a menor. Esta ação é realizada através do método recuperarModelo() (anexo L), que ordena a lista de objetos de treinamento de acordo com a distância obtida em ordem crescente de distância. Os K primeiros objetos da lista são os objetos que serão selecionados como modelos para prever a nota do objeto de teste em questão.

O método calcularKnn() possui ainda o método calcularMedia() (anexo L), que utiliza-se dos valores dos objetos MapaConceitual, selecionados a partir do método gerarModelo(), para prever a nota do aluno. É realizada uma média entre os valores das notas do avaliador para a lista de modelos e o resultado é atribuído ao objeto de teste como a nota predita.

O resultado final deste módulo é um relatório complementar ao relatório de entrada. Trata-se de um relatório com informações existentes no relatório de entrada mais as informações das notas atribuídas pelo professor, juntamente com as notas previstas pelo sistema para cada MC. A Tabela 5.4-1 apresenta uma amostra deste relatório.

Tabela 5.4-1. Amostra do relatório de saída com as notas previstas para cada MC.

UNIGRAMAS	BIGRAMAS	TRIGRAMAS	NOTA_AVALIADOR	NOTA_PREDITA
0,134615	0,101911	0,06962	8	5,125
0,157729	0,125392	0,087227	8,25	7,583333333
0,182965	0,175549	0,161994	7,5	6,666666667
0,430518	0,428184	0,420485	9,75	5,791666667
0,338028	0,291317	0,239554	9	7,416666667
0,380952	0,37883	0,365651	9	8,958333333
0,386555	0,389972	0,387812	9,75	8,958333333

5.5. Módulo de Entrada/Saída

O módulo de Entrada/Saída representa toda e qualquer parte da ferramenta que esteja relacionada com eventos cuja função seja importação de informações ou geração de relatórios durante a análise de MCs. Entretanto, sua principal funcionalidade

encontra-se na integração entre a primeira parte da ferramenta que se refere à extração e manipulação da informação e a segunda que se refere à avaliação dessas informações.

A parte do módulo que lida com relatórios tem uma funcionalidade mais abrangente, pois é responsável não somente pela leitura, mas também pela geração de relatórios. A geração de relatórios é a principal característica deste módulo, havendo dois tipos de relatórios gerados pela ferramenta: (i) relatório quantitativo; (ii) relatório qualitativo.

O relatório quantitativo lida com os dados provenientes da análise realizada no módulo N-Gramas, onde estes dados são exportados em formato de relatório, que serve como entrada no módulo KNN. Em seguida, este relatório é importado a fim de atribuir notas para os MCs de acordo com os dados recebido.

O relatório quantitativo está relacionado à etapa da análise quantitativa, onde são pesquisadas informações de acordo com o assunto abordado na tarefa. O relatório é gerado com as informações presentes nos MCs de referência (tanto os considerados corretos como os considerados inadequados) e que não foram abordados pelo aluno.

O módulo de Entrada/Saída tem um papel funcional dentro do desenvolvimento da ferramenta de avaliação, agindo como um facilitador ao gerar relatórios que alimentam outros módulos. Portanto, detalhes de sua implementação não foram comentados, uma vez que não existe influência sobre a avaliação do MCs, que é o foco deste trabalho. Entretanto, sua implementação pode ser visualizada nos anexos.

5.6. Considerações

A idealização da ferramenta teve como premissa dividi-la em módulos de acordo com as funcionalidades e necessidades exigidas para a realização da avaliação. Assim, cada passo da avaliação tornou-se um módulo, onde os resultados obtidos servem como entrada para o próximo módulo a ser executado. Além disso, a modularização por funcionalidade facilita tanto mudanças evolutivas, onde há a adaptação de novas funcionalidades, como mudanças de ajustes e melhoria, quando é necessário realizar mudanças em regras que afetam funcionalidades já existentes. Estas mudanças podem ser realizadas sem comprometer as funcionalidades de outros módulos.

6. Experimentos Realizados e Resultados Obtidos

O capítulo 6 apresenta os resultados dos experimentos que foram realizados para validar o projeto de avaliação automática de MCs.

6.1. Introdução

Os experimentos realizados têm como finalidade validar os resultados provenientes da avaliação quantitativa feita pela ferramenta. Esta validação é baseada na porcentagem de acurácia obtida pelo escore da ferramenta em relação ao escore do docente. Estes experimentos também têm como objetivo mostrar a efetividade da ferramenta no que diz respeito à avaliação qualitativa. Esta efetividade é apresentada através do relatório-guia, sendo este um produto gerado para auxiliar ao aluno na adição de conhecimento, influenciando diretamente na avaliação quantitativa.

6.2. Experimentos Realizados

Foram realizados experimentos em que os MCs dos alunos foram submetidos à análise de N-gramas com a finalidade de identificar quais de suas métricas eram relevantes para a avaliação dos MCs. Ao final da análise de similaridade percebeu-se que dentre as métricas obtidas, as mais influentes eram bi-gramas e tri-gramas, sendo estas métricas as responsáveis pelo resultado da avaliação quantitativa.

Para a realização destes experimentos, foram coletados aproximadamente 400 MCs, provenientes de respostas existentes no ambiente LabSQL (Lino, 2007), um ambiente virtual para ensino de programação SQL e Banco de Dados, o qual a ferramenta foi integrada. O produto da avaliação qualitativa são os relatórios de orientação ao aluno baseados em proposições existentes nos modelos de referência e em proposições existentes apenas no MC desenvolvido pelo aluno.

6.3. Experimentos da Avaliação Quantitativa

O experimento da avaliação quantitativa teve como principal objetivo gerar escores que definissem o desempenho do aluno para aquele MC. Para alcançar este objetivo foi necessário definir a combinação de métricas (uni-grama, bi-grama e tri-grama) que estavam diretamente ligadas ao desempenho do MC do estudante. A decisão da melhor combinação é dividida em quatro etapas:

- Na primeira etapa, cada métrica é utilizada separadamente para realizar a previsão;
- Na segunda etapa, a métrica mais influente é selecionada: bigrama;

- Na terceira etapa, a métrica de bigramas é combinada com as outras para realizar a avaliação;
- Na quarta etapa, a combinação de maior influência na avaliação é selecionada: bigramas com trigramas.

Esta divisão de etapas por combinação de métricas deve-se à necessidade de saber qual a melhor forma de avaliar o MC de um aluno, sendo necessário descobrir qual delas influencia de maneira positiva na avaliação. Selecionada a melhor combinação de métricas, estas são aplicadas em conjunto com o KNN para calcular o escore do MC.

