
**Reúso de conteúdo da Web na Recomendação
Personalizada de Objetos de Aprendizagem:
uma abordagem baseada em um Algoritmo
Genético, Tecnologias da Web Semântica e uma
Ontologia**

Clarivando Francisco Belizário Júnior



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Clarivando Francisco Belizário Júnior

**Reúso de conteúdo da Web na Recomendação
Personalizada de Objetos de Aprendizagem:
uma abordagem baseada em um Algoritmo
Genético, Tecnologias da Web Semântica e uma
Ontologia**

Dissertação de mestrado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Fabiano Azevedo Dorça

Uberlândia
2018

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

B431r
2018 Belizário Júnior, Clarivando Francisco, 1987-
Reúso de conteúdo da web na recomendação personalizada de
objetos de aprendizagem [recurso eletrônico] : uma abordagem baseada
em um algoritmo genético, tecnologias da web semântica e uma
ontologia / Clarivando Francisco Belizário Júnior. - 2018.

Orientador: Fabiano Azevedo Dorça.

Dissertação (mestrado) - Universidade Federal de Uberlândia,
Programa de Pós-Graduação em Ciência da Computação.

Modo de acesso: Internet.

Disponível em: <http://dx.doi.org/10.14393/ufu.di.2018.988>

Inclui bibliografia.

Inclui ilustrações.

1. Computação. 2. Web semântica. 3. Objetos de Aprendizagem. I.
Dorça, Fabiano Azevedo, 1979- (Orient.) II. Universidade Federal de
Uberlândia. Programa de Pós-Graduação em Ciência da Computação.
III. Título.

CDU: 681.3

Maria Salete de Freitas Pinheiro - CRB-6/1262

UNIVERSIDADE FEDERAL DE UBERLÂNDIA – UFU
FACULDADE DE COMPUTAÇÃO – FACOM
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO – PPGCO

Os abaixo assinados, por meio deste, certificam que leram e recomendam para a Faculdade de Computação a aceitação da dissertação intitulada **“Reúso de conteúdo da Web na Recomendação Personalizada de Objetos de Aprendizagem: uma abordagem baseada em um Algoritmo Genético, Tecnologias da Web Semântica e uma Ontologia”** apresentada por **“Clarivando Francisco Belizário Júnior”** como parte dos requisitos exigidos para a obtenção do título de **Mestre em Ciência da Computação**.

Uberlândia, 24 de novembro de 2018

Orientador: _____
Prof. Dr. Fabiano Azevedo Dorça
Universidade Federal de Uberlândia

Banca Examinadora:

Prof. Dr. Renan Gonçalves Cattelan
Universidade Federal de Uberlândia

Prof. Dr. Rafael Dias Araújo
Universidade Federal de Uberlândia

Prof. Dr. Alessandro Vivas Andrade
Universidade Federal dos Vales do Jequitinhonha e Mucuri

Agradecimentos

Agradeço à minha família abençoada pela força invisível que nos alicerça.

Agradeço ao meu orientador Fabiano pelo conhecimento compartilhado, contribuindo para o meu progresso intelectual e moral.

Agradeço aos membros da banca pela contribuição na melhoria deste trabalho.

Agradeço ao CNPq pela bolsa concedida para a realização desta pesquisa.

Por fim, agradeço a todas as pessoas que com suas qualidades me inspiraram a me tornar uma pessoa melhor.

Resumo

O *e-learning* é um modelo de ensino eletrônico que pode ser adaptado aos estilos de aprendizagem de cada estudante. Nesse contexto, muitos dos recursos educacionais disponíveis não estão devidamente estruturados em algum padrão que facilite o seu reuso. Além disso, os objetos de aprendizagem (OAs) possuem certas peculiaridades e metadados (informações que os descrevem) que tornam a sua criação uma tarefa dispendiosa em termos de tempo e custo em dinheiro. Para solucionar esse problema, neste trabalho, desenvolve-se uma abordagem que utiliza conteúdo da Wikipédia para a criação de novos OAs e cria-se uma ontologia para a modelagem de OAs e de estudantes. O Sistema de Criação e Recomendação de Objetos de Aprendizagem (SCROA) que implementa essa abordagem possui dois tipos de recomendação, ambas assistidas por regras de inferência usadas para sugerir todos os OAs da ontologia que possuem alguma similaridade com os parâmetros de busca do usuário. Na primeira situação, recomenda-se os OAs que melhor atendem a esses parâmetros. No segundo tipo de recomendação, o usuário define também alguns conceitos que se espera que o aluno aprenda, então tem-se um Problema de Recomendação de Objetos de Aprendizagem baseado em conceitos, cujo objetivo é a recomendação de OAs que cobrem todos os conceitos e ao mesmo tempo atendam ao perfil do estudante e aos demais parâmetros definidos pelo usuário. Esse problema é resolvido por um algoritmo genético (AG). Quando os OAs sugeridos pela ontologia não são suficientes para cobrir todos os conceitos, antes da execução do AG, cria-se novos OAs pelo reuso de conteúdo wiki levando em conta sua qualidade. O AG garante que os OAs recomendados cobrem todos os conceitos e atendem ao perfil do estudante. Dada a eficiência da abordagem do SCROA, esse pode impactar na popularização de Sistemas Hipermídia Adaptativos que permitam que alunos de todas as classes sociais tenham um melhor aproveitamento e desempenho no processo ensino-aprendizagem.

Palavras-chave: Ensino Eletrônico. Web Semântica. Objetos de Aprendizagem. Estilos de Aprendizagem.

Abstract

E-learning is an electronic teaching model that can be adapted to the learning styles of every student. In this context, many of the available educational resources are not properly structured in any pattern that facilitates their reuse. In addition, learning objects (LOs) have certain peculiarities and metadata (information that describes them) that make their creation a time-consuming and costly task. To solve this problem, this work develops an approach that uses Wikipedia content to create new LOs and creates an ontology for modeling LOs and students. The system (SCROA) that implements this approach has two types of recommendations, both of which are assisted by inference rules used to suggest all ontology LOs that have some similarity to user search parameters. In the first type of recommendation, LOs that best meet these parameters are recommended. In the second type of recommendation, the user also defines some concepts that the learner is expected to learn, so there is a concept-based Learning Object Recommendation Problem, which aims at recommending LOs that cover all concepts and at the same time meet the profile of the student and other parameters defined by the user. A genetic algorithm (GA) solves this problem. When the LOs suggested by the ontology are not sufficient to cover all the concepts, before the GA is executed, new LOs are created by the reuse of wiki content taking into account their quality. The GA ensures that the recommended LOs cover all concepts and meet the student profile. Given the efficiency of the SCROA approach, this can influence the popularization of Adaptive Hypermedia Systems that allow students of all social classes to have a better performance in the teaching and learning process.

Keywords: E-learning. Semantic Web. Learning Objects. Learning Styles.

Lista de ilustrações

Figura 1 – Etapas fundamentais de um algoritmo genético	31
Figura 2 – Representação binária do cromossomo de um indivíduo	32
Figura 3 – Crossover de um ponto entre cromossomos binários	33
Figura 4 – Crossover incremental entre cromossomos binários	34
Figura 5 – Modelagem do Sistema de Criação e Recomendação de Objetos de Aprendizagem (SCROA)	48
Figura 6 – Classes e subclasses da ontologia	52
Figura 7 – As subclasses e propriedades da classe LearningObject	53
Figura 8 – Os tipos de recursos de aprendizagem dos padrões IEEE-LOM e CLEO	54
Figura 9 – Atributos das classes Student e Profile	55
Figura 10 – Modelagem do perfil do estudante conforme o FSLSM	56
Figura 11 – Processamento de um sumário da Wikipédia para a geração de conceitos	59
Figura 12 – Representação do cromossomo de um indivíduo	67
Figura 13 – Crossover incremental de indivíduos do AG que resolve o PROA	68
Figura 14 – Comparação da densidade semântica do OA ideal em relação aos demais OAs	74
Figura 15 – Comparação do grau de dificuldade do OA ideal em relação aos demais OAs	75
Figura 16 – Comparação do parâmetro qualidade do OA ideal em relação aos de- mais OAs	75
Figura 17 – Tempo do processo que compara o OA ideal com os demais OAs	75
Figura 18 – Comparação da densidade semântica média das 5 melhores soluções em relação ao OA ideal	77
Figura 19 – Comparação do grau de dificuldade médio das 5 melhores soluções em relação ao OA ideal	78
Figura 20 – Comparação da qualidade média das 5 melhores soluções em relação ao OA ideal	78
Figura 21 – Comparação entre o AG e um algoritmo totalmente aleatório	79

Lista de tabelas

Tabela 1 – Categorias do padrão IEEE-LOM	29
Tabela 2 – Comparação da literatura correlata com a proposta deste trabalho . . .	43
Tabela 3 – Os tipos de recursos recomendados para cada perfil de estudante . . .	57
Tabela 4 – Classes de qualidade da Wikipédia	58
Tabela 5 – Parâmetros utilizados nos testes	72
Tabela 6 – Pesos atribuídos ao OA ideal nos quatro testes do grupo A	74
Tabela 7 – Os 10 melhores OAs ranqueados em cada teste	74
Tabela 8 – Pesos atribuídos ao OA ideal nos quatro testes do grupo B	76
Tabela 9 – As 5 melhores soluções em cada teste do grupo B	77
Tabela 10 – Tempo (em segundos) de execução do AG	79

Lista de siglas

AG Algoritmo Genético

ARP Algoritmo de Recomendação Principal

FSLSM Felder-Silverman Learning Style Model

HTML HyperText Markup Language

IRI Internationalized Resource Identifier

NP Noun Phrase

OA Objeto de Aprendizagem

OWL Web Ontology Language

PROA Problema da Recomendação de Objetos de Aprendizagem

PCC Problema da Cobertura de Conjuntos

PR Problema do Recobrimento

PP Problema do Particionamento

PLN Processamento de Linguagem Natural

RDF Resource Definition Framework

SCROA Sistema de Criação e Recomendação de Objetos de Aprendizagem

SWRL Semantic Web Rule Language

URI Uniform Resource Identifier

URL Uniform Resource Locator

WS Web Semântica

XML eXtensible Markup Language

Sumário

1	INTRODUÇÃO	19
1.1	Motivação	21
1.2	Objetivos e Desafios da Pesquisa	21
1.3	Hipóteses	21
1.4	Contribuições	22
1.5	Organização da Dissertação	22
2	FUNDAMENTAÇÃO TEÓRICA	23
2.1	Wikipédia	23
2.2	Avaliação da qualidade de páginas da Wikipédia	24
2.3	Web Semântica	25
2.4	O desenvolvimento de ontologias	26
2.5	Estilos de aprendizagem e o modelo de Felder e Silverman . . .	27
2.6	Padrões de metadados de objetos de aprendizagem	28
2.7	O Problema da Cobertura de Conjuntos	30
2.8	Algoritmo genético	30
2.9	Algoritmo genético para o Problema da Cobertura de Conjuntos	31
2.9.1	Representação do indivíduo	32
2.9.2	Função de avaliação	32
2.9.3	Técnica de seleção	32
2.9.4	Cruzamento	33
2.9.5	Mutação	34
2.9.6	Soluções viáveis	35
2.9.7	Visão geral do algoritmo	36
3	TRABALHOS RELACIONADOS	39
3.1	Recomendação de recursos educacionais	40
3.1.1	Recomendação de objetos de aprendizagem baseada em conceitos	41

3.2	Reúso de conteúdo da Web e a utilização de tecnologias da Web Semântica	43
4	ABORDAGEM PROPOSTA	47
4.1	O Problema da Recomendação de Objetos de Aprendizagem (PROA)	50
4.2	Ontologia	51
4.2.1	Modelagem dos objetos de aprendizagem	53
4.2.2	Modelagem dos estudantes	55
4.2.3	Regras SWRL	56
4.3	Avaliação da qualidade de seções wiki	57
4.4	Identificação dos conceitos de seções wiki	58
4.4.1	Extração de conteúdo da Wikipédia com a Wikipedia API	59
4.4.2	O Processamento de Linguagem Natural com as bibliotecas spaCy e NLTK	60
4.5	Algoritmo de Recomendação Principal (ARP)	60
4.5.1	Etapas do ARP	61
4.5.2	Algoritmo de conversão de seções wiki em objetos de aprendizagem	64
4.5.3	Algoritmo genético para a solução do PROA	66
5	EXPERIMENTOS E ANÁLISE DOS RESULTADOS	71
5.1	Métodos e Experimentos	71
5.1.1	Testes do grupo A	73
5.1.2	Testes do grupo B	76
5.2	Avaliação dos Resultados	79
6	CONCLUSÃO	83
6.1	Principais Contribuições	84
6.2	Trabalhos Futuros	84
6.3	Contribuições em Produção Bibliográfica	85
	REFERÊNCIAS	87
	APÊNDICES	91
	APÊNDICE A – REGRAS SWRL IMPLEMENTADAS NA ONTOLOGIA	93

CAPÍTULO 1

Introdução

Lidar com as especificidades de cada aluno no contexto da sala de aula ainda é um problema pendente, pois as estratégias de ensino-aprendizagem do professor acabam sendo direcionadas para a turma. Uma área interdisciplinar que se preocupa com esse problema é a Informática na Educação. Nessa área, acredita-se que os alunos aprendem mais em menos tempo quando o conteúdo didático é adaptado a seus estilos de aprendizagem.

Daí surgiu a ideia da construção dos Sistemas Hipermídia Adaptativos, que criam um modelo para cada usuário considerando seus desejos, preferências e conhecimento para adaptar vários aspectos visíveis do sistema ao usuário (BRUSILOVSKY, 1996). Assim, esses sistemas são capazes de adaptar o conteúdo didático, doravante Objeto de Aprendizagem (OA), às especificidades de cada aluno, auxiliando os professores na potencialização do desempenho dos alunos e também auxiliando estudantes autodidatas a alcançarem seus objetivos sem a intervenção de um professor.

Embora possa haver muitos recursos educacionais disponíveis, muitos deles não estão devidamente estruturados em algum padrão que permite a sua recomendação. Além disso, os OAs possuem determinadas características e metadados (informações que os descrevem) que tornam a sua criação uma tarefa dispendiosa. O IEEE-LOM (LTSC, 2002) é um padrão popular utilizado na criação de metadados que retratam os OAs. Os OAs e seus metadados são armazenados em repositórios de objetos de aprendizagem. Alguns repositórios armazenam apenas os metadados. Nesse caso, algum campo contém o endereço do recurso descrito pelos metadados. O padrão IEEE-LOM, por exemplo, possui o campo “Entrada” da categoria “Geral” que armazena um identificador para o OA, por exemplo, um Uniform Resource Identifier (URI).

Nem todos os campos do padrão IEEE-LOM são muito utilizados. Além disso, alguns campos possuem um vocabulário reduzido. Uma maneira de ampliar os vocabulários reduzidos é utilizando alguma extensão. O Customized Learning Experience Online (CLEO, 2003) estende o padrão IEEE-LOM ampliando, por exemplo, o vocabulário dos tipos de recursos de aprendizagem associados aos objetos de aprendizagem.

Sabe-se que muitos conteúdos da Web, que podem ser utilizados no processo ensino-

aprendizagem, não estão descritos com metadados de objetos de aprendizagem. Além disso, a Web possui mais recursos que os repositórios de OAs. Esses recursos da Web não podem ser ignorados porque a criação de novos OAs é uma tarefa que envolve um custo com tempo e dinheiro. Portanto, é viável pensar no reuso de conteúdos da Web para facilitar o processo de criação de OAs. Acontece que, se por um lado a Web possui uma infinidade de novos conteúdos para reuso, por outro lado, as máquinas não compreendem esses conteúdos, por falta de metadados que os descrevem.

Para lidar com esse problema, surgiu a Web Semântica, que estende a Web tradicional permitindo que as máquinas compreendam a semântica das informações da Web. Neste trabalho, apresenta-se uma abordagem para o reuso de conteúdo da Wikipédia usando tecnologias da Web Semântica para a recomendação personalizada de OAs, contribuindo para a autoria (criação) de novos recursos educacionais e, conseqüentemente, impactando na popularização de Sistemas Hipermídia Adaptativos que permitem que alunos de todas as camadas sociais tenham um melhor desempenho no processo ensino-aprendizagem.

O conteúdo da Wikipédia pode ser adequadamente avaliado quanto a características como a confiabilidade das fontes, estrutura da página, clareza do conteúdo e adequação gramatical. Esses são alguns dos elementos utilizados pela comunidade da Wikipédia na avaliação da qualidade dos artigos wiki. Alguns usuários dessa comunidade marcam os artigos com rótulos de qualidade ajudando, assim, os editores a identificarem mais facilmente as páginas wiki que precisam ser melhoradas. Neste trabalho, convertem-se as seções wiki, que cobrem os conceitos que se espera que o aluno aprenda, em OAs, levando-se em conta a qualidade das páginas wiki das seções utilizadas.