Este experimento considerou, também, uma variação na quantidade de MCs que compõem o modelo de referência, onde essa quantia varia entre 1 e 4 MCs. As tabelas a seguir mostram os resultados das médias dos erros, considerando bi-gramas e tri-gramas, para duas das questões avaliadas dos testes, bem como a média entre elas, com o K variando entre 1 e 4.

Tabela 6.3-1. Média dos erros da comparação nota predita versus nota do avaliador para K=1 e K=2.

	K=1				K=2			
	1MC	2MC	3MC	4MC	1MC	2MC	3MC	4MC
64	1,19	1,09	0,99	1,06	1,2	1,20	1,2	1,39
115	1,28	1,20	1,33	1,28	1,12	1,23	1,27	1,02
MEDIA	1,24	1,14	1,16	1,17	1,16	1,22	1,24	1,21

Tabela 6.3-2. Média de erros da comparação nota predita versus nota do avaliador para K=3 e K=4.

	K=3				K=4			
	1MC	2MC	3MC	4MC	1MC	2MC	3MC	4MC
64	1,04	0,95	1,01	1,12	1	1,12	0,96	1,06
115	1,24	1,2	1,23	1,14	1,2	1,26	1,27	1,16
MEDIA	1,14	1,08	1,12	1,13	1,10	1,19	1,12	1,11

Como observado nas tabelas, o melhor modelo obtido no experimento quantitativo possui 2MC formando o modelo de referência e um K=3, possuindo uma média de erros de 1,08. Uma melhor visualização dos valores dos resultados obtidos entre a pontuação do especialista e a prevista pode ser acompanhada na Figura 6.4.

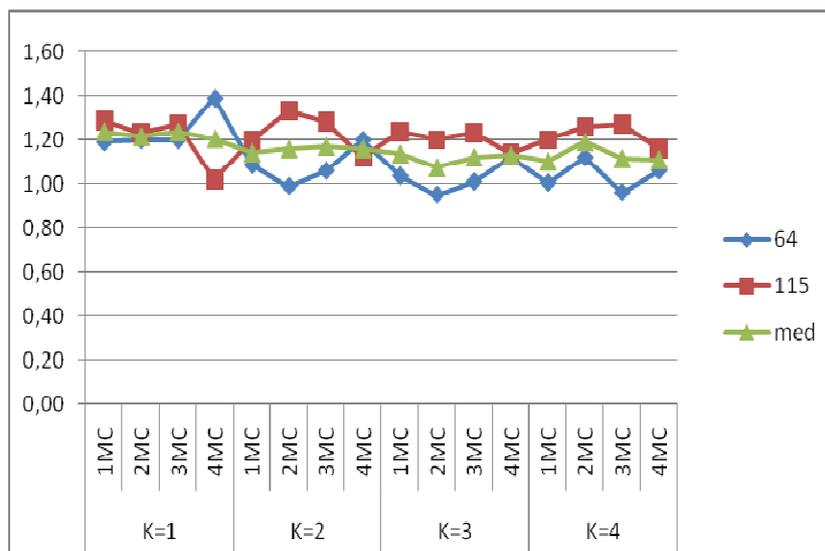


Figura 6.3-1. Média de erros para as questões 64 e 115, variando-se o número de MC (1 até 4) para compor a resposta modelo e o valor do K (1 até 4) para KNN: menor erro 1,08 com 2 MC e K=3.

A Figura 6.3-1 mostra a média de erros variando-se o número de MC, de 1 até 4, para compor a resposta modelo. A partir de 4 MCs a taxa de erros cresce. Por outro lado, os experimentos para o KNN mostraram que o melhor valor de K é 3, com a menor média de erros com 1,08 pt. A média de acertos para os experimentos realizados foi de 90% em relação à avaliação realizada por especialistas. Além disso, as métricas de bi-gramas e tri-gramas foram consideradas como as mais influentes na análise de similaridade, uma vez que foi observado que são as métricas determinantes para a previsão de um escore na análise do KNN, onde o K variou entre 1 e 4.

A análise sobre a acurácia da ferramenta em relação à predição das notas pode ser observada nas figuras Figura 6.3-2 e Figura 6.3-3, onde se verifica as diferenças entre as notas atribuídas por um especialista e as notas preditas na ferramenta pela técnica KNN para uma amostra de respostas das questões 64 e 115 analisadas.

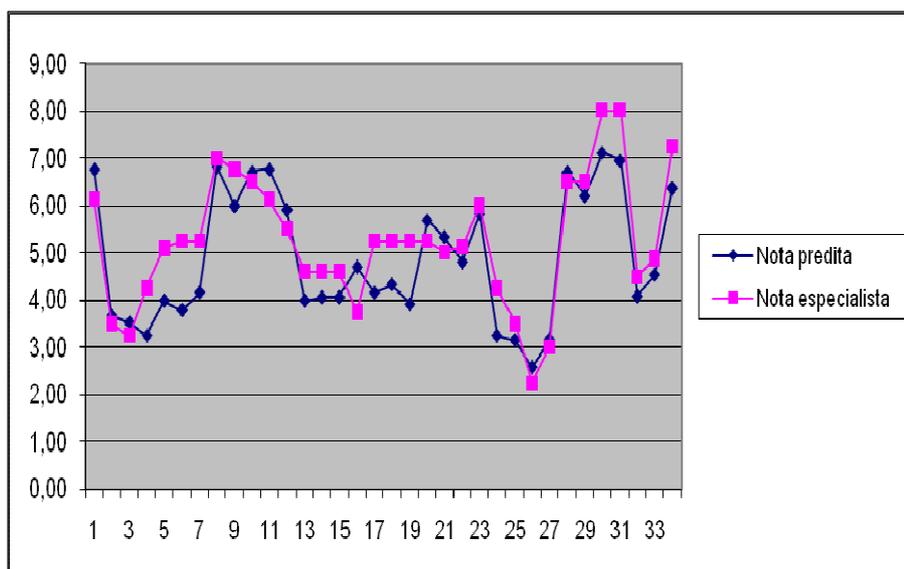


Figura 6.3-2. Notas do especialista *versus* notas preditas pelo sistema para a questão 64, utilizando 2MC modelo e classificador KNN com K=3.

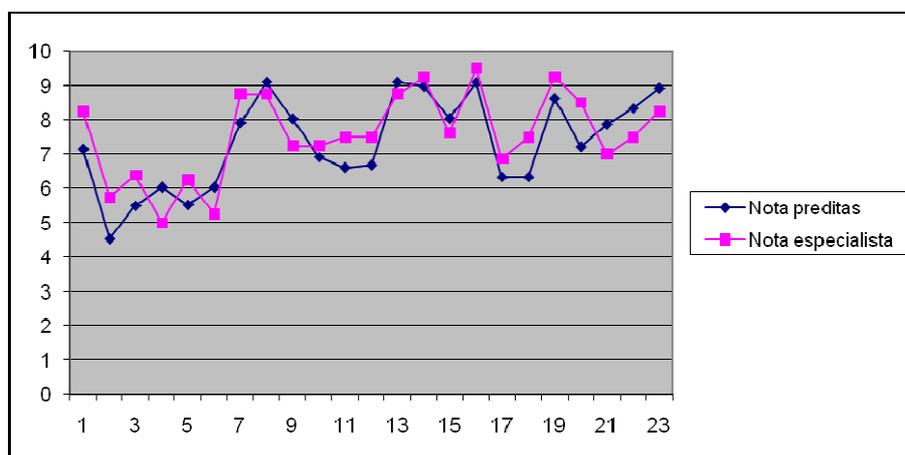


Figura 6.3-3. Notas do especialista comparadas com as notas preditas pelo sistema para a questão 115, utilizando 2 MC modelo e classificador KNN com K=3.

Os resultados alcançados através da avaliação automática alcançam uma acurácia considerável, em média 90% de acerto em relação à nota atribuída por um docente. Esta acurácia valida a ferramenta, mostrando que é compatível com as demais ferramenta de avaliação citadas no capítulo 3. Além disso, a ferramenta utiliza seu próprio score como nota de avaliação do aluno, liberando o professor que fica centrado na evolução do desempenho e do conhecimento do estudante.

O relatório-guia não é necessariamente um resultado, entretanto possui grande influencia sobre o score do aluno, sendo considerado um produto gerado pelo sistema. Este relatório-guia tem como propósito participar do processo de aquisição e

sedimentação do conhecimento do aluno. Assim, as informações são disponibilizadas para os alunos a fim de testar a capacidade de inferência do estudante. A inferência permite que o conhecimento seja guardado de maneira mais efetiva na estrutura cognitiva do indivíduo.

7. Considerações Finais

O capítulo 7 conclui todo o trabalho realizado, falando dos resultados alcançados, ponderando melhorias e propondo trabalhos futuros.

7.1. Conclusões

O tema de avaliação automática remete à dificuldade existente na tarefa de avaliação, principalmente de questões abertas, devido à sobrecarga que ela causa sobre o docente. Esta sobrecarga não permite que o docente acompanhe de maneira eficaz o aprendizado do aluno. Como solução para este problema, foi apontada a avaliação automática, que tem como objetivo abordar soluções para diminuir esta sobrecarga.

A avaliação automática tem como objetivo o auxílio ao docente no processo de verificação do conhecimento adquirido pelo estudante. Entretanto, o tipo de avaliação utilizado atualmente é mais voltado para questões objetivas. Além disso, este tipo de avaliação tem como efeito um aprendizado voltado à memorização, que rapidamente é esquecido pelo aluno, e não à retenção do conhecimento, que se sedimenta melhor na estrutura cognitiva do indivíduo.

A utilização de MCs é interessante não somente como auxílio ao docente, mas como uma metodologia alternativa que visa à melhoria no nível de aprendizado do estudante, pois sua utilização viabiliza uma retenção maior do conhecimento, resultando em um maior desenvolvimento do pensamento criativo. Por outro lado, a utilização de MCs possibilita ao docente um melhor acompanhamento sobre o desenvolvimento do estudante, evidenciando tanto os conceitos melhor apreendidos pelo aluno como suas principais dificuldades.

A abordagem utilizada neste trabalho tem como objetivo auxiliar tanto ao professor como ao aluno. O professor possui maior liberdade para acompanhar o processo de aprendizagem do aluno, uma vez que o sistema realiza a atribuição de notas. O aluno tem a possibilidade de expandir seu conhecimento gradualmente, podendo melhorar seu aprendizado através do *feedback* disponibilizado pelo sistema. Assim, acredita-se que com este trabalho estamos caminhando no sentido de automatizar a avaliação de questões não objetivas integradas a ambientes de EaD.

7.2. Resultados Alcançados

A ferramenta idealizada e implementada tem como objetivo realizar uma avaliação quantitativa, visando o auxílio ao docente. Os experimentos realizados para a avaliação quantitativa alcançaram uma média de 90% de acertos em relação a avaliadores humanos. O relatório de diagnóstico tem como finalidade o refinamento do conhecimento do aluno, visando realizar uma avaliação qualitativa. Este refinamento ocorre durante o processo de avaliação, cujo objetivo final é o direcionamento à melhor resposta. Entretanto, a avaliação qualitativa não foi validada, sendo o relatório-guia, gerado para este propósito, um produto incipiente que necessita de experimentos a fim de comprovar sua eficiência.

7.3. Trabalhos Futuros

As questões trabalhadas na ferramenta proposta são classificadas como fracamente dirigidas (*low-directedness*) [Ruiz-Primo 2004], onde o aluno constrói todo um MC apenas com o conhecimento adquirido durante o curso. Uma proposta de trabalho futuro seria a extensão da ferramenta desenvolvida a fim de realizar avaliação sobre questões altamente dirigidas (*high-directedness*). Estas questões são caracterizadas por tarefas como preenchimento de conceitos dentro de um dado MC ou complementação de um MC semi-pronto com proposições.

Um ponto de melhoria à ferramenta desenvolvida seria a validação do módulo de Avaliação Qualitativa, oferecendo aos alunos os relatórios e mensurando o grau de aumento do conhecimento que é obtido. Além disso, a validação do módulo deveria ser feita também em conjunto com a opinião dos alunos, uma vez que, como usuários, eles apontarão suas necessidades e informarão se a ferramenta é realmente eficiente no auxílio ao qual se propõe.

Outra proposta de trabalho futuro é a integração de uma ferramenta gráfica de desenvolvimento de MCs a esta ferramenta de avaliação. A união destas ferramentas facilitaria o processo de avaliação tanto para professor como para o aluno, uma vez que professor e aluno não precisariam de mais de uma ferramenta para executar suas tarefas. Além disso, o acoplamento destas ferramentas geraria um ambiente de avaliação de MCs independente e consistente, contudo passível de adaptação a um ambiente de

EAD. Assim, em um projeto maior está sendo investigada a integração desta ferramenta com o ambiente Moodle.