A abordagem apresentada neste trabalho define uma ontologia na qual são modelados os estudantes e os metadados de seções wiki. Assim é possível proceder com a recomendação de OAs levando em conta o estilo de aprendizagem de cada estudante. Define-se o Problema da Recomendação de Objetos de Aprendizagem (PROA) como um problema de cobertura de conjuntos. Não existe uma definição clássica para o PROA, porque a recomendação de OAs depende das demandas de cada contexto. Se o objetivo, por exemplo, for recomendar OAs associados a um dado assunto, um algoritmo simples de busca pode ser usado para filtrar aqueles OAs que abordam tal assunto.

No contexto deste trabalho, o objetivo é mais complexo. Visa-se atender a contextos onde se espera que os OAs recomendados cubram conceitos específicos. Desta forma, o PROA é um “problema de recomendação de OAs baseada em conceitos”. No contexto educacional, os conceitos que se espera que o aluno aprenda fazem parte comumente do plano de curso. O objetivo do PROA é recomendar os OAs que cobrem esses conceitos e satisfazem os parâmetros de busca do usuário. Esse problema é resolvido usando um Algoritmo Genético (AG), o qual trata-se de uma técnica de busca que procura soluções aproximadas para um problema de otimização e busca.

1.1 Motivação

Sabe-se que o computador pode adaptar conteúdos didáticos à maneira de aprender de cada aluno, contribuindo para que os estudantes tenham um melhor desempenho. Se isso ainda não é uma realidade para todos, deve-se principalmente ao fato de que os Sistemas Hiperídia Adaptativos possuem um alto custo de implementação, inclusive na criação do conteúdo educacional.

Uma expectativa inerente ao processo ensino-aprendizagem é que os materiais didáticos cubram os conceitos que se espera que os alunos aprendam. No contexto do ensino eletrônico, as pesquisas no estado da arte modelam esses conceitos usando ontologias. Essa modelagem de domínios do conhecimento é uma tarefa dispendiosa e envolve o conhecimento de especialistas.

Neste trabalho, visa-se diminuir o custo com tempo, pessoas e dinheiro por meio da automatização do processo de reúso de conteúdo wiki na criação de OAs e apresenta-se uma solução alternativa para o “problema da recomendação de objetos de aprendizagem baseada em conceitos” que não depende da modelagem de domínios do conhecimento.

1.2 Objetivos e Desafios da Pesquisa

O objetivo geral deste trabalho é estruturar adequadamente usando uma ontologia o conteúdo wiki que cobre conceitos de interesse do aprendiz facilitando a aprendizagem por meio da recomendação personalizada de OAs, derivados do conteúdo wiki, que atendem aos estilos de aprendizagem dos estudantes. Os objetivos específicos para alcançar essa meta são:

- a) Desenvolver uma ontologia para a modelagem de estudantes e OAs;
- b) Desenvolver e validar uma abordagem para a conversão automática de conteúdo wiki em OAs;
- c) Resolver o “problema da recomendação de OAs baseada em conceitos” garantindo que os OAs recomendados atendam a todos os conceitos que se espera que o aluno aprenda.

1.3 Hipóteses

As duas principais hipóteses deste trabalho com suas respectivas questões são:

1. As seções wiki podem ser convertidas em OAs.
 - a) O processo de criação de novos OAs pelo reúso de conteúdo wiki pode ser automatizado?
 - b) O conteúdo wiki tem boa qualidade?
2. O “problema da recomendação de objetos de aprendizagem baseada em conceitos” pode ser resolvido por um AG.

a) Os OAs recomendados cobrem todos os conceitos que o usuário deseja, atendendo ao estilo de aprendizagem do estudante?

1.4 Contribuições

Este trabalho contribui na automatização do processo de criação de novos OAs fazendo o reúso de conteúdo da Wikipédia e, conseqüentemente, favorece a diminuição do alto custo com tempo, dinheiro e pessoas no desenvolvimento de cursos *on-line* e materiais didáticos para o ensino presencial. Outras contribuições significativas são:

- a) Criação de uma abordagem para a conversão de conteúdo wiki em OAs que atendem ao perfil do estudante;
- b) Desenvolvimento de uma ontologia para:
 - a representação dos metadados de OAs em conformidade com o padrão IEEE-LOM e sua extensão CLEO;
 - a modelagem do estudante, inclusive do seu perfil de aprendizagem;
 - utilização de regras de inferências criadas para auxiliar na recomendação de OAs adequados ao perfil do estudante.
- c) Utilização de um AG para proceder com a recomendação dos OAs que atendem ao perfil do estudante e cobrem todos os conceitos que se espera que o estudante aprenda.

1.5 Organização da Dissertação

A fundamentação teórica que embasa este trabalho é tratada no Capítulo 2. A literatura correlata é discutida no Capítulo 3. A abordagem proposta é detalhada no Capítulo 4, no qual, define-se o Problema da Recomendação de Objetos de Aprendizagem e explicam-se os algoritmos criados. Os experimentos realizados são avaliados no Capítulo 5. O trabalho é concluído no Capítulo 6.

Fundamentação Teórica

A abordagem apresentada neste trabalho trata da recomendação personalizada de objetos de aprendizagem usando conteúdo da Wikipédia. A abordagem de recomendação utilizada é baseada em conteúdo. Para proceder com a recomendação, as seções wiki são transformadas em OAs. A qualidade do conteúdo wiki é um aspecto importante a ser considerado nesse processo. A Wikipédia é apresentada na Seção 2.1 e a estratégia de avaliação do seu conteúdo é explicada na Seção 2.2.

A Web Semântica (ver Seção 2.3) possui, dentre outras tecnologias, ontologias e regras de inferências (ver Seção 2.4) utilizadas, respectivamente, na modelagem do conhecimento e na inferências de novos conhecimentos. Neste trabalho, desenvolve-se uma ontologia para a representação dos metadados dos OAs criados e a modelagem de estudantes e seus perfis (ver Seção 2.5). As informações sobre os novos OAs criados são denominadas metadados, os quais seguem algum padrão que facilita a recuperação dos OAs. O padrão escolhido neste trabalho para modelar os OAs é o IEEE-LOM (ver Seção 2.6).

O PROA pode ser modelado como o PCC, o qual é apresentado na Seção 2.7. Este problema é resolvido usando um AG. Os fundamentos básicos de um AG são explicados na Seção 2.8 e uma solução clássica para o PCC é apresentada na Seção 2.9.

2.1 Wikipédia

A Wikipédia é uma grande enciclopédia digital. Na sua página principal, consultada em 30 de agosto de 2018, consta que ela possui um acervo com mais de 5 milhões e 708 mil artigos em inglês¹ e mais de 1 milhão de artigos em português². O conteúdo da Wikipédia pode ser copiado, modificado e redistribuído sob a licença Creative Commons BY-SA, desde que o novo conteúdo gerado respalde essas mesmas condições a terceiros e atribua crédito aos autores do artigo da Wikipédia que foi usado, o que pode ser feito citando um link direto para esse artigo.

¹ Disponível em: <https://en.wikipedia.org>

² Disponível em: <https://pt.wikipedia.org>

Os artigos são criados e editados de maneira colaborativa por qualquer pessoa. Nenhum artigo precisa ser criado em sua versão final. Pode-se criar um esboço e publicá-lo para que outros usuários contribuam para a sua melhoria até o artigo receber o mais alto status de qualidade.

A qualidade dos artigos é regulada pelas próprias políticas de criação e edição de artigos da Wikipédia. Alguns usuários que participam da comunidade podem editar e marcar artigos com rótulos de qualidade. Esses rótulos são úteis para os editores identificarem mais facilmente os artigos que precisam ser melhorados.

2.2 Avaliação da qualidade de páginas da Wikipédia

A comunidade da Wikipédia pode avaliar a qualidade dos artigos com quase uma dezena de rótulos que servem para classificar dos piores até os melhores artigos. Acontece que rotular esses artigos manualmente é um processo inviável, dado que, nos últimos anos, a quantidade de artigos da Wikipédia tem crescido exponencialmente e, associado a isso, artigos são editados a todo instante, o que pode alterar a qualidade do artigo. Para lidar com esse problema, diversas pesquisas vêm sendo desenvolvidas no que tange a avaliação automática de artigos wiki.

Os artigos rotulados manualmente pelos usuários da Wikipédia podem servir na composição de um *dataset* de treinamento para um classificador que seja capaz de identificar a qualidade de artigos wiki. Na literatura, há duas abordagens para a escolha dos atributos do *dataset*. Na primeira, extraem-se as características do conteúdo do artigo, enquanto que na outra abordagem, as características são extraídas do histórico de edições realizadas pelos usuários editores.

A extração de características é importante porque considerar todo o conteúdo do artigo além do conteúdo de seu histórico de edições torna o processo de classificação computacionalmente caro em termos de tempo, embora produza os melhores resultados. O trabalho de Dang e Ignat (2017), por exemplo, atinge o estado da arte com um classificador baseado em redes neurais que considera como entrada o artigo wiki inteiro. Porém o tempo gasto pelo classificador o torna inviável para os contextos em que o usuário espera uma resposta rápida.

Warncke-Wang, Cosley e Riedl (2013) utilizaram um classificador baseado em uma árvore de decisão para identificar a qualidade de artigos da Wikipédia. Os autores mostraram que com a extração de apenas cinco características das páginas wiki é possível obter resultados significativos. Essas características estão prontamente disponíveis e foram definidas pelos autores da seguinte maneira:

1. $\text{Completeness} = 0.4 * \text{NumBrokenWikilinks} + 0.4 * \text{NumWikilinks}$;
2. $\text{Informativeness} = 0.6 * \text{InfoNoise} + 0.3 * \text{NumImagens}$ (InfoNoise é o conteúdo da página propriamente dito, sem código de programação e sem *stopwords*);

3. NumHeadings: números de seções do artigo;
4. ArticleLength: tamanho do artigo;
5. NumReferences/ArticleLength: uma medida do número de citações por meio da contagem das *tags* <ref>, as quais são usadas em citações do tipo *footnote*. A divisão pelo tamanho do artigo (ArticleLength) é importante para amenizar a correspondência (quanto maior o artigo, maior o número de citações) que existe entre as duas variáveis.

Os autores também mostraram que esse modelo é compatível com aqueles que apresentam mais de 15 características. O classificador apresentado por Warncke-Wang, Cosley e Riedl (2013) é capaz de classificar os artigos wiki em sete classes diferentes, quais sejam: FA, GA, A, B, C, Start e Stub, todas definidas pela própria Wikipédia. Os detalhes de cada classe são apresentados na Tabela 4 do Capítulo 4. Os autores ainda dividem os artigos em duas grandes classes, os artigos GoodEnough, contendo as classes FA, GA e A, e NeedsWork, contendo as classes B, C, Start e Stub.

2.3 Web Semântica

A World Wide Web, ou simplesmente Web, evolui desde o seu surgimento. Popularizada a partir de 1995, a Web 1.0 permitia a comunicação por meio de páginas estáticas do tipo HyperText Markup Language (HTML). Essa linguagem permite que qualquer documento seja publicado na forma eletrônica e disponibilizado para o mundo todo. Data dessa época, o surgimento dos grandes portais como o UOL. Os primeiros sites brasileiros eram de notícias, depois surgiram os de entretenimento, compra e pesquisa.

No início do século XXI, surge a Web 2.0 que contava com uma arquitetura de desenvolvimento que permitiu que as aplicações fossem compartilhadas por toda a rede. Ela é uma Web participativa que promove a colaboração entre usuários e permite que o usuário final gere conteúdo utilizando blogs, wikis e redes sociais. Isso favoreceu o crescimento exponencial de informações, elevando a importância dos motores de busca.

O próximo desafio é a Web 3.0, que caminha para dar significado ao seu conteúdo por meio da estruturação e atribuição de semântica aos seus dados permitindo interpretar, conectar e disponibilizar os dados sob medida para a necessidade e o interesse de cada usuário. A Web 3.0, também denominada Web Semântica (BERNERS-LEE; HENDLER; LASSILA, 2001), estende a Web tradicional por meio de tecnologias que promovem a modelagem e manipulação do conhecimento pelas máquinas.

As linguagens eXtensible Markup Language (XML), Resource Definition Framework (RDF) e Web Ontology Language (OWL) são utilizadas para representar a informação na Web Semântica (WS) e torná-la interoperável entre aplicações, sendo a OWL a linguagem com o maior potencial de representatividade do conhecimento e a XML a linguagem com

menor poder de representação da informação. Os elementos da linguagem OWL estendem os elementos da linguagem RDF, os quais estendem os elementos da linguagem XML. A seguir, apresentam-se ordenadamente as camadas da arquitetura da WS começando pela mais abstrata.

1. Confiança
2. Prova
3. Lógica
4. Ontológica - OWL
5. RDF
6. XML
7. URI/IRI

Visualizando essa arquitetura em sete camadas, compreende-se mais facilmente como a Web Semântica estende a Web tradicional. Os elementos URI e sua generalização Internationalized Resource Identifier (IRI) são utilizados para identificar de maneira única os recursos da WS. A linguagem XML permite a criação de documentos formados por dados estruturados, aos quais a WS dá significado.

A camada RDF possui triplas em XML formadas por sujeito, predicado e objeto capazes de representar a semântica das informações da Web. A linguagem OWL aumenta o poder dessa representação semântica servindo para instanciar ontologias na Web. A camada ontológica potencializa a compreensão de informações da internet por computadores.

A recuperação de informações contidas nos grafos da WS é realizada por meio da linguagem de consulta para dados RDF, denominada SPARQL. Essa linguagem se parece muito com a SQL para bancos de dados relacionais, inclusive no que se refere às cláusulas (*select*, *from* e *where*), modificadores de consulta (*order by* e *distinct*) e operadores lógicos e de comparação.

A Semantic Web Rule Language (SWRL) é uma linguagem padrão da WS para a implementação de regras que podem ser usadas na inferência de novos conhecimentos pela camada lógica. As fases seguidas pelos *reasoners* nesse processo de inferência são utilizadas pela camada de prova para validar esse processo de inferência. Por fim, a camada de confiança almeja garantir que o conhecimento inferido seja confiável.

2.4 O desenvolvimento de ontologias

Gruber (1993, p. 199) afirma que “uma ontologia é uma especificação explícita de uma conceitualização”, a qual, por sua vez, corresponde a objetos, conceitos e outras entidades de uma dada área de interesse com suas relações. Consequentemente, uma ontologia pode ser utilizada para modelar os conceitos de um domínio de conhecimento e suas relações, bem como objetos de aprendizagem.

O Protégé (GENNARI et al., 2003) é um editor de ontologias de código aberto e também um framework para a construção de ontologias e gerenciamento de conhecimentos. Existem diversas metodologias para a criação de ontologias usando o Protégé. Destaca-se a metodologia Ontology Development 101 (NOY; MCGUINNESS, 2001), cuja popularidade se deve à sua facilidade de uso e orientações de boas práticas de desenvolvimento. As principais etapas de construção de uma ontologia seguindo essa metodologia são:

- Etapa 1. Determine o domínio e o escopo da ontologia;
- Etapa 2. Considere reutilizar ontologias existentes;
- Etapa 3. Enumere termos (conceitos) importantes da ontologia;
- Etapa 4. Defina as classes e a hierarquia de classes;
- Etapa 5. Defina as propriedades das classes;
- Etapa 6. Defina o objeto (*range*) das propriedades;
- Etapa 7. Crie as instâncias.

A metodologia Ontology Development 101 também orienta quanto à importância de criar uma convenção de nomes para os conceitos, tornando mais fácil o entendimento da ontologia. Embora não haja uma metodologia definitiva e nem mais correta, todas essas etapas e convenções contribuem para a legibilidade da ontologia por pessoas e facilitam a implementação correta da mesma.

No desenvolvimento de ontologias, podem-se também utilizar padrões. Por exemplo, o padrão Value Partition está associado à utilização de uma classe como uma partição. Uma classe é particionada por um grupo de subclasses se estas são mutualmente exclusivas (disjuntas) e cobrem completamente a classe pai. Por exemplo, uma classe de pessoas pode ser particionada pelas subclasses Homem e Mulher. Uma instância da subclasse Homem não pode pertencer à subclasse Mulher, e vice-versa. Além disso, se uma instância pertence à classe Pessoa, então necessariamente trata-se de um homem ou uma mulher.

2.5 Estilos de aprendizagem e o modelo de Felder e Silverman

Os estilos de aprendizagem referem-se ao conceito de que os indivíduos diferem em relação a qual modo de instrução ou estudo é mais adequado para eles (PASHLER et al., 2008). Um dos modelos mais utilizados na modelagem dos estilos de aprendizagem do estudante foi proposto por Felder e Silverman (1988). Esse modelo é comumente referido como Felder-Silverman Learning Style Model (FSLSM) e sua popularidade deve-se ao fato dele cobrir mais aspectos psicológicos do que outros modelos (DEBORAH; BASKARAN; KANNAN, 2014). O FSLSM cobre quatro dimensões polares:

1. Entrada: Visual (prefere imagens e diagramas) e Verbal (prefere escrita e explicações comentadas);
2. Organização: Sequencial (aprende com passos incrementais) e Global (pensamento holístico);
3. Percepção: Sensitivo (concreto e prático) e Intuitivo (conceitual e inovador);
4. Processamento: Ativo (gosta de trabalhar em grupo) e Reflexivo (pensa sobre seu aprendizado e gosta de trabalhar sozinho).