7.4. Publicações Relacionadas com a Dissertação

- 7.4.1. CALDAS, V. M.; FAVERO, E. L. Uma Proposta de Avaliação Automática de Mapas Conceituais para Ambientes de Ensino a Distância. In: XXXV Conferência Latino Americana de Informática, Pelotas. Anais da Conferência Latino Americana de Informática, 2009;
- 7.4.2. CALDAS, V. M.; FAVERO, E. L. Uma Ferramenta de Avaliação Automática para Mapas Conceituais como Auxílio ao Ensino em Ambientes de Educação a Distância. In: Simpósio Brasileiro de Informática na Educação, 2009, Florianópolis. Anais do Simpósio Brasileiro de Informática na Educação, Anais do SBIE, 2009. v. 20.

8. Referências

- Anohina, A., Grudspenkis, J. (2007). "A Concept Map Based Intelligent System for Adaptive Knowledge Assessment". Proceeding of the 2007 conference on Databases and Information Systems IV: Selected Papers from the Seventh International Baltic Conference.
- Anohina A., Graudina V., Grundspenkis J. (2006). "Using Concept Maps in Adaptive Knowledge Assessment". Proceedings of the 15th International Conference on Information Systems Development "Methods and Tools, Theory and Practice". Budapest, Hungary.
- Anohina A, Grundspenkis J (2006) "Prototype of multiagent knowledge assessment system for support of process oriented learning". Proceedings of the 7th Int. Baltic Conf., on DB&IS.
- Ausubel, D.P. (2000) "The acquisition and retention of knowledge: A cognitive view". Dordrecht: Kluwer Academic Publishers, p. 212.
- Ausubel, D. (1980). "Psicologia Educacional". 2. Ed. – Rio de Janeiro: Interamericana.
- Brasil. Ministério da Educação. Secretaria de Educação Profissional e Tecnológica. Decreto nº 5.622. Brasília, 19 de dez./2005. Disponível em: <http://portal.mec.gov.br/setec/arquivos/pdf_legislacao/tecnico/legisla_tecnico_dec_5622.pdf>. Acesso em: 25 nov. 2010.
- Buchweitz, B. (1984). "O uso de mapas conceituais na análise do currículo". Educação e Seleção, v. 3, n.10.
- Cañas, A. J., Leake, D. B., & Wilson D. C. (1999). "Managing, Mapping and Manipulating Conceptual Knowledge". AAAI Workshop Technical Report WS-99-10: Exploring the Synergies of Knowledge Management & Case-Based Reasoning, AAAI Press, Menlo Calif. July.
- Goldsmith, T., Johnson, P., Acton, W. (1991). "Assessing structural knowledge". Journal of Educational Psychology, vol. 83, pp. 88-96.

- Gouli, E., Gogoulou, A., Papanikolaou, K., Grigoriadou, M. (2005). "Evaluating Learner's Knowledge Level on Concept Mapping Tasks". Proceedings of the 5th IEEE International Conference on Advanced Learning Technologies. Kaohsiung, Taiwan.
- Gouli, E., Gogoulou, A., Papanikolaou, K., Grigoriadou, M. (2004). "Compass: An Adaptive Web-Based Concept Map Assessment Tool". Proceedings of the First International Conference on Concept Mapping. Pamplona, Spain.
- Gouli, E., Gogoulou, A., Grigoriadou, M. (2003). "A Coherent and Integrated Framework Using Concept Maps for Various Educational Assessment Functions". University of Athens, Greece.
- Hearst M.A. (2000). The Debate on Automated Essay Grading. IEEE Intelligent Systems, vol. 15, no. 5, pp. 22-37, September/October.
- Kondrak, G. (2005). "N-Gram Similarity and Distance". Lectures Notes in Computer Science, vol. 3772, pp 115-126.
- Lin, C., Hovy E. (2003). "Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics". Proceedings of HLT-NAACL p. 71-78. Edmonton, Canadá.
- Lino, A. D. P., Silva, A. S., Santos, T.L.T., Harb, M.P.A.H., Favero, E.L., Brito, S.R. (2007) "Avaliação automática de consultas SQL em ambiente virtual de ensinoaprendizagem". In Conferencia Ibérica de Sistemas y Tecnologías de la Información. CISTI.
- McClure, J.R., Sonak, B., Suen, H.K. (1999). "Concept Map Assessment of Classroom Learning: Reliability, Validity and Logistical Pratically". Journal of Research in Science Teaching", vol. 36, no. 4, pp.-475-492.
- Moodle (2008). Disponível Em: <[Http://Moodle.Org/](http://Moodle.Org/)>. Acesso Em: 18 Abr. 2008
- Moreira, M.; Buchweitz, B. (1987). "Mapas Conceituais – Instrumentos Didáticos e Análise de Currículo". São Paulo: Moraes.
- Novak, J.D. Cañas, A.J. (2006). "The theory underlying concept maps and how to construct them". Technical Report IHMC Cmap Tools 2006-1, Florida Institute for

Human and Machine Cognition. Available online at:
<http://cmap.ihmc.us/Publications/ResearchPapers/TheoryUnderlyingConceptMaps.pdf>

Novak J.D. (1998) “Learning, creating and using knowledge Concept Maps as Facilitative Tools in Schools and Corporations”. 2^a ed. Publicado por Lawrence Erlbaum Associates.

Novak, J. D. and Gowin, D. B. (1984). “Learning how to learn”. New York and Cambridge, UK: Cambridge University Press.

Novak J.D. (1998) “Learning, creating and using knowledge Concept Maps as Facilitative Tools in Schools and Corporations”. 2^a ed. Publicado por Lawrence Erlbaum Associates.

Rocha, F. E. L. (2007) “Avaliação da Aprendizagem: Uma Abordagem Qualitativa Baseada em Mapas Conceituais, Ontologias e Algoritmos Genéticos”. Tese de Doutorado, Centro Tecnológico, Universidade Federal do Pará, Brasil.

Rocha, F. E. L., Costa Jr, J. V., Favero, E. L. (2007). Qualitative Learning Assessment via Concept Maps. In: Revista de Informática Teórica e Aplicada, 14(1).

Ruiz-Primo, M.A. (2004) “Examining concept maps as an assessment tool”. In A. J. Cañas, J. D. Novak, & F. M. González. (Eds), Concept Maps: Theory, Methodology, Technology. Proceedings of the First Conference on Concept Mapping p. 555-562. Pamplona: Universidad Pública de Navarra.

Ruiz-Primo, M. A. Shultz, S. E., Li, M., Shavelson, R. J. (2001). Comparison of the reliability and validity of scores from two concept-mapping techniques. Journal of Research in Science Teaching, 38(2), 260-278.

Ruiz-Primo, M.A., Schultz, E. S., & Shavelson, R.J. (1996) “Concept map-based assessments in science: An exploratory study”. In American Educational Research Association, New York, NY.

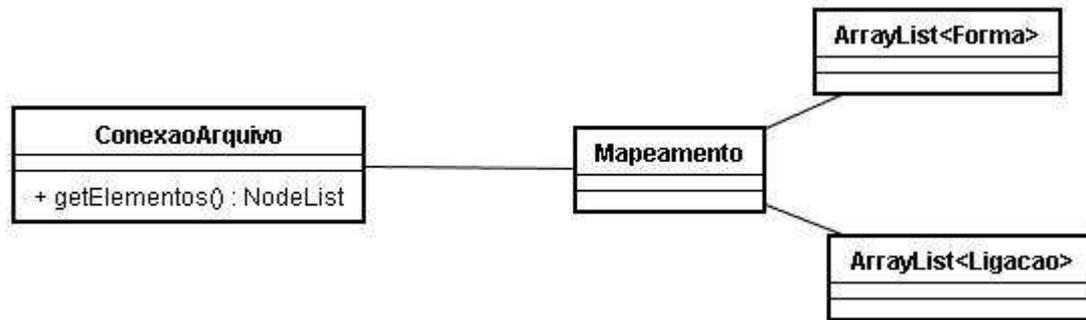
Schaal, S. (2008) “Concept Mapping In Science Education Assessment: An Approach To Computer-Supported Achievement Tests In An Interdisciplinary Hypermedia

Learning Environment”. Proceedings of the Third International Conference on Concept Mapping. Tallinn, Estonia & Helsinki, Finland.

Teleduc (2008). Disponível em: <<http://teleduc.nied.unicamp.br/teleduc>>. Acesso em: 23 mai. 2008

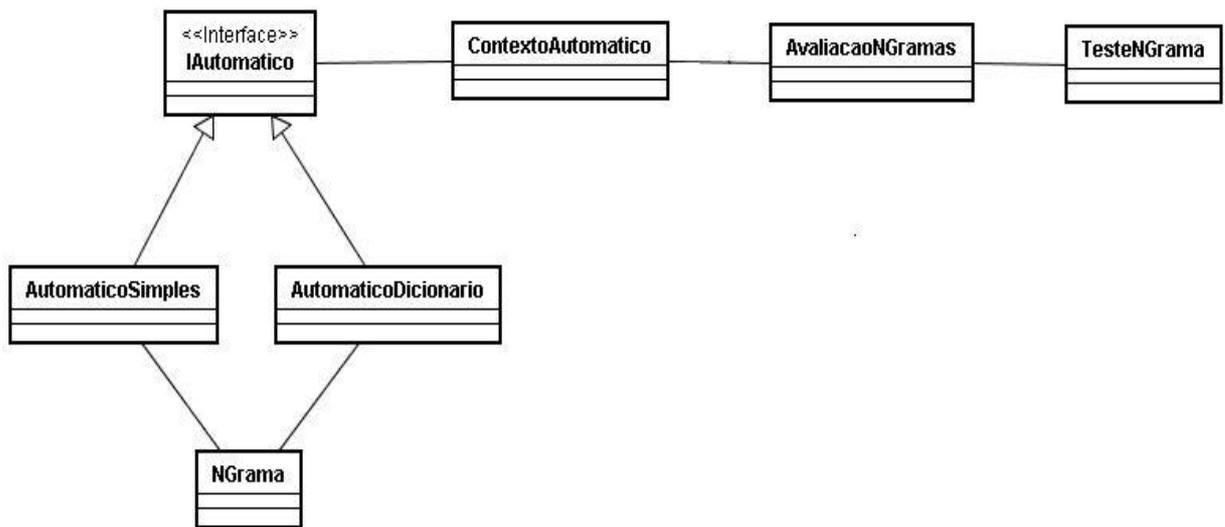
Yin et.al. (2005). “Comparison of Two Concept-Mapping Techniques: Implications for Scoring, Interpretation, and Use”. Journal Of Research In Science Teaching, vol. 42, no. 2, pp. 166-184.

Anexo A – Modelagem da Extração de Informação



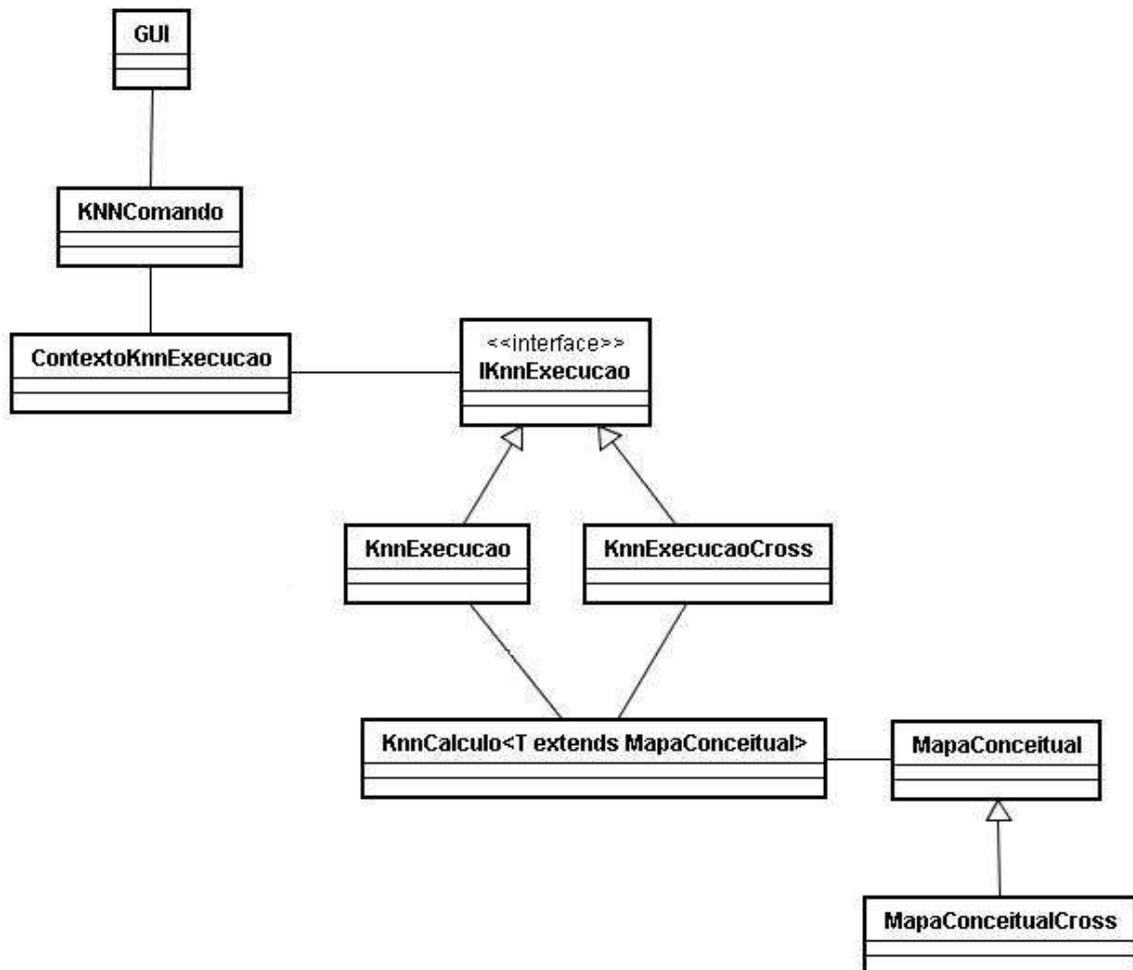
Módulo de Extração de Informação.