No e-learning, o questionário ILS (Index of Learning Styles) criado por Solomon e Felder (2005) é aplicado tradicionalmente para a identificação dos estilos de aprendizagem do estudante.

2.6 Padrões de metadados de objetos de aprendizagem

De um modo geral, um objeto de aprendizagem (OA) é um conteúdo digital que pode ser reutilizado em vários contextos de aprendizagem. Cada OA é composto por um conteúdo e seus metadados, os quais seguem algum padrão para que se garanta a reusabilidade. Esses metadados são armazenados em repositórios de OAs, facilitando o reúso dos conteúdos educacionais.

O padrão IEEE-LOM é reconhecido mundialmente por facilitar a criação e a busca de OAs. Esse padrão define nove categorias para a descrição de OAs na linguagem XML. Essas categorias são apresentadas na Tabela 1. Cada categoria é composta por um conjunto de campos relevantes para a identificação de um dado OA. Os valores assumidos por cada um desses campos são utilizados na recuperação/busca do recurso desejado.

O padrão IEEE-LOM serve de base para outros padrões, como o padrão de metadados OBAA (VICCARI et al., 2010) e o padrão SCORM (ADL, 2001). O padrão OBAA estende o padrão IEEE-LOM garantindo a interoperabilidade entre plataformas no contexto educacional brasileiro. O padrão SCORM (Sharable Content Objects Reference Model) fornece interoperabilidade entre diferentes sistemas de gerenciamento da aprendizagem.

O padrão SCORM objetiva a portabilidade ao permitir que diversos tipos de objetos de aprendizagem se comuniquem de maneira eficiente com as diversas plataformas de *e-learning* existentes. O padrão viabiliza o encapsulamento e sequenciamento de OAs de diversos tamanhos, tais como imagens, exercícios ou mesmo um curso completo, facilitando o reúso desse conteúdo entre as diversas plataformas compatíveis com esse padrão.

O padrão IEEE-LOM para descrição de metadados pode ser incompleto em alguns contextos e, por outro lado, descrever metadados não muito usados. Para solucionar o problema da sua incompletude, há algumas extensões do padrão IEEE-LOM, como a extensão definida pelo CLEO. As contribuições do CLEO em relação ao IEEE-LOM são:

Tabela 1 – Categorias do padrão IEEE-LOM

Categoria	Descrição
1. Geral	Agrega informações gerais que descrevem o OA, tais como identificador, título e palavras-chave.
2. Ciclo de Vida	Descreve o histórico do OA desde sua criação até seu estado atual usando atributos como versão e contribuição, a qual se refere às pessoas ou organizações que contribuíram na criação, edição ou publicação do OA.
3. Meta-metadados	Reúne atributos sobre os metadados, tais como contribuição, data de criação e a língua em que eles são descritos.
4. Técnico	Inclui as características técnicas do OA, tais como formato, tamanho, duração e observações de instalação.
5. Educacional	Descreve as características pedagógicas do OA, tais como tipo de interatividade, tipo do recurso de aprendizagem, densidade semântica, faixa etária, grau de dificuldade e tempo de aprendizagem.
6. Direitos	Agrega os direitos de propriedade intelectual e as condições de uso do OA.
7. Relação	Define os relacionamentos do OA com outros OAs, como por exemplo: “é parte de”, “é baseado em” e “é requerido por”.
8. Anotação	Provê comentários de pessoas que já utilizaram o OA.
9. Classificação	Classifica o OA dentro de uma taxonomia.

- 1) Vocabulários adicionais para aperfeiçoar o nível de agregação (categoria geral);
- 2) Vocabulários alternativos para os tipos de recursos de aprendizagem (categoria educacional) e o propósito (categoria classificação);
- 3) Novos elementos para a categoria educacional.

Um dos incrementos propostos pelo CLEO está relacionado ao tipo de recurso de aprendizagem. O IEEE-LOM define 15 tipos de recursos diferentes:

1. *Exercise*: Atividades para fixação e aplicação dos conteúdos estudados;
2. *Simulation*: Utilizada para o estudo da realidade projetada em modelos computacionais;
3. *Questionnaire*: Instrumento para coleta de informações;
4. *Diagram*: Representação visual estruturada e simplificada de algum conceito ou ideia;
5. *Figure*: Ilustração utilizada para acrescentar informação ou sintetizar um texto;
6. *Graph*: Utilizado para exibir dados, demonstrar padrões e tendências;
7. *Index*: Agregação dos conceitos de um trabalho com referência às páginas nas quais cada conceito é encontrado;
8. *Slide*: Corresponde a uma página de uma apresentação que pode conter textos e imagens;
9. *Table*: Utilizada para apresentação de dados, geralmente numéricos, em linhas e colunas;
10. *Narrative text*: Texto que possui tempo, espaço, drama e resolução. Utilizado para contar histórias, eventos passados e entreter pessoas;

11. *Exam*: Prova utilizada para avaliar o conhecimento dos aprendizes;
12. *Experiment*: Conjuntos de testes realizados para se verificar uma hipótese;
13. *Problem statement*: É a descrição de um problema, utilizada para a compreensão do mesmo;
14. *Self-assessment*: A autoavaliação é utilizada como uma ferramenta de autoconhecimento;
15. *Lecture*: Apresentação oral para informar a respeito de um assunto.

O CLEO acrescenta a eles mais 29 tipos, dentre os quais, os tipos:

1. *Additional Resource*: Informações adicionais fornecidas ao aluno;
2. *Assessment*: Avalia o desempenho do aluno;
3. *Definition*: Dá o significado de um termo;
4. *Example*: Demonstra um conhecimento;
5. *Introduction*: Informação introdutória.

2.7 O Problema da Cobertura de Conjuntos

Dada uma matriz binária a_{ij} com m linhas e n colunas, o Problema da Cobertura de Conjuntos (PCC) é um problema de cobertura de linhas por um subconjunto das colunas com custo mínimo. Seja $x_j = 1$, se a coluna j (com custo $c_j > 0$) está na solução, e $x_j = 0$, caso contrário, então o PCC é definido como:

$$\text{Minimize } \sum_{j=1}^n c_j x_j \quad (1)$$

$$\text{Sujeito a } \sum_{j=1}^n a_{ij} x_j \geq 1, \quad i = 1, \dots, m \quad (2)$$

$$x_j \in \{0, 1\}, \quad j = 1, \dots, n \quad (3)$$

A restrição da Eq. (2) garante que cada linha seja coberta por ao menos uma coluna. Se todos os coeficientes de custo c_j são iguais, então o problema é denominado PCC de custo único e, nesse caso, o custo de uma solução está estritamente relacionado ao número de elementos da solução. Note que, por menor que seja o custo de uma coluna, ela pode contribuir para uma melhor solução apenas se ela cobrir ao menos um conceito.

2.8 Algoritmo genético

Um AG faz parte de uma família de modelos computacionais inspirados pela evolução biológica (WHITLEY, 1994). Essa família de procedimentos se fundamenta em analogias

a conceitos biológicos como seleção natural, hereditariedade, mutação e recombinação de genes (*crossover*) para encontrar soluções aproximadas em problemas de busca e otimização.

Os algoritmos genéticos seguem a ideia de sobrevivência do mais adaptado. A primeira preocupação na utilização dos AGs é com relação à representação computacional dos cromossomos. Cada potencial solução é codificada em uma estrutura de dados simples semelhante ao cromossomo. Cada cromossomo consiste de genes (uma sequência de DNA) que determinam as características de um indivíduo. Durante o processo de reprodução (cruzamento ou *crossover*) ocorre a recombinação dos genes dos pais para a geração de ao menos um indivíduo filho. Os descendentes criados podem sofrer mutações que alteram os seus genes. A adaptação de cada indivíduo pode ser medida pelo seu sucesso durante a vida. A Figura 1 modela as etapas fundamentais de um algoritmo genético.

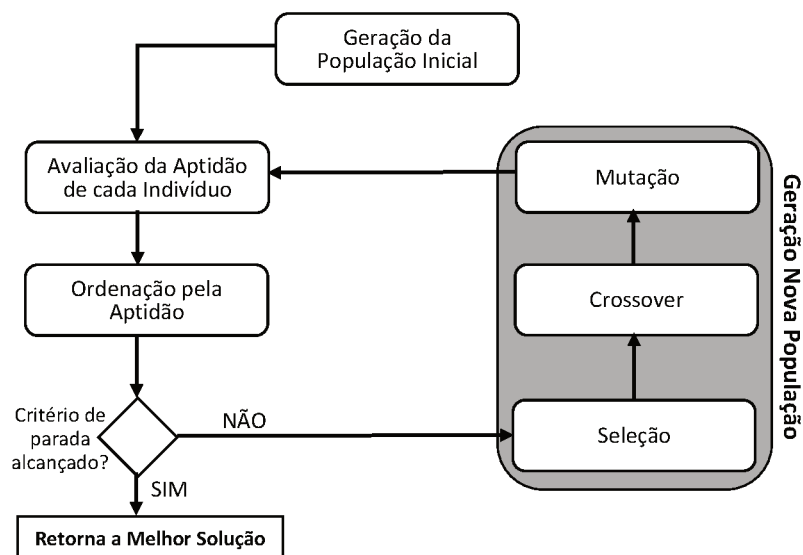


Figura 1 – Etapas fundamentais de um algoritmo genético

Primeiramente, inicializa-se uma população inicial com n indivíduos. Soluções (indivíduos) de uma população são escolhidas e usadas para formar uma nova população (reprodução). Espera-se que a nova população seja melhor que a anterior. As soluções utilizadas na geração de novos filhos são escolhidas de acordo com uma função de adaptação vinculada à função objetivo.

2.9 Algoritmo genético para o Problema da Cobertura de Conjuntos

Nesta seção, apresenta-se um algoritmo genético proposto por Beasley e Chu (1996) para o Problema da Cobertura de Conjuntos (PCC) definido na Seção 2.7. No AG que

resolve o PCC, assume-se, sem perda de generalidade, que as colunas das instâncias do PCC estão ordenadas em ordem crescente de custo e que as colunas de igual custo são ordenadas em ordem decrescente do número de linhas que elas cobrem.

2.9.1 Representação do indivíduo

O primeiro passo no projeto de um algoritmo genético é conceber um esquema de representação adequado. A representação binária é uma escolha usual. O AG que resolve o PCC utiliza um vetor binário de n bits como a estrutura do cromossomo de cada indivíduo (ver Figura 2), onde n é o número de colunas no PCC. O valor 1 no i -ésimo bit implica que a coluna i está na solução.

Colunas (genes)	1	2	3	4	5	...	$n-1$	n
Vetor de inteiros	1	1	0	1	0	...	1	0

Figura 2 – Representação binária do cromossomo de um indivíduo

Fonte: Adaptado de Beasley e Chu (1996, p. 394)

2.9.2 Função de avaliação

A aptidão de um indivíduo está diretamente associada ao valor da sua função objetivo. Com a representação binária, a *fitness* de um indivíduo i é calculada por:

$$f_i = \sum_{j=1}^n c_j s_{ij} \quad (4)$$

onde s_{ij} é o valor do j -ésimo bit (coluna) no vetor correspondente ao i -ésimo indivíduo e c_j é o custo do bit (coluna) j . O cálculo do c_j depende do contexto de cada trabalho. Na Seção 4.1 da proposta desta dissertação, o PROA é associado ao PCC e o cálculo do c_j é detalhado.

2.9.3 Técnica de seleção

Existem várias técnicas de seleção dos pais. O AG que resolve o PCC utiliza o método de seleção por torneio, que funciona formando dois conjuntos de indivíduos, cada um consistindo de k indivíduos sorteados da população aleatoriamente. Dois indivíduos com a melhor *fitness*, cada um selecionado de um dos dois conjuntos, são escolhidos para o cruzamento. Usar um valor maior para k tem o efeito de aumentar a pressão de seleção nos indivíduos mais aptos. O AG do PCC utiliza o método da seleção por torneio binário, ou seja, $k = 2$, pois ele pode ser implementado de maneira eficiente e possui uma boa performance.

2.9.4 Cruzamento

Em um AG tradicional, os operadores simples de *crossover*, como o *crossover* de um ou dois pontos, são usados com frequência. Esses operadores de *crossover* funcionam gerando aleatoriamente um ou mais pontos de *crossover* e, em seguida, trocando partes dos dois vetores pais para produzir dois vetores filhos. O *crossover* de um ponto (ver Figura 3) é formalmente definido a seguir.

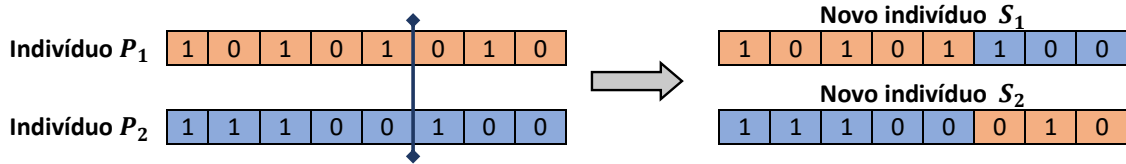


Figura 3 – Crossover de um ponto entre cromossomos binários

Sejam P_1 e P_2 os vetores pais $P_1[1], \dots, P_1[n]$ e $P_2[1], \dots, P_2[n]$, respectivamente. Gere um ponto de *crossover* k , $1 < k \leq n$. Então os vetores filhos S_1 e S_2 são:

$$S_1 = P_1[1], \dots, P_1[k], P_2[k+1], \dots, P_2[n],$$

$$S_2 = P_2[1], \dots, P_2[k], P_1[k+1], \dots, P_1[n].$$

O operador de *crossover* de dois pontos é semelhante ao operador de *crossover* de um ponto, exceto pelo fato de gerar dois pontos de cruzamento em vez de um. Esses operadores não são muito eficientes no contexto do AG do PCC. À medida que o AG converge, espera-se que a maioria dos genes na porção direita das cadeias P_1 e P_2 tenha um valor igual a zero. Isso ocorre porque as soluções do PCC consistirão principalmente em colunas de baixo custo neste estágio (lembre-se de que as colunas são ordenadas em ordem crescente de custo). Portanto, se o ponto de cruzamento for escolhido em qualquer lugar dentro dessa região, as soluções filhas resultantes serão idênticas às suas soluções pais.

Além disso, quando o AG converge, as estruturas da solução variam pouco de uma para outra. Sendo assim, a capacidade dos operadores de *crossover* de um ponto e dois pontos para gerar novas estruturas a partir das estruturas pais pode ser limitada.

Outra técnica de *crossover* comumente usada é o operador de *crossover* uniforme. Cada gene na solução filho é criado copiando o gene correspondente de um ou outro pai, escolhido de acordo com um gerador de números binários aleatórios. Se o número aleatório for 0, o gene é copiado do primeiro pai, se for 1, o gene é copiado do segundo pai.

O operador de fusão, proposto por Beasley e Chu (1996), é um operador de *crossover* incremental (ver Figura 4) que leva em conta a estrutura e a *fitness* relativa das soluções pais. O operador de fusão produz apenas um único filho (diferentemente dos operadores de *crossover* de um ponto e dois pontos, onde dois filhos são produzidos) e é definido a seguir.

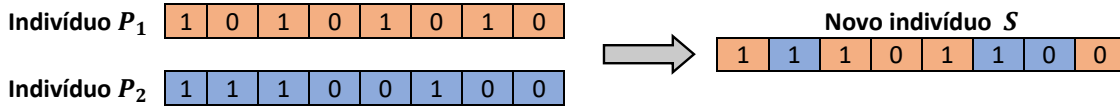


Figura 4 – Crossover incremental entre cromossomos binários

Sejam f_{p_1} e f_{p_2} as aptidões dos vetores pais P_1 e P_2 , respectivamente, e S o vetor filho. Assumindo um problema de minimização, então para todo $i = 1, \dots, n$:

- a) se $P_1[i] = P_2[i]$, então $S[i] = P_1[i] = P_2[i]$,
- b) se $P_1[i] \neq P_2[i]$, então:
 - $S[i] = P_1[i]$ com probabilidade $p = f_{p_2}/(f_{p_1} + f_{p_2})$,
 - $S[i] = P_2[i]$ com probabilidade $1 - p$.

Por exemplo, se $f_{p_1} = 3$, $f_{p_2} = 7$ e $P_1[i] \neq P_2[i]$, a probabilidade de $S[i] = P_1[i]$ é $7/(3 + 7) = 0.7$ e a probabilidade de $S[i] = P_2[i]$ é $1 - 0.7 = 0.3$.

Esse operador de *crossover* baseia-se na aptidão relativa dos dois pais na escolha de quais valores genéticos serão passados para o filho, supondo que a herança de um gene particular de um pai mais apto provavelmente contribui mais para a aptidão geral do filho do que a de um pai menos apto. A vantagem desse operador de fusão baseado em aptidão, em relação aos operadores de *crossover* de um ponto e dois pontos, é a sua capacidade de gerar novas soluções quando as duas soluções pais são semelhantes em estrutura.

2.9.5 Mutação

A mutação é um operador importante na expansão aleatória do espaço de busca e funciona invertendo cada bit na solução com uma pequena probabilidade. Estudos recomendam que ao menos um bit de cada solução sofra mutação. Beasley e Chu (1996) mostraram que, para que o AG do PCC seja efetivo, uma taxa de mutação maior é necessária quando o AG converge. Sendo assim, em um estágio inicial do AG, o operador de *crossover* é o principal responsável pela busca e, portanto, a taxa de mutação é definida em um valor baixo para permitir o mínimo de interrupção. Conforme o AG progride, o operador de *crossover* se torna menos produtivo e, portanto, a taxa de mutação aumenta. Quando o AG finalmente converge, a taxa de mutação também se tornará estável a uma taxa constante.

Esse esquema de mutação é calculado por:

$$\left\lceil \frac{m_f}{1 + \exp(-4m_g(t - m_c)/m_f)} \right\rceil \quad (5)$$

onde t é o número de soluções filhas que foram geradas, m_f especifica a taxa de mutação estável final, m_c especifica o número de soluções filhas esperadas em que uma taxa de

mutação de $m_f/2$ é alcançada e m_g especifica o gradiente em $t = m_c$. Como o número de bits mutados só pode assumir valores inteiros, o valor de retorno na Eq. (5) é arredondado para o valor inteiro mais próximo. O valor de m_f é especificado pelo usuário e os valores de m_c e m_g são determinados com base na taxa na qual o AG converge para um problema específico.

Para exemplificar o uso da Eq. (5), considere $m_g = 2.0$, $m_c = 200$ e $m_f = 10$. Então, para $t \leq 197$, a Eq. (5) resulta em 1, ou seja, apenas um bit da solução é escolhido para ser mutado. Se $t = 198$, dois bits são mutados, se $t = 199$, quatro bits são mutados, se $t = 200$, cinco bits são mutados, se $t = 201$, sete bits são mutados, se $t = 202$, nove bits são mutados, e finalmente, para $t \geq 203$, dez bits são mutados.

Note que a variação na taxa de mutação ocorre para valores bem próximos de m_c e a estabilização da taxa de mutação (m_f) ocorre para $t \geq 203$. Desse modo, os primeiros indivíduos gerados possuem apenas um bit mutado e quando o AG converge a taxa de mutação aumenta e depois se estabiliza. Após alguns experimentos, Beasley e Chu (1996), sugerem que o esquema de mutação seja calculado considerando $m_g = 2.0$, $m_c = 200$ e $5 \leq m_f \leq 10$.

2.9.6 Soluções viáveis

As soluções geradas pelos operadores de fusão e mutação podem violar as restrições do problema, ou seja, algumas linhas podem ficar descobertas. Beasley e Chu (1996) apresentaram operadores adicionais para tornar todas as soluções viáveis. Os autores propuseram um operador heurístico capaz de manter a viabilidade das soluções geradas e uma etapa adicional de otimização local para tornar o AG mais eficaz.

As etapas necessárias para tornar viável cada solução envolvem a identificação de todas as linhas descobertas e a adição de colunas, de modo que todas as linhas sejam cobertas. A busca por essas colunas ausentes é baseada na proporção

$$\frac{\text{custo de uma coluna}}{\text{número de linhas descobertas que ela cobre}} \quad (6)$$

Depois que as colunas são adicionadas e uma solução se torna viável, uma etapa de otimização local é aplicada para remover quaisquer colunas redundantes na solução. Uma coluna é redundante se ao ser removida da solução, a solução ainda permanece viável. O algoritmo segue abaixo.

Sejam:

I = o conjunto de todas as linhas,

J = o conjunto de todas as colunas,

α_i = o conjunto das colunas que cobrem a linha i , $i \in I$,

β_j = o conjunto de linhas cobertas pela coluna j , $j \in J$,

S = o conjunto de colunas na solução,

U = o conjunto de linhas não cobertas,

w_i = o número de colunas, em S , que cobrem a linha i , $i \in I$.

1. Inicialize $w_i = |S \cap \alpha_i|$, $\forall i \in I$.
2. Inicialize $U = \{i \mid w_i = 0, \forall i \in I\}$.
3. Para cada linha i em U (em ordem crescente de i):
 - a) encontre a primeira coluna j (em ordem crescente de j) em α_i que minimiza $c_j/|U \cap \beta_j|$,
 - b) adicione j a S e faça $w_i = w_i + 1, \forall i \in \beta_j$. Faça $U = U - \beta_j$.
4. Para cada coluna j em S (em ordem decrescente de j), se $w_i \geq 2, \forall i \in \beta_j$, faça $S = S - j$ e $w_i = w_i - 1, \forall i \in \beta_j$.
5. S é agora uma solução viável para o PCC que contém nenhuma coluna redundante.

Os passos 1 e 2 identificam as linhas descobertas. As etapas 3 e 4 são heurísticas gulosas, pois, na etapa 3, as colunas com baixo custo estão sendo consideradas primeiro e, na etapa 4, colunas com altos custos são descartadas primeiro sempre que possível.

2.9.7 Visão geral do algoritmo

O algoritmo genético para o PCC proposto por Beasley e Chu (1996) pode ser resumido pelas etapas a seguir.

1. Gere uma população inicial de N soluções randômicas. Faça $t = 0$.
2. Selecione duas soluções P_1 e P_2 da população usando a seleção por torneio binário.
3. Combine P_1 e P_2 para formar uma nova solução C usando o operador de *crossover* por fusão.
4. Proceda com a mutação de k colunas escolhidas aleatoriamente em C , onde k é determinado pelo esquema de mutação variável.
5. Torne C viável e remova colunas redundantes em C aplicando o operador heurístico.
6. Se C é idêntico a qualquer uma das soluções na população, vá para a etapa 2; caso contrário, faça $t = t + 1$ e vá para a etapa 7.
7. Substitua uma solução escolhida randomicamente com uma *fitness* acima da média na população por C .
8. Repita as etapas 2-7 até que $t = M$ soluções não duplicadas tenham sido geradas. A melhor solução encontrada é aquela com a menor *fitness* na população.

Assim que uma nova solução filha é gerada, ela substitui algum indivíduo da população com *fitness* acima da média. Note que quanto menor a *fitness*, melhor o desempenho do indivíduo. Esse é um método de substituição incremental (em inglês, *steady-state replacement method*). A vantagem desse método é que as melhores soluções são sempre mantidas

na população e a nova solução mais recentemente gerada é prontamente disponibilizada para seleção e reprodução, acelerando a convergência do AG.

Cada indivíduo da população inicial é gerado conforme o método a seguir:

1. Inicialize $S_p = \emptyset$. Inicialize $w_i = 0, \forall i \in I$.
2. Para cada linha i em I :
 - a) selecione randomicamente uma coluna j em α_i ,
 - b) adicione j a S_p e faça $w_i = w_i + 1, \forall i \in \beta_j$.
3. Faça $T = S_p$.
4. Selecione aleatoriamente uma coluna j , $j \in T$ e faça $T = T - j$. Se $w_i \geq 2, \forall i \in \beta_j$, faça $S_p = S_p - j$ e faça $w_i = w_i - 1, \forall i \in \beta_j$.
5. Repita a etapa 4 até $T = \emptyset$.

A etapa 2 garante a geração de soluções viáveis e a etapa 4 elimina colunas redundantes, assim como o operador de viabilidade heurístico, porém as colunas são eliminadas de maneira aleatória, em vez de se considerar o custo.

Trabalhos Relacionados

Neste capítulo, discutem-se os principais trabalhos relacionados à recomendação de recursos educacionais. O Problema da Recomendação de Objetos de Aprendizagem (PROA), no contexto deste trabalho, é definido como um problema de cobertura de conjuntos (ver Seção 4.1). Como cada contexto possui uma demanda diferente, não existe uma definição clássica para o PROA. Portanto, os trabalhos relacionados à recomendação de recursos educacionais usam abordagens e parâmetros diferentes para alcançarem seus objetivos.

Os trabalhos relacionados à recomendação de recursos educacionais são apresentados na Seção 3.1. Nesta pesquisa, não se encontrou algum trabalho que apresenta uma solução efetiva para o PROA. Porém, há trabalhos que apresentam soluções para problemas semelhantes. Eles são abordados na Seção 3.1.1.

Para resolver o PROA, implementa-se um AG inspirado em algumas etapas do AG (da Seção 2.9) que resolve o PCC proposto por Beasley e Chu (1996). De fato, o AG que resolve o PROA resolve também o PCC, dada a forte relação teórica existente entre os dois problemas.

Embora haja soluções mais modernas para o PCC, elas apresentam mais contribuições teóricas do que práticas. Os diversos algoritmos que implementam uma solução aproximada para o PCC são comparados quanto à qualidade da solução e quanto ao tempo de execução. As comparações são realizadas quase sempre com o *benchmark* OR-Library¹, que possui instâncias grandes do PCC. O AG dos autores Beasley e Chu (1996) apresenta para essas instâncias soluções muito boas de maneira rápida. Assim, algumas características desse AG foram exploradas para a criação do AG que resolve o PROA.

Neste capítulo, apresentam-se ainda, na Seção 3.2, os trabalhos relacionados ao uso de ontologias e/ou da Web na recomendação de OAs e pesquisas que utilizaram a Wikipédia na aprendizagem.

¹ Disponível em: <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/scpinfo.html>

3.1 Recomendação de recursos educacionais

Nos últimos anos, diversos trabalhos têm tratado da recomendação de recursos educacionais explorando uma variedade de tecnologias. Eles utilizam a recomendação de OAs com o objetivo de personalizar o ensino, levando em conta as preferências, interesses e trajetórias dos estudantes. O modelo FSLSM é o mais utilizado na modelagem dos estilos de aprendizagem dos estudantes (NASCIMENTO et al., 2017).

Os sistemas de recomendação são classificados de acordo com a técnica utilizada na recomendação. No contexto do ensino eletrônico, as técnicas de recomendação são divididas comumente em três categorias: filtragem colaborativa, baseada em conteúdo e híbrida.

Na filtragem colaborativa, os OAs recomendados para um grupo de estudantes com preferências e perfis similares são recomendados para novos estudantes que possuem características similares às preferências dos estudantes desse grupo. Na filtragem baseada em conteúdo, a recomendação de OAs leva em conta a similaridade entre o conteúdo dos OAs e o perfil dos estudantes.

A recomendação híbrida combina a filtragem colaborativa com a baseada em conteúdo para se beneficiar das vantagens de ambas as técnicas aumentando a performance. Alguns trabalhos mostraram que a combinação de diferentes técnicas de recomendação de fato aumenta a performance (CHEN et al., 2014; NILASHI; IBRAHIM; ITHNIN, 2014).

Tarus, Niu e Khadidja (2017) propuseram uma técnica de recomendação baseada em ontologia e filtragem colaborativa para a recomendação personalizada de materiais de aprendizagem. A ontologia utilizada modela características do estudante, tais como estilos de aprendizagem e níveis de conhecimento e de habilidade.

Klašnja-Milićević, Vesin e Ivanović (2018) apresentaram um método de recomendação personalizada de OAs que utiliza algoritmos baseados no uso das chamadas “tags mais populares”. As tags são criadas de maneira colaborativa e são utilizadas para melhorar a busca e recomendação de recursos educacionais.

Tarus, Niu e Kalui (2018) propuseram uma abordagem de recomendação híbrida que combina o contexto computacional, mineração de padrões sequenciais e algoritmos de filtragem colaborativa para a recomendação de recursos educacionais. O contexto computacional é utilizado para agregar informações contextuais sobre o aprendiz, tais como nível de conhecimento e metas de aprendizagem. Um algoritmo de mineração de padrões sequenciais é utilizado para minerar *logs* da Web e descobrir os padrões de acesso sequencial do estudante.

Xiao et al. (2018) desenvolveram um sistema para a recomendação de recursos educacionais para aprendizes inscritos em cursos *on-line* formais. O sistema de recomendação combina regras de associação e as filtrações colaborativa e baseada em conteúdo para a recomendação personalizada de materiais de aprendizagem, levando em conta o perfil do estudante, seu histórico de navegação e o tempo gasto em um determinado conteúdo no sistema de aprendizagem *on-line*.

Christudas, Kirubakaran e Thangaiah (2018) apresentaram uma abordagem baseada em um algoritmo genético para a recomendação personalizada de OAs considerando os estilos de aprendizagem dos estudantes. O AG implementado pelos autores é uma forma modificada do AG tradicional. Esse AG força a compatibilidade dos OAs dentro do espaço de busca, ajudando a reduzi-lo e a preenchê-lo com melhores cromossomos. Esse AG supera o AG tradicional e seu desempenho é comparável ao método de otimização por enxame de partículas.

Nesta dissertação, o AG implementado também reduz o espaço de busca, aceitando na população apenas soluções viáveis. Trabalhos como o de Christudas, Kirubakaran e Thangaiah (2018) mostram que a redução do espaço de busca e otimizações locais tornam os AGs comparáveis a técnicas mais modernas de otimização e busca.

3.1.1 Recomendação de objetos de aprendizagem baseada em conceitos

Há problemas bem similares ao PCC que são utilizados na modelagem de problemas de recomendação de recursos educacionais. Nos trabalhos de Acampora et al. (2008), Acampora, Gaeta e Loia (2010), Acampora et al. (2011) e Gaeta et al. (2013), os autores modelam o PROA como um problema de locação de facilidades, cujo objetivo é alocar m atividades didáticas (como OAs) a n conceitos do caminho de aprendizagem (os conceitos são ordenados por suas relações de pré-requisitos).

Esse problema pode ser formalizado como um problema de Programação Linear Inteira. Seja y_i ($i = 1, \dots, m$) um vetor binário que assume o valor 1 se o recurso de aprendizagem i é utilizado, e 0, caso contrário. Seja x_{ij} uma matriz binária que assume valor 1 se o conceito j é coberto pelo OA i , e 0, caso contrário. Assim, o problema é definido como:

$$\text{Minimize } \sum_{i=1}^m p(i)y_i + \sum_{i=1}^m \sum_{j=1}^n d(i,j)x_{ij} \quad (7)$$

$$\text{Sujeito a } \sum_{j=1}^n x_{ij} = 1 \quad i = 1, \dots, m \quad (8)$$

$$x_{ij} \leq y_i \quad i = 1, \dots, m \quad j = 1, \dots, n \quad (9)$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \dots, m \quad j = 1, \dots, n \quad (10)$$

$$y_i \in \{0, 1\} \quad i = 1, \dots, m \quad (11)$$

onde $p(i)$ representa o custo da introdução do i -ésimo OA na sequência de OAs recomendados ao estudante e as distâncias $d(i, j)$ são calculadas por uma função que compara as

preferências de aprendizagem do estudante com os valores de metadados do i -ésimo OA que cobre o conceito j .

Acampora et al. (2008) implementaram um algoritmo memético cujas principais etapas são a busca genética, com as operações de *crossover* e mutação, e a busca local visando a otimização das soluções encontradas. O sistema proposto pelos autores possui o Modelo do Conhecimento, o Modelo do Aprendiz e o Modelo Didático, os quais interagem por meio de processos específicos visando-se a recomendação personalizada. O Modelo do Conhecimento utiliza ontologias que modelam conceitos e suas relações, os quais são fundamentais na construção do caminho de aprendizagem. Mais tarde, Acampora, Gaeta e Loia (2010) utilizaram a otimização memética em um sistema multiagente.

Acampora et al. (2011) resolveram o problema de recomendação da Eq. (7) utilizando um algoritmo memético que utiliza técnicas evolucionárias seguidas de estratégias de busca local. Dentre as técnicas de otimização evolucionárias, eles utilizaram AG e o método do enxame de partículas, as quais são executadas em paralelo pelo Agente Evolucionário. O Agente de Busca Local controla a execução em paralelo das estratégias de busca local, tais como a Busca Tabu e a meta-heurística *Simulated Annealing*. A troca de soluções pelos métodos de otimização é controlada por um conjunto de regras *fuzzy*.

A solução proposta por Acampora et al. (2011) é melhor empregada no contexto da Web 2.0, que é caracterizada por repositórios distribuídos, uma variedade de atividades educacionais e caminhos de aprendizagem repletos de assuntos. O mesmo problema, Eq. (7), foi resolvido por Gaeta et al. (2013) usando simplesmente um algoritmo guloso. Contudo, os autores alertaram que embora essa seja uma abordagem rápida, não garante resultados muito bons.

Todos esses trabalhos utilizam uma abordagem de recomendação baseada em conceitos, apresentando uma solução para o problema definido na Eq. (7). Os conceitos são modelados em ontologias de domínios do conhecimento. Dada a definição de Gruber (1993), que afirma que “uma ontologia é uma especificação explícita de uma conceitualização”, é natural pensar que uma ontologia pode ser utilizada, principalmente, para modelar conceitos e suas relações. A relação do tipo “pré-requisito”, por exemplo, é utilizada para ordenar conceitos levando-se em conta que a aprendizagem de um dado conceito depende do conhecimento de outro. A lista de conceitos ordenados que se espera que o aluno aprenda é comumente denominada “caminho de aprendizagem”.

A construção de um caminho de aprendizagem, geralmente, tem um alto custo. A criação de domínios do conhecimento é uma tarefa dispendiosa que envolve o conhecimento de especialistas. Esse problema é uma das principais motivações desta dissertação, que visa a recomendação de OAs usando uma abordagem baseada em conceitos sem a necessidade de utilizar domínios do conhecimento.