Anexo B – Modelagem N-Gramas



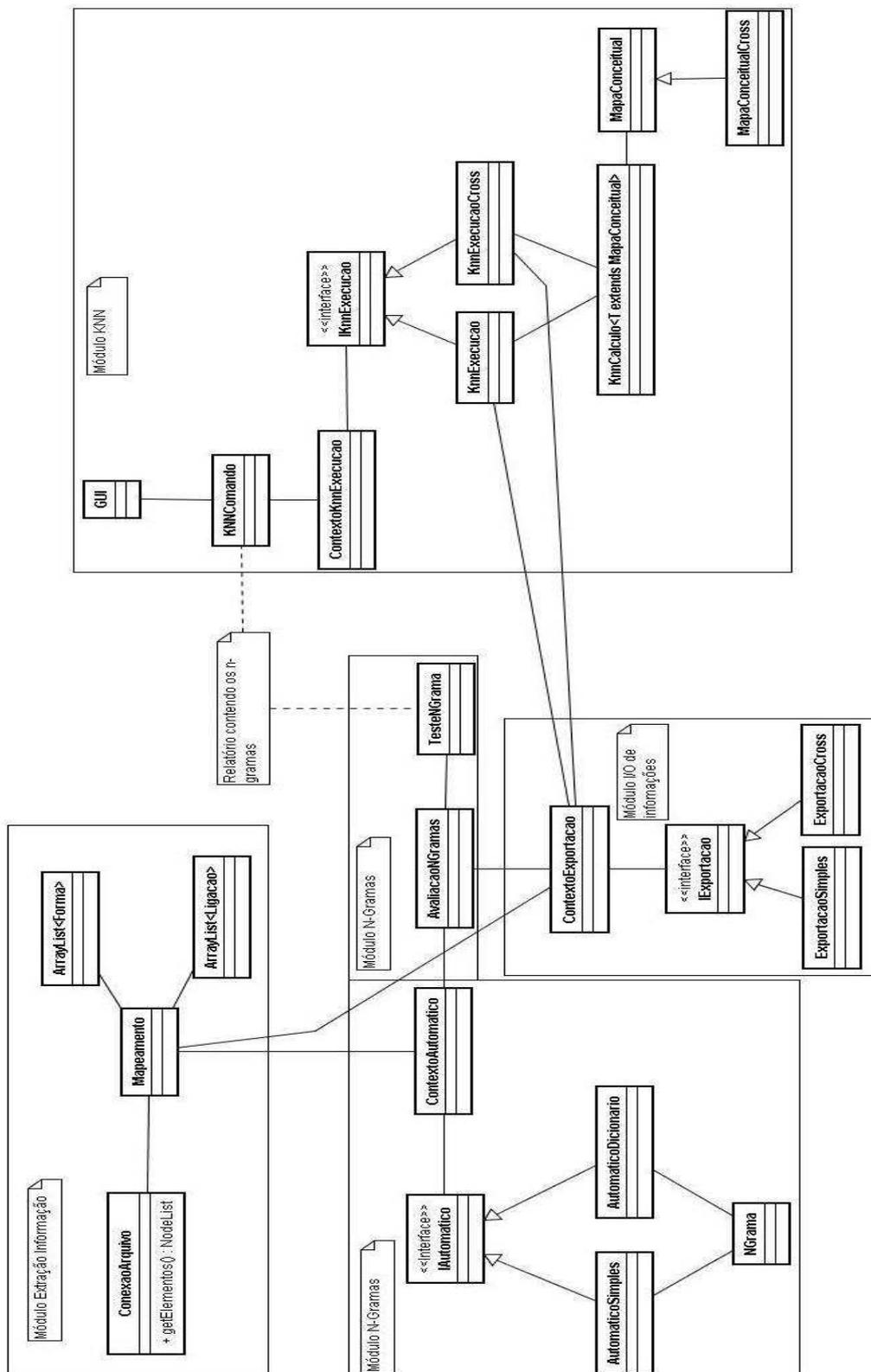
Composição do Módulo de N-Gramas.

Anexo C – Modelagem do KNN



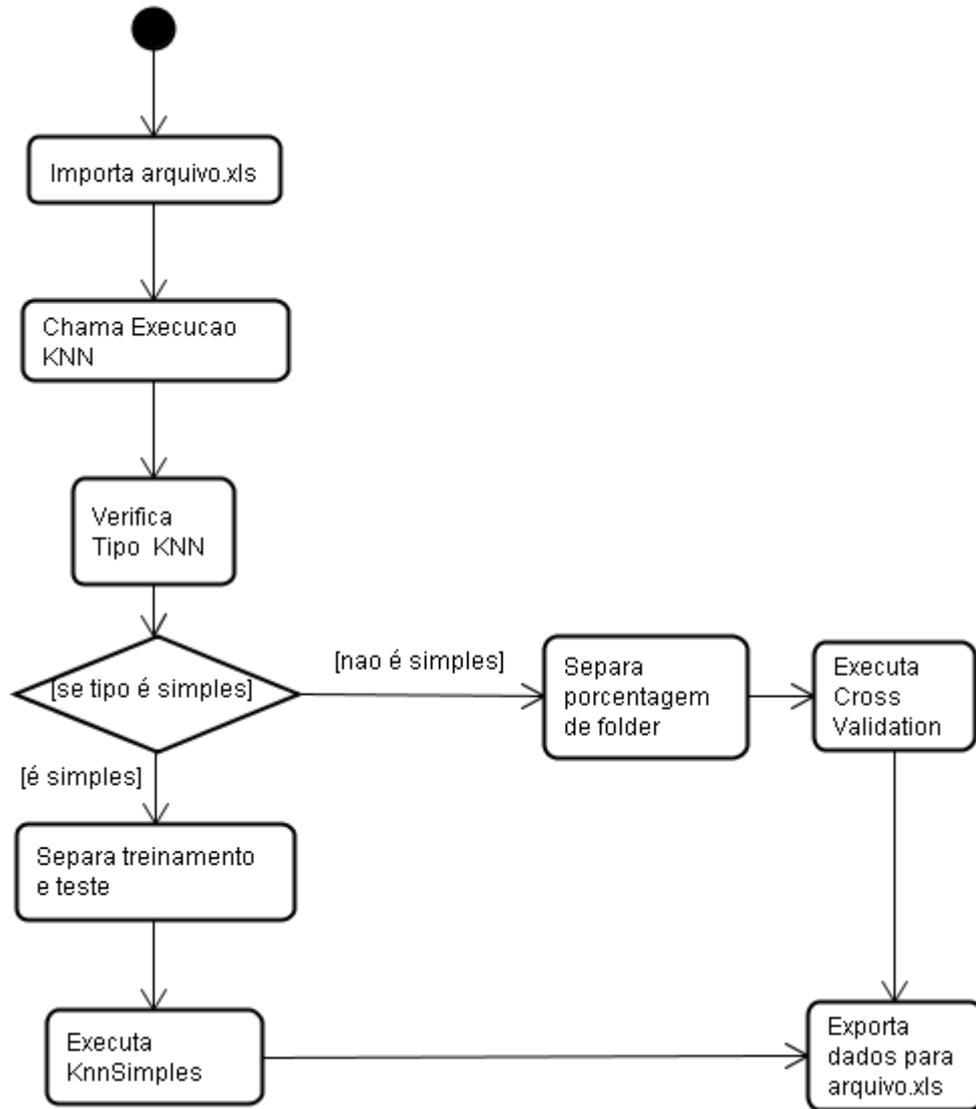
Estrutura do módulo KNN, abrangendo seus modelos de predição (Simple e Cross-Validation)