3.2 Reúso de conteúdo da Web e a utilização de tecnologias da Web Semântica

Nesta seção, apresentam-se os trabalhos relacionados ao uso de ontologias e/ou da Web na recomendação de OAs e trabalhos que utilizaram a Wikipédia na aprendizagem. Embora a literatura correlata careça de trabalhos que implementem ontologias para representar OAs, há muitos trabalhos que as utilizam no contexto da aprendizagem para modelar domínios de conhecimento e estudantes.

A literatura correlata também carece de trabalhos que fazem o reúso de conteúdo da Web na criação de objetos de aprendizagem explorando tecnologias da WS e ontologias. Contudo há trabalhos que implementam parte desse processo. A Tabela 2 apresenta oito desses trabalhos relacionando-os com a abordagem apresentada nesta pesquisa.

Tabela 2 – Comparação da literatura correlata com a proposta deste trabalho

Referência	Criação de OAs	Padrões de metadados	Reúso de conteúdo da Web	Ontologia ou tecnologias da WS	Técnica de recomendação de OAs
Menolli, Malucelli e Reinehr (2011)	Sim	IEEE-LOM	Sim	Não	Não
Jácome Júnior, Mendes Neto e Silva (2012)	Não	SCORM	Não	Sim	Algoritmo Genético
Gaeta et al. (2013)	Não	IEEE-LOM	Não	Sim	Algoritmo Guloso
Limongelli, Gasparetti e Sciarrone (2015)	Não	Não	Sim	Não	Filtragens por conteúdo e colaborativa
Abech et al. (2016)	Não	IEEE-LOM	Sim	Sim	Regras SWRL e consultas SPARQL
Ramírez-Arellano, Bory-Reyes e Hernández-Simón (2017)	Sim	SCORM	Sim	Não	Filtragem baseada em conteúdo
Ouf et al. (2017)	Não	Não	Não	Sim	Regras SWRL
Pereira et al. (2018)	Não	IEEE-LOM	Sim	Sim	Consultas SPARQL
Nossa proposta	Sim	IEEE-LOM CLEO	Sim	Sim	Regras SWRL e Algoritmo Genético

No processo de recomendação de OAs, os autores Gaeta et al. (2013) (ver Subseção 3.1.1) utilizaram ontologias para modelar domínios de conhecimento. Jácome Júnior, Mendes Neto e Silva (2012) implementaram ontologias para representar os perfis estático e dinâmico dos estudantes.

Em relação a Wikipédia, há muitos trabalhos da literatura correlata que a utilizaram na aprendizagem. Para citar um exemplo, Limongelli, Gasparetti e Sciarrone (2015)

desenvolveram um módulo de um sistema para auxiliar professores na construção de cursos que contêm apenas páginas da Wikipédia. O processo de recomendação das páginas considera as filtragens por conteúdo e social, na qual o professor pode se beneficiar das escolhas já realizadas por outros usuários que possuem um estilo de ensino parecido com o seu.

Por outro lado, encontrou-se apenas um trabalho na literatura em que os autores, Menolli, Malucelli e Reinehr (2011), propuseram uma abordagem para gerar objetos de aprendizagem a partir de páginas wiki, utilizando tecnologias semânticas e o padrão IEEE-LOM. Na abordagem proposta por eles, utilizam-se técnicas de mineração de texto nas páginas wiki para a geração de alguns metadados, os quais são revisados por um especialista humano, que pode criar os demais metadados manualmente. Essa abordagem não utiliza ontologias e tecnologias da WS. Além disso, não se encontrou nenhum trabalho que a implementa.

Abech et al. (2016) desenvolveram um modelo, denominado EduAdapt, para recomendar os OAs mais apropriados ao contexto e estilos de aprendizagem do estudante. O modelo pode ser acoplado a ambientes de aprendizagem para ser utilizado em dispositivos móveis. Ele conta com uma ontologia que utiliza regras de inferências que auxiliam na adaptação do conteúdo e que utiliza classes para modelar os estudantes, os OAs, as características do dispositivo móvel e o contexto do estudante (por exemplo, se ele está parado ou em movimento).

Ramirez-Arellano, Bory-Reyes e Hernández-Simón (2017) desenvolveram um sistema para ranquear OAs por meio de consultas baseadas em termos. Os OAs melhor ranqueados são agrupados em um pacote do padrão SCORM (ADL, 2001), permitindo o seu compartilhamento entre sistemas de gerenciamento da aprendizagem. Os OAs são intercalados em um mesmo pacote com base nos termos da pesquisa e no estilo de aprendizagem do estudante.

Ouf et al. (2017) implementaram uma ferramenta para um ecossistema de aprendizagem inteligente usando ontologias e regras SWRL. Uma ontologia é utilizada para modelar os estudantes e outras três são implementadas para adequar a eles os componentes do processo de aprendizagem, tais como OAs, métodos de ensino-aprendizagem relevantes e atividades de aprendizagem preferidas.

Pereira et al. (2018) apresentaram uma infraestrutura para a recomendação de recursos educacionais com base em informações, tais como o perfil do usuário e o contexto educacional, extraídas da rede social Facebook. Técnicas de extração de informações e tecnologias da WS são utilizadas para extrair, enriquecer e definir os perfis e interesses dos usuários. A abordagem de recomendação é baseada em repositórios de OAs, dados ligados e repositórios de vídeos, beneficiando-se do tempo gasto pelo usuário na Web.

Todos esses trabalhos fazem uso da Web, seja pelo reúso de conteúdo (o que engloba os repositórios de OAs) e/ou a utilização de tecnologias da WS. Mas nenhum combina

ontologias com a criação de novos recursos educacionais. O diferencial desta dissertação em relação a esses trabalhos é a criação de OAs pelo reúso de conteúdo da Web mais a recomendação desses OAs com o auxílio de uma ontologia. Além disso, utiliza-se o padrão IEEE-LOM combinado com a sua extensão CLEO.

Abordagem Proposta

Para lidar com os desafios desta pesquisa e solucionar o Problema da Recomendação de Objetos de Aprendizagem (PROA), definido na Seção 4.1, apresenta-se a seguir uma abordagem baseada em um AG e em uma ontologia com regras SWRL para a recomendação de OAs que satisfazem os parâmetros de busca do usuário. Essa abordagem define um Sistema de Criação e Recomendação de Objetos de Aprendizagem (SCROA). Criou-se um protótipo do SCROA (sem interface gráfica) para testar e validar a eficiência da abordagem.

O usuário interage com o protótipo do SCROA (ver Figura 5) por meio das tradicionais operações de criação, leitura, atualização e deleção (do inglês CRUD). Essas operações são utilizadas para a manipulação de estudantes e metadados de OAs armazenados na ontologia. Cada estudante modelado na ontologia possui um nome, uma matrícula e um perfil associado ao FSLSM. Em relação aos OAs, há quatro tipos diferentes:

- a) OA ideal: Armazena os parâmetros de busca escolhidos pelo usuário. O OA ideal é assim denominado porque espera-se que os OAs recomendados sejam o mais similar possível a ele. O usuário cria um OA ideal (ou torna ativo um OA ideal já existente) e define alguns parâmetros de busca. Há um parâmetro do OA ideal, o tipo de recurso de aprendizagem, que não é definido pelo usuário, mas sim inferido por regras SWRL usando o perfil do estudante;
- b) OA permanente: Refere-se a um OA, cujos metadados já estavam armazenados na ontologia, porque havia sido criado pelo usuário ou recomendado;
- c) OA sugerido: Corresponde a um OA permanente filtrado por regras de inferência que consideram a similaridade entre o OA permanente e o OA ideal;
- d) OA temporário: É criado (usando uma seção wiki) e armazenado temporariamente na ontologia durante o processo de recomendação. O OA temporário torna-se permanente ou é excluído da ontologia.

O OA ideal pode variar de um contexto para outro. Cada contexto costuma utilizar um subconjunto dos campos definidos pelo IEEE-LOM, e a ontologia do SCROA está pronta

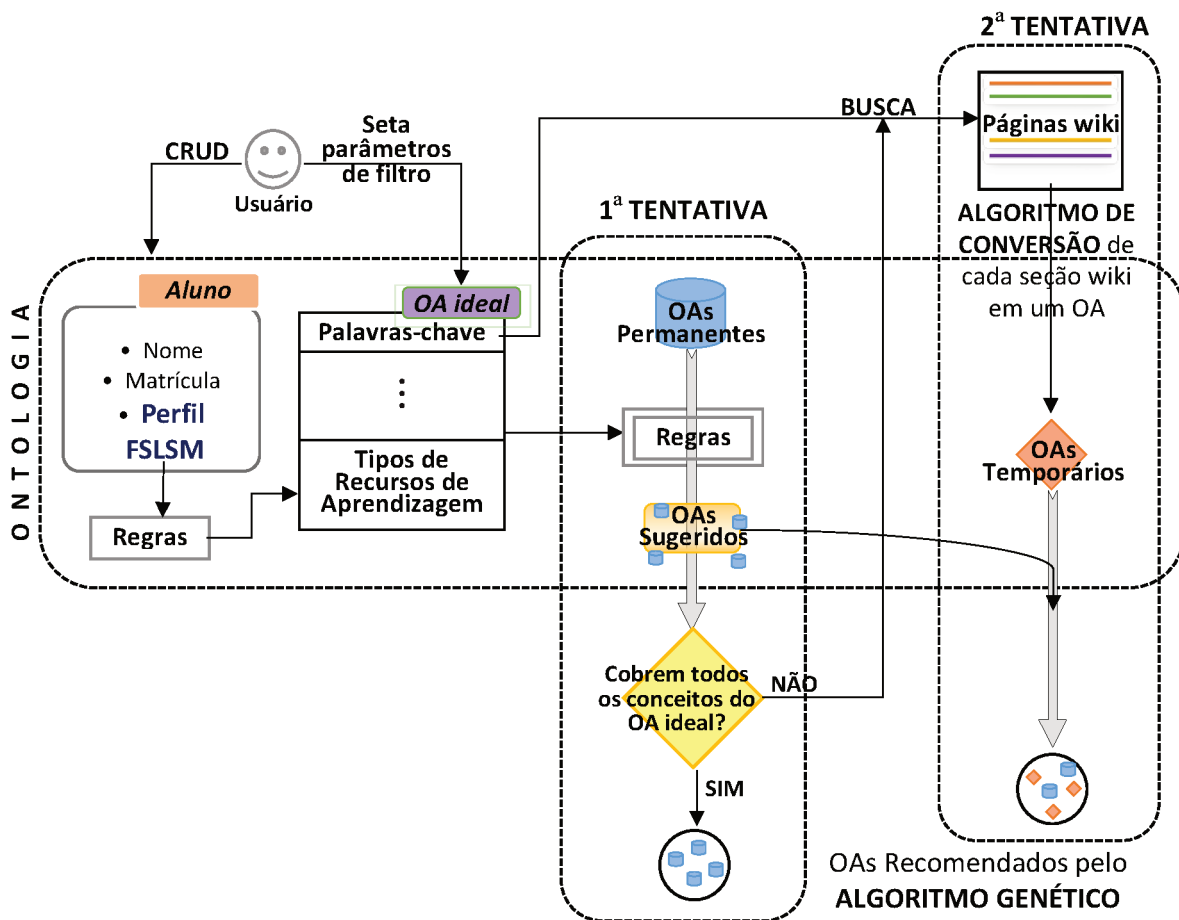


Figura 5 – Modelagem do Sistema de Criação e Recomendação de Objetos de Aprendizagem (SCROA)

para isso, pois implementa todos os campos desse padrão, estando, portanto, aberta às demandas dos ambientes educacionais. Os parâmetros de busca (do OA ideal) disponíveis para manipulação pelo usuário são apresentados a seguir:

1. Título (1.2 IEEE-LOM): Nome dado ao OA;
2. Descrição (1.4 IEEE-LOM): Descrição do conteúdo do OA;
3. Palavras-chave (1.5 IEEE-LOM): Descrevem o tópico do OA;
4. Tipo de interatividade (5.1 IEEE-LOM): Modo de aprendizagem predominante do OA, que pode ser *active*, *expositive* ou *mixed*;
5. Tipo de recurso de aprendizagem (5.2 IEEE-LOM mais extensão CLEO): Corresponde ao(s) tipo(s) predominante(s) do OA, como por exemplo, um exercício;
6. Nível de interatividade (5.3 IEEE-LOM): Representa o grau de interatividade do OA para com o aprendiz, podendo assumir os valores *very low*, *low*, *medium*, *high* ou *very high*;

7. Densidade semântica (5.4 IEEE-LOM): Refere-se à quantidade de conceitos citados por unidade de espaço ou tempo de apresentação do conteúdo, podendo ser *very low*, *low*, *medium*, *high* ou *very high*;
8. Grau de dificuldade (5.8 IEEE-LOM): Está vinculado ao nível de dificuldade do OA para o público-alvo, podendo ser *very easy*, *easy*, *medium*, *difficult* ou *very difficult*.

O usuário não precisa preencher todos esses parâmetros. De fato, como o SCROA trata-se de um sistema de recomendação baseado em conteúdo, é importante que o usuário preencha ao menos as palavras-chave (conceitos), pois o objetivo do AG é a recomendação de OAs que cobrem os conceitos do OA ideal.

Na ontologia do SCROA, não se armazena o conteúdo educacional, mas apenas os seus metadados. Note que os campos do padrão IEEE-LOM utilizados tratam, em sua maioria, de metadados educacionais. Contudo, todos os campos de todas as categorias do IEEE-LOM foram implementados na ontologia. Assim, eles podem ser utilizados conforme a necessidade de cada contexto educacional.

Nesta abordagem, prioriza-se a recomendação dos OAs permanentes da ontologia, sempre que esses sejam suficientes para cobrir todos os conceitos que se espera que o aluno aprenda. Apenas se não houver OAs suficientes para isso, é que se procede com a conversão de conteúdo wiki em OAs que cobrem os conceitos não cobertos pelos OAs permanentes.

Desta maneira, o SCROA realiza até duas tentativas no processo de recomendação de OAs:

- a) Primeira tentativa: Utilizam-se regras SWRL para sugerir os OAs permanentes (persistidos na ontologia) que possuem alguma similaridade com o OA ideal ativo. Então utiliza-se um algoritmo genético para recomendar aqueles OAs, dentre os OAs sugeridos, mais similares ao OA ideal. Se os OAs sugeridos na primeira tentativa forem incapazes de cobrir todos os conceitos do OA ideal, então a primeira tentativa falha (mesmo antes da execução do algoritmo genético). Assim, realiza-se a segunda tentativa.
- b) Segunda tentativa: Busca-se para cada conceito descoberto uma página wiki cujas seções são transformadas em OAs temporários na ontologia. Desse modo, os OAs temporários estão associados a seções wiki que abordam os conceitos (palavras-chave) do OA ideal. Note que os OAs temporários podem cobrir, além dos conceitos descobertos, os conceitos cobertos pelos OAs sugeridos. Dentre os OAs sugeridos e temporários, aqueles mais similares ao OA ideal são recomendados por meio do algoritmo genético. Os OAs recomendados para o estudante são transformados em OAs permanentes caso ainda não tenham esse status.

O protótipo do SCROA foi implementado na linguagem Python. O Python é uma linguagem de programação de alto nível, funcional e orientada a objetos. Ela prioriza

a legibilidade, possuindo uma sintaxe concisa e clara. Além disso, a linguagem Python possui uma biblioteca padrão repleta de recursos e conta com diversos outros módulos disponibilizados por terceiros.

O PROA é apresentado na Seção 4.1. A ontologia do SCROA, com suas respectivas classes, propriedades e regras, é apresentada na Seção 4.2. Na Seção 4.3, apresenta-se um modo de avaliar a qualidade das seções wiki vinculadas aos OAs temporários. O processo de extração dos conceitos que cada seção wiki cobre é apresentado na Seção 4.4. O Algoritmo de Recomendação Principal (ARP) que coordena o SCROA é apresentado na Seção 4.5 e suas etapas são apresentadas na Subseção 4.5.1. O ARP possui dois procedimentos principais: o algoritmo de conversão de seções wiki em OAs utilizado na segunda tentativa de recomendação de OAs (ver Subseção 4.5.2) e o AG utilizado na primeira e segunda tentativa (ver Subseção 4.5.3).

4.1 O Problema da Recomendação de Objetos de Aprendizagem (PROA)

Os trabalhos que utilizam algoritmos aproximados para resolver o PCC consideram a estrutura matriz como instância do PCC. Porém, há uma outra maneira de definir o PCC, baseada em conjuntos, que pode ajudar na compreensão do problema, favorecendo a definição do PROA.

Imagine, por exemplo, uma situação em que um estudante precise aprender cinco conceitos, os quais pertencem ao conjunto finito $X = \{C_1, C_2, C_3, C_4, C_5\}$. Considere uma coleção de subconjuntos de X dada por $F = \{OA_1, OA_2, OA_3, OA_4, OA_5, OA_6\}$, sendo $OA_1 = \{C_1, C_2\}$, $OA_2 = \{C_3\}$, $OA_3 = \{C_4\}$, $OA_4 = \{C_5\}$, $OA_5 = \{C_2, C_3, C_4, C_5\}$ e $OA_6 = \{C_2, C_3\}$. Cada elemento de F é um objeto de aprendizagem que cobre um conjunto de conceitos. O OA_1 , por exemplo, cobre os conceitos C_1 e C_2 . O objetivo é encontrar o menor número de OAs que juntos cobrem todos os elementos (conceitos) de X . Esse é um problema de Programação Linear Inteira conhecido na literatura como o Problema do Recobrimento (PR) (GOLDBARG; LUNA, 2005) e de difícil solução, pertencendo à classe NP-Hard (GAREY; JOHNSON, 1979). O PR é equivalente ao Problema da Cobertura de Conjuntos de custo único.

Para o exemplo anterior, tem-se a solução $\{OA_1, OA_5\}$. Note que o conceito C_2 é coberto por mais de um objeto de aprendizagem, quais sejam, OA_1 e OA_5 . Para resolver esse problema da redundância, basta acrescentar ao PR a seguinte restrição: a interseção dos elementos (tomados dois a dois) do conjunto solução deve ser vazia. Nesse caso, tem-se o Problema do Particionamento (PP) (GOLDBARG; LUNA, 2005). A única solução do PP para o mesmo exemplo é a partição $\{OA_1, OA_2, OA_3, OA_4\}$, que garante que cada conceito de X seja coberto exatamente uma vez com o menor número de OAs possível.

Esse exemplo foi cuidadosamente criado para mostrar um impasse que existe entre os dois problemas. Enquanto que a solução do PR garante um número de OAs igual ou menor que o número de OAs da solução do PP, esta, por sua vez, garante uma partição. Vale ressaltar que o PCC não garante a minimização do número de OAs (colunas) como o PR e não tem o objetivo de cobrir cada conceito (cada linha) de maneira única como o PP. Contudo, os três problemas têm um objetivo em comum: cobrir todas as linhas (conceitos).

Devido à natureza em comum desses problemas, um algoritmo aproximado que resolve o PCC, embora não garanta a minimização do número de colunas, contribui para a sua redução (lembre-se que cada coluna possui um custo maior que zero). Além disso, cada coluna de uma solução para o PCC cobre ao menos um conceito de maneira única, ou seja, nenhuma coluna é inútil ou redundante. Note que essa característica contribui também na redução do número de redundâncias de linhas.

No contexto deste trabalho, o PROA é definido como o PCC. Porém é conveniente um pré-processamento das instâncias do PROA antes da sua solução como um problema de cobertura de conjuntos. Na prática, pode haver OAs (colunas) que não cobrem nenhum dos conceitos (linhas) necessários. Colunas desse tipo podem ser eliminadas das instâncias.

Outro aspecto importante é o cálculo do c_j da Eq. (1). No contexto deste trabalho, o c_j corresponde ao custo de escolha do objeto de aprendizagem j . O custo de escolha de um OA_j depende de quão próximo ele é do objeto de aprendizagem ideal (OA_{ideal}). O OA_{ideal} é criado com os parâmetros de busca definidos pelo usuário e os OAs recomendados são os mais similares ao OA_{ideal} . O valor de um dado parâmetro do OA_{ideal} é comparado com os valores (desse mesmo parâmetro) dos OAs disponíveis para recomendação.

Formalmente, seja α_i o valor do parâmetro i . A função c_j da Eq. (1) é calculada pela comparação dos valores, de cada parâmetro, correspondentes ao OA_{ideal} e a um OA_j :

$$c_j = \sum_{i=1}^k (|\alpha_{i(ideal)} - \alpha_{i(j)}|) \quad (12)$$

onde k é o número de parâmetros, $\alpha_{i(ideal)}$ é um parâmetro do OA_{ideal} e $\alpha_{i(j)}$ é um parâmetro do OA_j .

O objetivo do PROA é a recomendação de OAs que cobrem os conceitos do OA_{ideal} e satisfaçam os parâmetros do OA_{ideal} .

4.2 Ontologia

A ontologia do SCROA foi construída conforme a metodologia *Ontology Development 101* (NOY; MCGUINNESS, 2001), utilizando-se a ferramenta Protégé e a linguagem OWL. As classes e subclasses da ontologia são apresentadas na Figura 6.

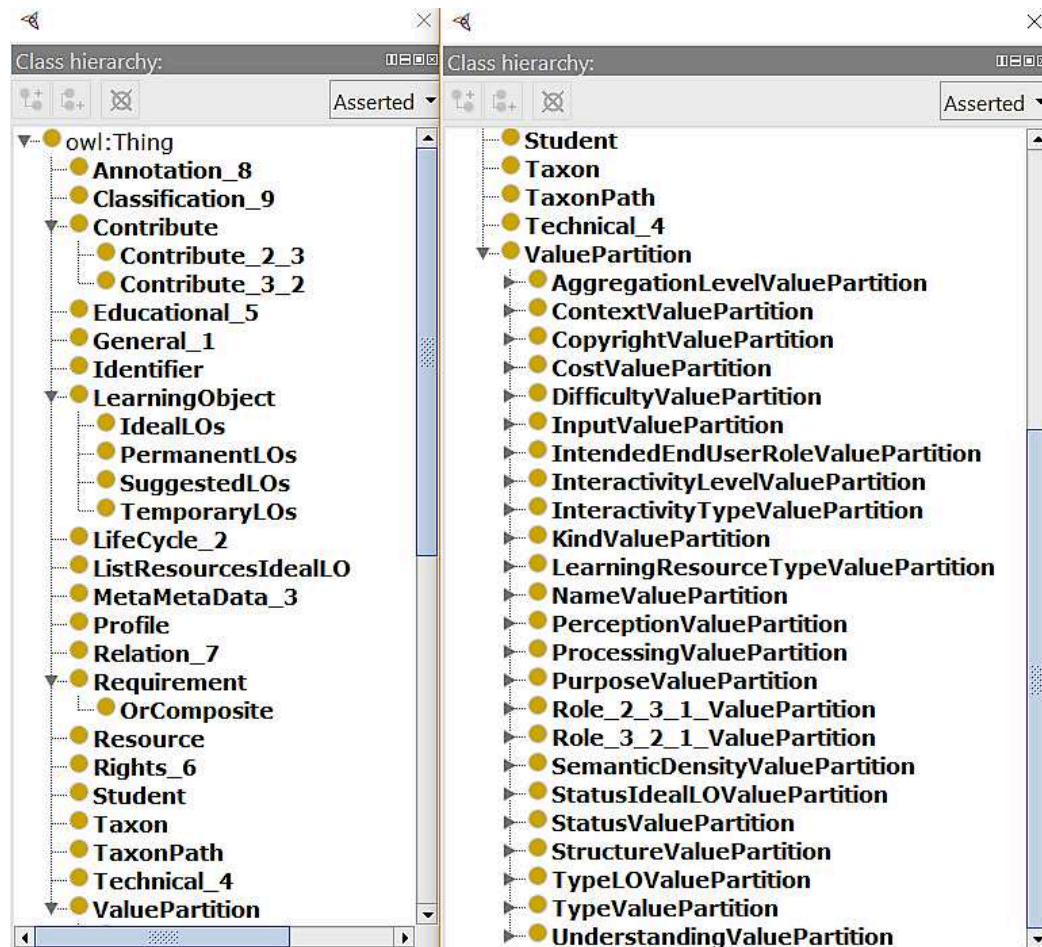


Figura 6 – Classes e subclasses da ontologia

Fonte: Elaborada pelo autor com a ferramenta Protégé

As classes *LearningObject* e *Student* agregam, respectivamente, instâncias de objetos de aprendizagem e instâncias de estudantes. O estudante tem um perfil (instância da classe *Profile*) que corresponde ao FSLSM. Cada objeto de aprendizagem tem as nove categorias do padrão IEEE-LOM, as quais são representadas na ontologia pelas classes *General_1*, *LifeCycle_2*, *MetaMetaData_3*, *Technical_4*, *Educational_5*, *Rights_6*, *Relation_7*, *Annotation_8* e *Classification_9*.

As classes *Contribute*, *Contribute_2_3*, *Contribute_3_2*, *Identifier*, *Requirement*, *OrComposite*, *Resource*, *Taxon* e *TaxonPath*, associadas ao padrão IEEE-LOM, são todas *range* (de tipo não primitivo) de alguma propriedade.

As propriedades, cujo *range* é um conjunto de valores fixos, tais como a propriedade *hasDifficulty* que possui o *range* *VeryEasy*, *Easy*, *Medium*, *Difficult*, *VeryDifficult*, foram implementadas usando o padrão *Value Partition* (ver Seção 2.4). Todas as classes que seguem esse padrão possuem a terminação “*ValuePartition*” (ver Figura 6) e suas subclasses correspondem aos valores fixos.

A biblioteca Owlready2¹ é um módulo do Python que facilita a manipulação de ontologias, carregando-as como objetos do Python. Essa biblioteca permite importar, editar e salvar ontologias. O usuário pode criar classes, propriedades e instâncias, inclusive definir restrições de propriedades. É possível também criar novas ontologias e fazer inferências.

4.2.1 Modelagem dos objetos de aprendizagem

Os OAs foram modelados conforme o padrão IEEE-LOM. Para diferenciar os diversos tipos de OAs da ontologia, criaram-se quatro classes diferentes de OAs, todas subclasses de LearningObject. As 10 propriedades da classe LearningObject são apresentadas na Figura 7.

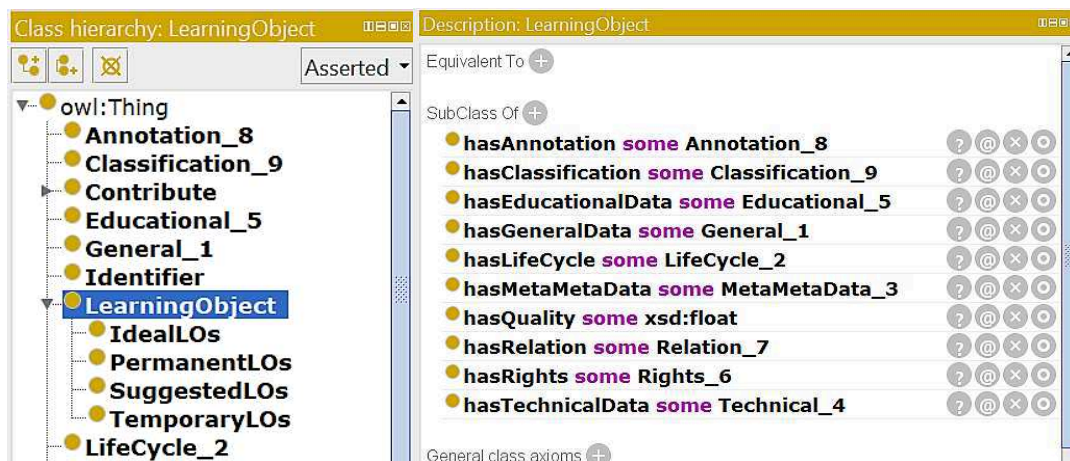


Figura 7 – As subclasses e propriedades da classe LearningObject

Fonte: Elaborada pelo autor com a ferramenta Protégé

Essas propriedades da classe LearningObject são herdadas pelas suas subclasses, quais sejam:

- a) IdealLOs - cada instância dessa classe corresponde a um OA ideal que é recomendado para um dado estudante. Apenas um OA ideal (com estado ativo) é considerado no processo de recomendação. Além dos 10 atributos herdados da classe LearningObject, a classe IdealLOs possui mais duas propriedades:
 - isRecommendedFor: cada instância de IdealLOs é recomendada para uma instância da classe Student;
 - hasState: todos os OAs ideais têm estado inativo, exceto o OA ideal utilizado no processo de recomendação, o qual tem estado ativo.
- b) PermanentLOs - é a classe que agrega as instâncias de OAs armazenadas permanentemente na ontologia.

¹ Disponível em: <https://pythonhosted.org/Owlready2/>

- c) SuggestedLOs - contém as instâncias de OAs permanentes que possuem alguma similaridade ao OA ideal ativo. Essas instâncias são OAs sugeridos por regras de inferência.
- d) TemporaryLOs - armazena temporariamente os OAs criados com conteúdo wiki para serem disponibilizados no processo de recomendação.

Utilizou-se o vocabulário alternativo do CLEO para os tipos de recursos de aprendizagem. A Figura 8 apresenta os tipos de recursos de aprendizagem dos vocabulários IEEE-LOM e CLEO implementados na ontologia.

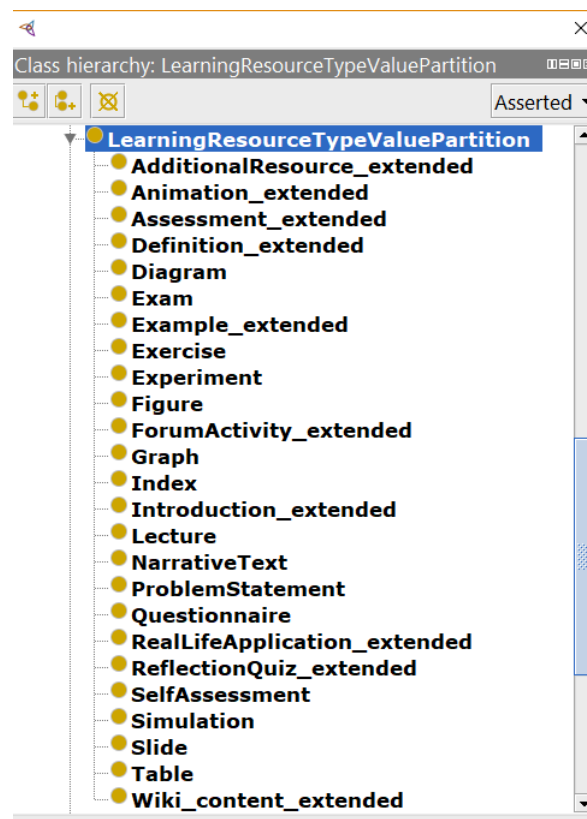


Figura 8 – Os tipos de recursos de aprendizagem dos padrões IEEE-LOM e CLEO

Fonte: Elaborada pelo autor com a ferramenta Protégé

Os termos que não são do padrão IEEE-LOM terminam com a expressão “_extended” (ver Figura 8) para diferenciá-los do vocabulário do IEEE-LOM. Implementaram-se os 15 tipos definidos pelo IEEE-LOM, mais 5 tipos estendidos pelo CLEO, que são *additional resource*, *assessment*, *definition*, *example* e *introduction*, e mais 4 tipos derivados do trabalho de Graf, Kinshuk e Ives (2010):

1. *Animation*: Multimídia animada utilizada na demonstração dos conceitos do curso;
2. *Forum activity*: Atividades que encorajam os alunos a fazerem perguntas e interagir com seus colegas e o professor em tópicos de discussão;

3. *Real life application*: Demonstra como o conhecimento aprendido pode ser aplicado em situações da vida real;
4. *Reflection quiz*: Incluem perguntas sobre o conteúdo de uma seção, encorajando os aprendizes a refletirem sobre o conteúdo aprendido.

Esses tipos escolhidos são os mais comumente usados na prática pedagógica. Além deles, criou-se mais um tipo, denominado *wiki content*, utilizado para representar as seções wiki. Os tipos dos padrões IEEE-LOM e CLEO são explicados na fundamentação teórica (Seção 2.6). Embora nem todos os tipos do CLEO tenham sido implementados, a ontologia é aberta para a implementação de outros tipos, inclusive de outros padrões, conforme a necessidade de cada contexto educacional.

4.2.2 Modelagem dos estudantes

No contexto deste trabalho, optou-se por uma modelagem simplificada do estudante, considerando um dos seus principais atributos, seu perfil, além das propriedades nome e número de matrícula, importantes para diferenciar um estudante dos outros. Contudo, a ontologia é aberta para ser enriquecida com novos atributos que cada contexto educacional necessitar. A Figura 9 apresenta os atributos da classe Student e da classe Profile.

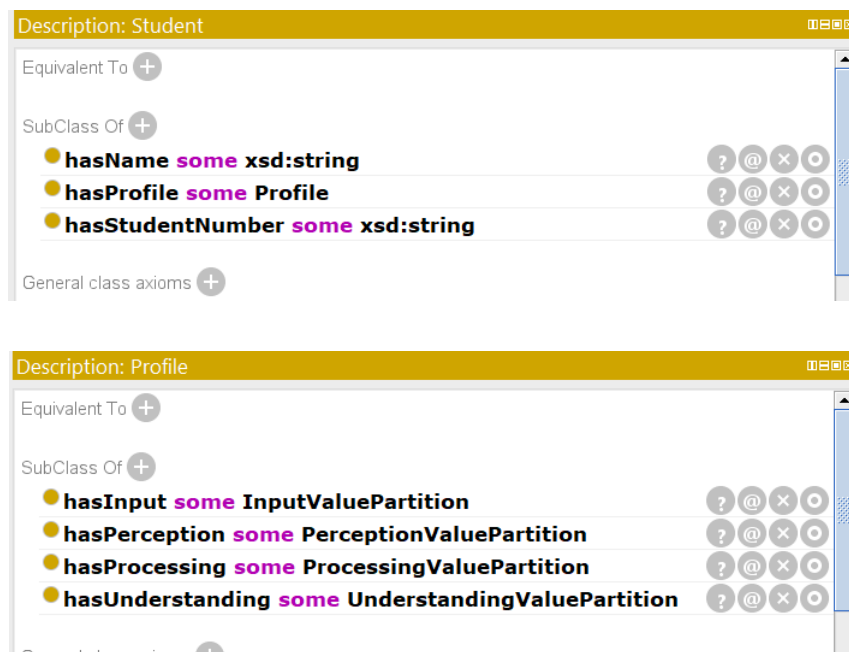


Figura 9 – Atributos das classes Student e Profile

Fonte: Elaborada pelo autor com a ferramenta Protégé

O perfil do estudante definido pela classe Profile segue o modelo FSLSM (ver Figura 9). As dimensões desse modelo, implementadas pelo padrão Value Partition, são detalhadas na Figura 10.