Anexo D – Modelagem do Sistema



Modelagem completa e interação entre os módulos existentes do sistema.

Anexo E – Fluxo KNN



Fluxo dos dados durante o processamento do módulo KNN.

Anexo F – Métodos

compararLigacoesString()

```

private StringBuilder compararLigacoesString(NodeList nodeList) {
    conceitos = new ArrayList<Forma>();
    ligacoes = new ArrayList<Ligacao>();
    palavrasLigacao = new ArrayList<Forma>();

    StringBuilder sb = new StringBuilder();
    listarElementos(nodeList);
    for (Ligacao ligacao : ligacoes) {

        String origem = ligacao.getOrigem();

        if(origem.equalsIgnoreCase(listarConceitos(origem).getId()
        ) {
            continue;
        }
        if(origem.equalsIgnoreCase(listarPalavras(origem).getId()))
        {
            if
            (origem.equalsIgnoreCase(listarOrigemLigacoes(origem)
            .getOrigem())) {

                sb.append(listarConceitos(
                listarDestinoLigacoes(origem).getOrigem()).getPalavra()+ " "+
                listarPalavras(ligacao.getOrigem()).getPalavra()+ " "+
                listarConceitos(ligacao.getDestino()).getPalavra() + " ");
            }
        }
    }
    return sb;
}

```

Anexo G – Métodos

compararLigacoesArray()

```
private ArrayList<String> compararLigacoesArray(NodeList nodeList) {
    conceitos = new ArrayList<Forma>();
    ligacoes = new ArrayList<Ligacao>();
    palavrasLigacao = new ArrayList<Forma>();
    ArrayList<String> resultado = new ArrayList<String>();
    listarElementos(nodeList);

    for (Ligacao ligacao : ligacoes) {

        String origem = ligacao.getOrigem();

        if
        (origem.equalsIgnoreCase(listarConceitos(origem).getId()))
        {
            continue;
        }
        if
        (origem.equalsIgnoreCase(listarPalavras(origem).getId())) {
            if
            (origem.equalsIgnoreCase(listarOrigemLigacoes(origem)
                .getOrigem())) {
                resultado.add(listarConceitos(

                    listarDestinoLigacoes(origem).getOrigem()
                        .getPalavra());
                resultado.add(listarPalavras(ligacao.getOrigem()).
                    getPalavra());

                resultado.add(listarConceitos(ligacao.getDestino())
                    .getPalavra());
            }
        }
        }
    }
    return resultado;
}
```

Anexo H – Métodos

listarConceitos() e de listagem de ligacoes

```
private Forma listarConceitos(String id) {
    Forma forma = new Forma();
    for (Forma conceito : conceitos) {
        if (conceito.getId().equalsIgnoreCase(id)) {
            forma = conceito;
            break;
        }
    }
    return forma;
}

private Ligacao listarOrigemLigacoes(String id) {
    Ligacao nova = new Ligacao();
    for (Ligacao ligacao : ligacoes) {
        if (ligacao.getOrigem().equalsIgnoreCase(id)) {
            nova = ligacao;
            break;
        }
    }
    return nova;
}

private Ligacao listarDestinoLigacoes(String id) {
    Ligacao nova = new Ligacao();
    for (Ligacao ligacao : ligacoes) {
        if (ligacao.getDestino().equalsIgnoreCase(id)) {
            nova = ligacao;
            break;
        }
    }
    return nova;
}
```

Anexo I – Método

compararLigacoesRelatorio()

```

private ArrayList<String> compararLigacoesRelatorio(NodeList nodeList)
{
    conceitos = new ArrayList<Forma>();
    ligacoes = new ArrayList<Ligacao>();
    palavrasLigacao = new ArrayList<Forma>();
    ArrayList<String> resultado = new ArrayList<String>();
    listarElementos(nodeList);

    for (Ligacao ligacao : ligacoes) {

        String origem = ligacao.getOrigem();

        if (origem.equalsIgnoreCase(listarConceitos(origem).
            getId())) {
            continue;
        }
        if (origem.equalsIgnoreCase(listarPalavras(origem).
            getId())) {
            if (origem.equalsIgnoreCase(
                listarOrigemLigacoes(origem)
                    .getOrigem())) {
                resultado.add(listarConceitos(
                    listarDestinoLigacoes(origem).getOrigem())
                        .getPalavra());

                resultado.add(listarPalavras(ligacao.getO
                    rigem()).getPalavra());
                resultado.add(listarConceitos(ligacao.get
                    Destino()).getPalavra());
                conceitosRelatorio.add(listarConceitos(
                    listarDestinoLigacoes(origem).getOrigem())
                        .getPalavra());
                conceitosRelatorio.add(listarConceitos(ligacao.getDestino())
                    .getPalavra());
            }
        }
    }
    return resultado;
}