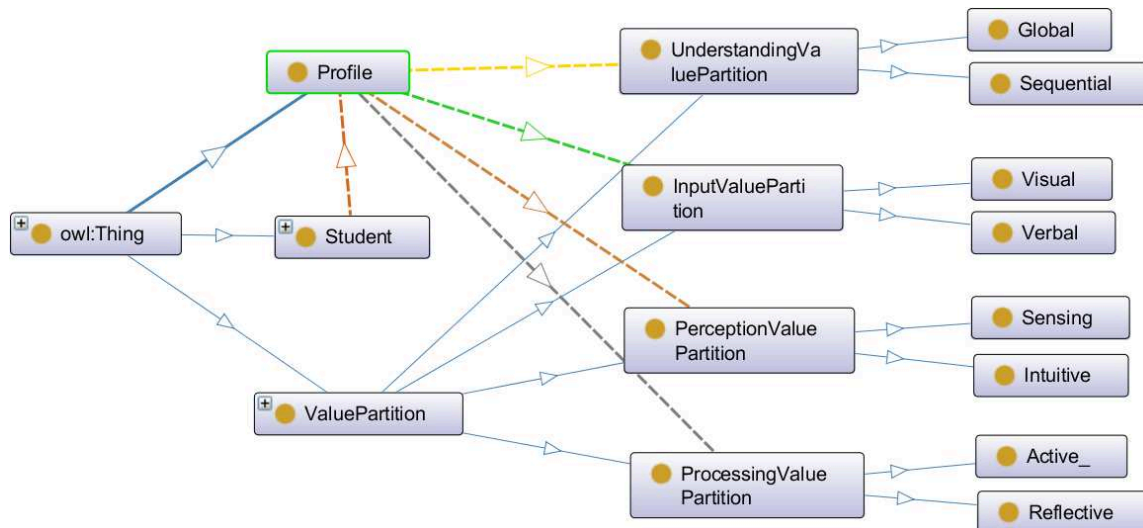


Figura 10 – Modelagem do perfil do estudante conforme o FSLSM

Fonte: Elaborada pelo autor com o *plugin* OntoGraf no Protégé

O questionário ILS (Index of Learning Styles) pode ser aplicado pelo professor para a identificação dos estilos de aprendizagem do estudante.

4.2.3 Regras SWRL

As regras implementadas na ontologia possuem duas diferentes finalidades. Algumas regras são utilizadas na inferência dos tipos de recursos de aprendizagem mais adequados para um dado perfil de estudante. Outras regras são usadas na sugestão de OAs que podem ser recomendados. Essas duas situações podem ser identificadas facilmente na Figura 5. O processo de inferência é executado pelo *reasoner* HermiT, que é acoplado à biblioteca Owlready2.

Na Tabela 3, apresentam-se os tipos de recursos de aprendizagem que devem ser recomendados para diferentes tipos de perfis de estudantes. Esta tabela foi construída com base na teoria de Graf, Kinshuk e Ives (2010) que aborda quais tipos de OAs devem ser recomendados para cada tipo de perfil de estudante associado ao FSLSM.

Todas as 32 regras implementadas na ontologia estão no Apêndice A. As regras de número 25 a 31 são utilizadas para inferir os OAs que possuem alguma similaridade em relação ao OA ideal ativo na ontologia. Esses OAs são inferidos como instâncias da classe SuggestedLOs e disponibilizados no processo de recomendação. Essas regras são importantes para filtrar aqueles OAs que possuem alguma chance de serem recomendados,

Tabela 3 – Os tipos de recursos recomendados para cada perfil de estudante

Regra	Estudante		OAs
	Dimensão	Perfil	Tipo de Recurso de Aprendizagem
1	entrada	verbal	additionalReading
2	entrada	verbal	forumActivity
3	entrada	visual	animation
4	organização	global	example
5	organização	global	realLifeApplication
6	organização	sequencial	additionalReading
7	organização	sequencial	animation
8	organização	sequencial	exercise
9	organização	sequencial	reflectionQuiz
10	organização	sequencial	selfAssessment
11	percepção	intuitivo	additionalReading
12	percepção	intuitivo	exercise
13	percepção	intuitivo	reflectionQuiz
14	percepção	sensitivo	animation
15	percepção	sensitivo	example
16	percepção	sensitivo	exercise
17	percepção	sensitivo	realLifeApplication
18	processamento	ativo	animation
19	processamento	ativo	exercise
20	processamento	ativo	forumActivity
21	processamento	ativo	selfAssessment
22	processamento	reflexivo	additionalReading
23	processamento	reflexivo	example
24	processamento	reflexivo	reflectionQuiz

separando-os dos demais OAs da ontologia. Dentre os OAs persistidos (instâncias da classe PermanentLOs) na ontologia, apenas os sugeridos (instâncias da classe SuggestedLOs) pelas regras de inferência podem ser recomendados.

Se um OA temporário já estiver persistido na ontologia, ou seja, for instância da classe PermanentLOs, então o OA temporário redundante é eliminado e o OA permanente correspondente é inferido como instância da classe SuggestedLOs por meio da regra 32. Esse processo de inferência é importante para garantir que a versão permanente do OA temporário excluído tenha chance de ser recomendada. Vale lembrar que os OAs recomendados devem ser, necessariamente, instâncias da classe SuggestedLOs ou TemporaryLOs.

4.3 Avaliação da qualidade de seções wiki

Para avaliar a qualidade das seções wiki, utiliza-se o classificador proposto por Warncke-Wang, Cosley e Riedl (2013). A terceira coluna da Tabela 4 corresponde a uma das adaptações necessárias para a utilização do classificador no contexto do SCROA. Essa adaptação, que mapeia cada classe em um valor numérico, permite a utilização da Eq. (12).

As classes na tabela estão ordenadas pelo seu padrão de qualidade. Os artigos com

o mais alto nível de qualidade (FA) recebem o valor máximo igual a 1 e os artigos sem nenhuma qualidade recebem nota 0. A qualidade dos artigos wiki corresponde a um dos k parâmetros definidos pela Eq. (12). Seja α_q esse parâmetro de qualidade e, para utilizá-lo na Eq. (12), definem-se a seguir o $\alpha_{q_{ideal}}$ e o α_{q_j} do j -ésimo OA comparado com o OA ideal:

- a) $\alpha_{q_{ideal}} = 1$ (O OA ideal, certamente, possui o mais alto nível de qualidade, pertencendo à classe FA);
- b) α_{q_j} é o valor correspondente à classificação de qualidade recebida pelo j -ésimo OA conforme a Tabela 4.

Tabela 4 – Classes de qualidade da Wikipédia

Classe	Critérios	Valor
(FA) Featured Articles	Contém os artigos de maior qualidade, eles são bem escritos, compreensíveis, neutros (sem opiniões), possuem fontes confiáveis, são estáveis (o conteúdo não precisa de alterações constantes), seguem uma boa estrutura (com resumo, divisão em seções e citações bem formatadas), possuem imagens apropriadas e tamanho adequado (foca no tópico principal sem conteúdo desnecessário).	1
A	O artigo é bem organizado e essencialmente completo. Apenas pequenos problemas de estilo e outros detalhes precisam ser corrigidos antes da submissão como um artigo FA.	0.85
(GA) Good Articles	O artigo requer pequenas alterações para atingir um melhor nível de qualidade.	0.70
B	O artigo está quase completo e sem grandes problemas, mas requer alterações adicionais para alcançar o padrão de bom artigo (GA).	0.50
C	O artigo é substancial, mas carece de conteúdo importante ou contém material irrelevante. O artigo cita mais do que uma fonte confiável e é mais desenvolvido em estilo, estrutura e qualidade do que a classe Start, mas falha em um ou mais dos critérios da classe B.	0.30
Start	É um artigo em construção e bem incompleto, mas possui uma quantidade significativa de bom conteúdo. Não segue o Manual de Estilo da Wikipédia. Pode ou não citar fontes confiáveis e adequadas.	0.15
Stub	O artigo possui uma descrição muito básica do tópico, e todos os artigos de qualidade muito ruim pertencem a essa classe.	0

A solução do problema original é a recomendação de um grupo de OAs com o menor número de diferenças em relação ao OA ideal. Assim, quanto maior for a qualidade de um OA, maior as chances dele ser recomendado, pois menor é a diferença entre ele e o OA ideal.

4.4 Identificação dos conceitos de seções wiki

O retorno de uma busca na Wikipédia são páginas wiki, cujas seções são candidatas a objetos de aprendizagem. Para que as seções dessas páginas se tornem de fato OAs, é necessário que elas tratem dos conceitos escolhidos pelo usuário, ou seja, os conceitos que

se espera que o estudante aprenda. Na segunda tentativa de recomendação (ver Figura 5), deve-se primeiramente fazer a extração de conceitos das seções wiki para confrontá-los com os conceitos inseridos pelo usuário no OA_{ideal} .

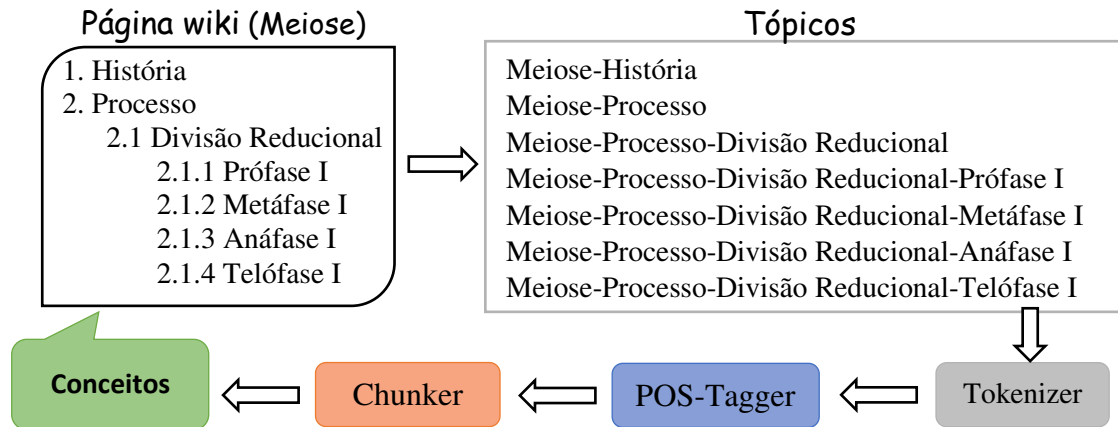


Figura 11 – Processamento de um sumário da Wikipédia para a geração de conceitos

O método utilizado para a extração dos conceitos das seções wiki (ver Figura 11) baseia-se na intuição de que o próprio título da seção mais a hierarquia de títulos na qual ela está inserida possam ser usados para a extração dos conceitos que ela trata. Neste caso, após apontar a classe gramatical (POS-tagging) de cada palavra encontrada nessa hierarquia de títulos pelo *tokenizer*, utiliza-se o *chunker* para formar cada Noun Phrase (NP). Uma NP pode ser qualquer tipo de substantivo, inclusive substantivos intercalados por preposições e caracterizados por adjetivos. Embora alguns substantivos indesejáveis possam ser selecionados como conceitos, quase sempre os títulos das seções contêm os principais conceitos relacionados a elas.

A cada seção wiki é atribuído um tópico formado pelo título da seção acrescido dos seus títulos hierarquicamente superiores até o título da página wiki. Cada tópico é utilizado para a extração dos conceitos da sua respectiva seção. Note que algumas seções (como a seção “2. Processo”), podem ter ou não algum conteúdo antes da sua primeira subseção. As seções com quase nenhum conteúdo são ignoradas. Sabe-se que o tempo gasto com as técnicas de Processamento de Linguagem Natural (PLN) tende a aumentar, proporcionalmente, em relação ao tamanho da entrada. A vantagem do método apresentado na Figura 11 é que ele evita o processamento do conteúdo das seções wiki, tornando-se mais rápido.

4.4.1 Extração de conteúdo da Wikipédia com a Wikipedia API

A Wikipedia API² é um módulo do Python que dá suporte à extração de seções, títulos, links e categorias da Wikipédia. De fato, é possível extrair praticamente todas

² Disponível em: <https://pypi.org/project/Wikipedia-API/>

as partes de uma página wiki utilizando comandos simples. Essa API pode ser usada também na busca por páginas wiki, usando para isso o método *page*, que espera como parâmetro o título da página procurada.

No escopo deste trabalho, a Wikipedia API foi utilizada principalmente para:

- a) A busca pela página wiki associada a cada conceito do OA ideal não coberto pelo OAs permanentes da ontologia;
- b) A extração dos títulos das seções wiki para o preenchimento do campo “Título (1.2 IEEE-LOM)” e do campo “Entrada (1.1.2 IEEE-LOM)”, o qual corresponde ao URI da seção wiki.

4.4.2 O Processamento de Linguagem Natural com as bibliotecas spaCy e NLTK

O objeto de estudo da área de PLN é a compreensão da linguagem humana. O spaCy³ é um software de código aberto que facilita a manipulação da linguagem natural no Python. Embora o spaCy seja uma biblioteca completa, o programador pode optar em utilizar também a biblioteca NLTK⁴, a qual agrega métodos intuitivos para o tratamento da linguagem natural e é amplamente usada para ensino e pesquisa. Ao contrário do NLTK, o spaCy foca na produção de novos programas para o mercado.

No contexto deste trabalho, utilizam-se essas ferramentas no processo de identificação dos conceitos das seções wiki. O spaCy é utilizado nas etapas Tokenizer e POS-Tagger. A biblioteca NLTK é utilizada na etapa Chunker.

4.5 Algoritmo de Recomendação Principal (ARP)

O processo de recomendação de objetos de aprendizagem é auxiliado por regras implementadas em SWRL na ontologia. As regras de inferência são apoiadas por um AG utilizado na primeira ou na segunda tentativa de recomendação de OAs.

É difícil dizer até que ponto regras SWRL podem auxiliar na resolução de problemas de otimização e como elas seriam capazes de avaliar se uma solução é melhor que outra. A tentativa de resolver problemas dessa natureza com regras de inferência, provavelmente, levaria a um conjunto infundável de regras grandes e complexas que não resolveriam esses problemas de maneira eficiente.

Contudo, regras SWRL são especialmente boas para avaliar se determinadas instâncias satisfazem propriedades. Assim, aquelas instâncias de OAs (quase) idênticas ao OA ideal ativo (ou seja, com a mesma densidade semântica, mesmo grau de dificuldade, etc.) podem ser sugeridas por meio de regras de inferência. O Algoritmo de Recomendação Principal

³ Disponível em: <https://spacy.io/>

⁴ Disponível em: <https://www.nltk.org/>

(ARP), cujas etapas são apresentadas na Seção 4.5.1, utiliza regras de inferência que filtram os OAs com reais chances de serem recomendados. Os OAs sugeridos pelo processo de inferência são utilizados como entrada do AG que resolve o PROA.

Nesta primeira tentativa de recomendação, os OAs sugeridos pelas regras de inferência podem não cobrir todos os conceitos do OA ideal e, neste caso, o AG não é executado, pois o PROA não pode ser resolvido, devido à falta de OAs que cubram todos os conceitos do OA ideal. Para contornar esse problema, realiza-se uma segunda tentativa de recomendação, na qual, seções wiki que cobrem os conceitos do OA ideal são convertidas em OAs (ver Subseção 4.5.2), possibilitando, assim, a execução do AG que resolve o PROA (ver Subseção 4.5.3). Note que esse AG e o algoritmo de conversão de seções wiki em OAs são exibidos na Figura 5, que modela o SCROA.

4.5.1 Etapas do ARP

O ARP segue as etapas a seguir.

1. Selecione um OA ideal (OA_{ideal}) da ontologia.
2. Torne o OA_{ideal} selecionado como ativo na ontologia.
3. Selecione um estudante (*aluno*) da ontologia.
4. Crie na ontologia a relação: $OA_{ideal} isRecommendedFor aluno$.
5. Execute o processo de inferência.
6. Crie uma lista de OAs (*lista_oas*) com os OAs sugeridos pela inferência.
7. Se há nenhum conceito no OA_{ideal} , então finalize executando o procedimento que compara os parâmetros do OA_{ideal} com os parâmetros dos OAs de *lista_oas* para retornar os OAs mais similares ao OA_{ideal} . Caso contrário (se há conceitos), vá para o passo 8.
8. Se os OAs de *lista_oas* forem suficientes para cobrir todos os conceitos do OA_{ideal} , então vá para o passo 14. Caso contrário, vá para o passo 9.
9. Execute o método de criação de OAs com conteúdo de seções wiki.
10. Torne os OAs criados, no passo 9, instâncias da classe *TemporaryLOs*.
11. Execute o processo de inferência.
12. Delete de *TemporaryLOs* apenas as instâncias que já estão permanentes na ontologia. Essas instâncias permanentes (repetidas nos OAs temporários) são inferidas, no passo 11, como OAs sugeridos.
13. Atribua à variável *lista_oas*, os OAs resultantes da união das instâncias de *SuggestedLOs* com as instâncias de *TemporaryLOs*.
14. Execute o procedimento que gera uma matriz m e o vetor de custos, sendo m uma instância do PROA.
15. Execute o AG que resolve o PROA.