```

Anexo J – Classe KnnExecucao

```

/**
 * Classe que implementa a execução do KNN da forma mais simples,
 * separando uma porcentagem dos dados para treinamento e outra para
 * teste
 * @author vanessa.caldas
 * @version 1.2
 *
 */
public class KnnExecucao implements IKnnExecucao {

    private double porcentagemTeste;
    private KnnCalculo<MapaConceitual> calculo;
    private List<MapaConceitual> mapasTeste;
    private List<MapaConceitual> mapasTreinamento;
    List<MapaConceitual> listaTemporaria;

    public KnnExecucao(double porcentagemTeste) {
        // TODO Auto-generated constructor stub
        if(porcentagemTeste>0) {
            this.porcentagemTeste = porcentagemTeste;
        }
        else this.porcentagemTeste = 0.5;

        this.calculo = new KnnCalculo<MapaConceitual>();
        this.mapasTeste = new ArrayList<MapaConceitual>();
        this.mapasTreinamento = new ArrayList<MapaConceitual>();
        listaTemporaria = new ArrayList<MapaConceitual>();
    }

    @Override
    public void executarAvaliacao(int k, int numeroVariaveis,
        double porcentagem, int folders, String nomeArquivo)
    {
        ContextoExportacao<MapaConceitual> exportacao = new
ContextoExportacao<MapaConceitual>(new ExportacaoSimples());
        listaTemporaria = exportacao.importarExcel(nomeArquivo);
        separarLista();
        // calcularKNN(listaTreinamento, listaTeste, kVizinhos,
numeroVariaveis)
        calculo.calcularKNN(mapasTreinamento, mapasTeste, k,
numeroVariaveis);
        exportacao.exportarExcel(nomeArquivo+"resultado",
mapasTeste);
    }

    @Override
    public void separarLista() {
        // TODO Auto-generated method stub
        for(MapaConceitual mc: listaTemporaria) {
            for (int i = 0; i < Math.round(listaTemporaria.size()
* porcentagemTeste); i++) {
                mapasTeste.add(mc);
            }
            mapasTreinamento.add(mc);
        }
    }
}

```

Anexo K – Classe

KnnExecucaoCross

```

/**
 * Classe que representa a parte superficial da execução do KNN
 * utilizando Cross Validation. Utiliza a classe KnnCalculo para
 * realizar o cálculo do KNN. Esta classe trata o modo como os Mapas
 * Conceituais devem se comportar para realizar o Cross Validation.
 *
 * @author vanessa.caldas
 * @version 1.1
 *
 */
public class KnnExecucaoCross implements IKnnExecucao {

    private int folders;
    private List<MapaConceitualCross> mapasTeste;
    private List<MapaConceitualCross> mapasTreinamento;
    private List<MapaConceitualCross> listaTemporaria;
    private KnnCalculo<MapaConceitualCross> calculo;

    public KnnExecucaoCross(int numeroFolders) {
        if(numeroFolders!=0) {
            this.folders = numeroFolders;
        }
        else this.folders = 10;
        this.calculo = new KnnCalculo<MapaConceitualCross>();
        mapasTeste = new ArrayList<MapaConceitualCross>();
        mapasTreinamento = new ArrayList<MapaConceitualCross>();
        listaTemporaria = new ArrayList<MapaConceitualCross>();
    }

    @Override
    public void executarAvaliacao(int k, int numeroVariaveis,
        double porcentagem, int folders, String nomeArquivo)
    {
        ContextoExportacao<MapaConceitualCross> contextoExportacao =
        new ContextoExportacao<MapaConceitualCross>(
            new ExportacaoCross());
        listaTemporaria =
        contextoExportacao.importarExcel(nomeArquivo);
        realizarRodizio(k, numeroVariaveis);
        contextoExportacao.exportarExcel(nomeArquivo,
        listaTemporaria); }

    @Override
    public void separarLista() {
        int numeroElementos = Math.round(listaTemporaria.size() /
        folders);
        for (int i = 0; i < numeroElementos; i++) {
            if ((listaTemporaria.get(i).isAnalisado() == true &&
        listaTemporaria.get(i).getNotaPreditada() != 0)
            || (listaTemporaria.get(i).isAnalisado() == false &&
        listaTemporaria.get(i).getNotaPreditada() == 0)) {
                listaTemporaria.get(i).setAnalisado(true);
                mapasTeste.add(listaTemporaria.get(i));
            }
        }
    }
}

```

```
        listaTemporaria.remove(i);
        }}
        mapasTreinamento.addAll(listaTemporaria);
    }

    private void resetarListas() {
        listaTemporaria.addAll(mapasTeste);
        mapasTeste.clear();
        mapasTreinamento.clear();
    }

    private void realizarRodizio(int k, int numeroVariaveis) {
        for (int i = 0; i < folders; i++) {
            separarLista();
            calculo.calcularKNN(mapasTreinamento, mapasTeste, k,
numeroVariaveis);
            resetarListas();
        }
    }
}
```

Anexo L – Método gerarModelo()

```
/**
 * Método gerarModelo
 */
public List<T> gerarModelo(List<T> treinamento, T teste, int k,
                          int variaveis) {
    List<T> modelo = new ArrayList<T>();
    switch (variaveis) {
        case 0:
            analisarUnigrama(treinamento, teste);
            break;
        case 1:
            analisarBigrama(treinamento, teste);
            break;
        case 2:
            analisarTrigrama(treinamento, teste);
            break;
        case 3:
            analisarUniBigrama(treinamento, teste);
            break;
        case 4:
            analisarUniTrigrama(treinamento, teste);
            break;
        case 5:
            analisarBiTrigrama(treinamento, teste);
            break;
        case 6:
            analisarUniBiTrigrama(treinamento, teste);
            break;
        default:
            break;
    }
    modelo = recuperarModelos(treinamento, k);
    return modelo;
}
```

Método gerarModelo()

Anexo M – Métodos recuperarModelo() e calcularMedia()

```
/**
 * Método que retorna os K modelos de resposta mais próximos
 */
private List<T> recuperarModelos(List<T> treinamento, int k) {
    ordenarLista(treinamento);
    List<T> modelos = new ArrayList<T>();
    for (int i = 0; i < k; i++) {
        modelos.add(treinamento.get(i));
    }
    return modelos;
}
```

Método que seleciona os K objetos MapaConceitual a menor distância euclidiana.

```
/**
 * Método que calcula a nota do aluno baseada nos Mapas Modelo.
 */
public double calcularMedia(List<T> modelos, int k) {
    double notas = 0;
    for (T modelo : modelos) {
        notas += modelo.getNotaAvaliador();
    }
    return notas /= k;
}
```

Método que seleciona os K objetos MapaConceitual a menor distância euclidiana.