16. Persista na ontologia os OAs temporários recomendados no passo 15, se houver.

Para facilitar a explicação do ARP, parte-se de uma configuração inicial em que alguns OAs (seus metadados) estão armazenados na ontologia, ou seja, são instâncias da classe *PermanentLOs* e ao menos um OA ideal e um estudante (os quais podem ser criados facilmente com a ferramenta Protégé) também estão persistidos na ontologia. Primeiramente, o usuário seleciona um OA ideal (linha 1), o qual é definido como ativo na ontologia, e todos os demais OAs ideais tornam-se inativos (linha 2). O usuário deve selecionar o aluno (linha 3) para o qual será feita a recomendação. Assim, na linha 4, o OA ideal ativo é atribuído ao estudante selecionado por meio da relação *isRecommendedFor*. Essa propriedade é utilizada apenas para relacionar um dado OA ideal com um dado estudante. Os OAs que são recomendados efetivamente para o estudante não possuem a propriedade *isRecommendedFor*, mas são instâncias da classe *SuggestedLOs* ou *TemporaryLOs*.

Os OAs sugeridos já estão persistidos permanentemente na ontologia, eles são instâncias da classe *PermanentLOs*, enquanto que apenas os OAs temporários recomendados se tornarão instâncias de *PermanentLOs*. Esse processo garante que a ontologia seja enriquecida apenas com metadados de conteúdos interessantes para o usuário. Sendo assim, os campos da categoria “Anotação” do padrão IEEE-LOM podem ser usados pelo aluno ou professor para avaliar o OA utilizado, e essa informação pode ser usada para sofisticar o processo de recomendação.

Após a inferência (linha 5), os OAs sugeridos, pelas regras de inferência de número 25 a 31 do Apêndice A, são armazenados em uma lista de OAs (linha 6). O processo de recomendação de OAs é mais simples quando não há nenhum conceito a ser coberto. Nesse caso, o OA_{ideal} não possui nenhum conceito (nenhuma palavra-chave) e a recomendação de OAs, dentre os disponíveis na classe *SuggestedLOs*, é realizada por um procedimento (linha 7) que compara os campos do OA_{ideal} com os campos dos OAs armazenados em *lista_oas*.

Esse procedimento de comparação também é utilizado pelo AG que resolve o PROA definido na Seção 4.1. Assim, cada campo do OA_{ideal} corresponde a um parâmetro α_i da Eq. (12) que calcula a dissimilaridade entre o OA ideal e os demais OAs. Dessa equação, define-se a equação que calcula a similaridade s_j entre o OA_{ideal} e um OA_j :

$$s_j = \sum_{i=1}^k (1 - |\alpha_{i(ideal)} - \alpha_{i(j)}|) \quad (13)$$

onde k é o número de parâmetros, $\alpha_{i(ideal)}$ é um parâmetro do OA_{ideal} e $\alpha_{i(j)}$ é um parâmetro do OA_j .

A maneira como a diferença entre $\alpha_{i(ideal)}$ e $\alpha_{i(j)}$ é calculada depende de cada tipo de parâmetro, conforme o esquema a seguir:

- a) Título (1.2 IEEE-LOM) - O cálculo do s_j de duas strings t_1 e t_2 é realizado pela medida de similaridade do cosseno. No caso do parâmetro título, calcula-se a

- similaridade cosseno entre o título do OA ideal (string t_1) e o título de um OA_j (string t_2);
- b) Descrição (1.4 IEEE-LOM) - Segue o mesmo processo de cálculo do parâmetro título, onde t_1 corresponde ao valor do campo descrição do OA_{ideal} e t_2 ao valor da descrição de um OA_j ;
 - c) Palavras-chave (1.5 IEEE-LOM) - Segue o mesmo processo de cálculo do parâmetro título, sendo os conceitos do OA_{ideal} a string t_1 e os conceitos do OA_j a string t_2 ;
 - d) Tipo de interatividade (5.1 IEEE-LOM) - A cada termo do vocabulário do campo é atribuído um valor de 0 a 1 de maneira gradativa. Assim, para o campo tipo de interatividade tem-se *active* = 0, *expositive* = 0.5 e *mixed* = 1. Por exemplo, seja α_i o parâmetro tipo de interatividade. Dado um OA_{ideal} com valor *expositive* ($\alpha_{i(ideal)} = 0.5$) e um dado OA_j com valor *mixed* ($\alpha_{i(j)} = 1$), então α_i é calculado pela Eq. (13) como $1 - |\alpha_{i(ideal)} - \alpha_{i(j)}|$ que é igual a 0.5;
 - e) Tipo de recurso de aprendizagem (5.2 IEEE-LOM mais extensão CLEO) - Segue o mesmo processo de cálculo do parâmetro título, sendo a lista de recursos de aprendizagem do OA_{ideal} a string t_1 e os recursos de aprendizagem do OA_j a string t_2 . O padrão IEEE-LOM permite que seja associado a um OA até 10 tipos diferentes de recursos de aprendizagem;
 - f) Nível de interatividade (5.3 IEEE-LOM) - Segue o mesmo processo de cálculo do parâmetro tipo de interatividade. Os valores atribuídos aos termos do vocabulário são *very low* = 0, *low* = 0.25, *medium* = 0.5, *high* = 0.75 e *very high* = 1;
 - g) Densidade semântica (5.4 IEEE-LOM) - Segue o mesmo processo de cálculo do parâmetro tipo de interatividade. Os valores atribuídos aos termos do vocabulário são *very low* = 0, *low* = 0.25, *medium* = 0.5, *high* = 0.75 e *very high* = 1;
 - h) Grau de dificuldade (5.8 IEEE-LOM) - Segue o mesmo processo de cálculo do parâmetro tipo de interatividade. Os valores atribuídos aos termos do vocabulário são *very easy* = 0, *easy* = 0.25, *medium* = 0.5, *difficult* = 0.75 e *very difficult* = 1;
 - i) Qualidade - A qualidade de um dado OA_j , por exemplo uma seção wiki, varia de 0 a 1. O parâmetro de qualidade do OA_{ideal} é fixado em $\alpha_{i(ideal)} = 1$ e a qualidade de OA_j é calculada pelo classificador apresentado na Seção 4.3.

Note que a similaridade (s_j) entre o OA_{ideal} e um OA_j corresponde a um valor do intervalo de $[0, k]$, onde k é o número de parâmetros. O procedimento que implementa o esquema anterior divide s_j por k para que o valor da similaridade varie no intervalo de $[0, 1]$. O OA ideal implementado em Python possui uma variável de ponderação para cada parâmetro, permitindo, assim, que alguns parâmetros tenham um peso maior que outros.

Por outro lado, se houver conceitos no OA_{ideal} para serem cobertos (linha 8), então tem-se o Problema da Recomendação de Objetos de Aprendizagem (PROA) definido na

Seção 4.1. Como nem sempre a ontologia dispõe de OAs suficientes para cobrirem todos os conceitos do OA_{ideal} , faz-se necessário a utilização de alguns procedimentos (linhas 9-13) para a criação de novos OAs derivados da Wikipédia com o objetivo de cobrir todos os conceitos do OA_{ideal} .

O método de criação de OAs com conteúdo de seções wiki é detalhado na Seção 4.5.2. Os OAs criados tornam-se instâncias da classe *TemporaryLOs* (linha 10). Após a execução do mecanismo de inferência (linha 11), é necessário deletar de *TemporaryLOs* os OAs temporários que já estão persistidos na ontologia (linha 12). Cada instância de *PermanentLOs* correspondente a um dos OAs temporários deletados torna-se instância da classe *SuggestedLOs*. É por meio da regra 32 (Apêndice A) que os OAs permanentes repetidos na classe de OAs temporários tornam-se instâncias da classe de OAs sugeridos.

Na linha 13, as instâncias resultantes da união da classe *SuggestedLOs* com a classe *TemporaryLOs* são armazenadas na variável *lista_oas*. Esses OAs são exatamente os OAs disponíveis para recomendação. O OA_{ideal} e os OAs sugeridos e/ou temporários de *lista_oas* são utilizados na construção de uma instância do PROA (linha 14). O método de construção dessa instância é explicado na Subseção 4.5.3.1.

A matriz m e o vetor de custos, criados na linha 14, são entradas do algoritmo genético da linha 15, o qual é detalhado na Subseção 4.5.3. Finalmente, os OAs temporários recomendados são persistidos na ontologia.

4.5.2 Algoritmo de conversão de seções wiki em objetos de aprendizagem

Nesta seção, apresenta-se o método de criação de OAs usando conteúdo de seções wiki. Esse método é executado na linha 9 do ARP.

Pseudocódigo `wiki_pages_2_LOs`

Entrada: `concepts_list` #lista de conceitos do OA ideal

Saída: lista de OAs

1. `wiki_pages, lo_list` \leftarrow [] #inicializa variáveis como uma lista vazia
2. `instance_number` \leftarrow 1
3. `n` \leftarrow `|concepts_list|`
4. **for** `i` \leftarrow 1 **to** `n` **do**
5. `wiki_page` \leftarrow `wikipediaapi.Wikipedia('pt').page(concepts_list[i])`
6. `wiki_pages.append(wiki_page)`
7. **end for**
8. `n` \leftarrow `|wiki_pages|`
9. **for** `i` \leftarrow 1 **to** `n` **do**
10. `wiki_page` \leftarrow `wiki_pages[i]`
11. `sections_list` \leftarrow `wiki_page.sections()`
12. `k` \leftarrow `|sections_list|`
13. **for** `j` \leftarrow 1 **to** `k` **do**

```

14.         section ← sections_list[k]
15.         topic ← create_topics(section.title, wiki_page)
16.         concepts_list ← extract_concepts(topic)
17.         lo ← Learning_object()
18.         lo.instance_name ← "LO_TEMP_" + str(instance_number)
19.         lo.concept ← concepts_list
20.         lo.title ← section.title
21.         lo.unique_identifier ← wiki_page.fullurl + "#" + lo.title.replace(" ", "_")
22.         lo_list.append(lo)
23.         instance_number ← instance_number + 1
24.     end for
25. end for
26. return lo_list

```

O pseudocódigo `wiki_pages_2_LOs` corresponde ao algoritmo de criação de OAs com conteúdo de seções wiki executado na etapa 7 do ARP (Seção 4.5.1). A entrada desse pseudocódigo é uma lista de strings, cada qual correspondendo a um conceito do OA ideal ativo na ontologia.

As variáveis `wiki_pages` e `lo_list` (linha 01) são como listas do Python. Uma lista pode ser inicializada vazia e depois preenchida incrementalmente pelo seu método *append* que adiciona um novo elemento no final da lista. A busca por páginas wiki é realizada pelo método *page* da biblioteca *wikipediaapi* (linha 05). A busca é direcionada a páginas em português ('pt') e o método *page* é executado *n* vezes, sendo *n* o número de conceitos (palavras-chave) do OA ideal. Assim, a variável `wiki_pages` (linha 06) é preenchida com *n* páginas wiki, considerando que o método de busca obteve sucesso todas as vezes. Como esse método pode retornar uma ou nenhuma página wiki, o número de páginas wiki (linha 08) é menor ou igual ao número de conceitos do OA ideal (linha 03).

No restante do pseudocódigo, o objetivo é gerar uma lista de OAs (`lo_list`) que armazena um OA para cada seção wiki das páginas retornadas pelo método de busca. Tais páginas são objetos com métodos que facilitam sua manipulação. O método *sections* (linha 11), por exemplo, retorna as seções de sua página wiki. Dada uma página wiki (linha 10) com *k* seções (linha 12), para cada seção (linha 14) é criado um tópico (linha 15) e para cada qual uma lista de conceitos (linha 16). Cada tópico é formado por uma hierarquia de títulos que começa pelo título da página wiki e termina no título da seção (*section.title*). O procedimento para a criação de tópicos e extração de conceitos é detalhado na Seção 4.4.

Os OAs são objetos da classe *Learning_object* (linha 17). O atributo *instance_name* de cada OA é utilizado para nomear a sua instância armazenada na ontologia. Como eles serão armazenados como OAs temporários, o valor de *instance_name* do primeiro OA criado é "TEMP_1", do segundo OA criado é "TEMP_2", e assim por diante. Na linha 18, note que esses nomes resultam da concatenação (operador +) da string "TEMP_" com o valor da variável `instance_number` transformado em uma string pelo método *str*.

Nas linhas 19, 20 e 21 os demais atributos de cada OA são preenchidos. Note que a lista de conceitos gerada (na linha 16) é atribuída à variável *lo.concept* (na linha 19). Observe que o título de cada OA (associado ao campo 1.2 do padrão IEEE-LOM) corresponde ao título da seção associada a ele (linha 20). Note também que o identificador universal de cada OA (campo 1.1.2 do IEEE-LOM) equivale ao URI da seção wiki (linha 21), formado pela concatenação de três strings: o Uniform Resource Locator (URL) da página wiki, o símbolo # e o título da seção wiki (com os espaços substituídos pelo *underline*). Cada OA preenchido é adicionado na lista de OAs (na linha 22) e no final do processo essa lista é retornada.

4.5.3 Algoritmo genético para a solução do PROA

O algoritmo genético explicado nesta seção corresponde ao AG exibido na Figura 5 e na linha 15 do ARP. Esse AG pode ser utilizado tanto na primeira quanto na segunda tentativa de recomendação. Ele é inspirado em alguns aspectos do AG que resolve o PCC, apresentado na Seção 2.9. A entrada do AG que resolve o PROA deve ser uma instância do PCC, ou seja, uma matriz, cujas linhas correspondem a conceitos do OA ideal e cujas colunas correspondem aos OAs disponíveis para recomendação. Sendo assim, antes da explicação do AG, apresenta-se na subseção a seguir uma etapa de pré-processamento da sua entrada para torná-la instância do PCC.

4.5.3.1 Pré-processamento da entrada do PROA

A lista de OAs das linhas 8 e 13 do ARP e o OA ideal ativo são as variáveis utilizadas na criação da matriz de entrada do AG. Esse procedimento é executado na linha 14 do ARP. Cada conceito (palavra-chave) de OA_{ideal} corresponde a uma linha dessa matriz e cada OA de *lista_oas* corresponde a uma coluna. Formalmente, dada uma matriz m_{ij} , instância do PROA, com m linhas e n colunas, então m é o número de conceitos do OA ideal e n o tamanho da lista de OAs. Cada célula (i, j) da matriz é 1 se a coluna j (OA_j) cobre a linha i (conceito i), e 0, caso contrário.

Um procedimento importante no pré-processamento da entrada é eliminar as linhas que não são cobertas por nenhuma coluna, pois o AG busca sempre cobrir todas as linhas. Além disso, as colunas que cobrem de maneira única ao menos uma linha são adicionadas diretamente na solução e todos as linhas (conceitos) que essas colunas cobrem são eliminadas da instância do problema. A prioridade dada no contexto deste trabalho é a mesma do PCC, ou seja, cobrir todos os conceitos do OA ideal (aqueles que podem ser cobertos).

Na matriz resultante, procede-se com o cálculo da função custo (c_j) de cada coluna (OA). Até nove parâmetros podem ser utilizados nesse cálculo, contudo esse número pode ser alterado para suprir as demandas de cada contexto. A utilização ou não de um dado

parâmetro depende do peso atribuído a ele. Apenas os parâmetros com peso maior que zero são utilizados no cálculo do c_j da Eq. (12). O processo de cálculo do c_j segue o mesmo esquema do cálculo da similaridade s_j da Eq. (13).

Uma vez que a matriz de entrada do AG tenha sido criada, as colunas são ordenadas em ordem crescente de custo. Então, procede-se com a execução do AG que resolve o PROA.

4.5.3.2 Representação do indivíduo

A estrutura do cromossomo de cada indivíduo do AG que resolve o PROA é dada por um vetor de n valores inteiros conforme a Figura 12. O n corresponde ao número de linhas da matriz de entrada m_{ij} e os valores inteiros correspondem às colunas. Um valor inteiro j na i -ésima posição implica que a linha i é coberta pela coluna j .

Linhas (genes)	1	2	3	4	5	...	$n - 1$	n
Vetor de inteiros (Colunas)	5	23	12	15	12	...	8	7

Figura 12 – Representação do cromossomo de um indivíduo

4.5.3.3 Função de avaliação

A aptidão de um indivíduo está diretamente associada ao valor da função objetivo do PCC (Eq. (1)). Com a representação de um indivíduo por valores inteiros, a *fitness* F de um indivíduo i é calculada por:

$$F_i = \sum_{j=1}^n c_j s_{ij} \quad (14)$$

onde s_{ij} é 1 se a coluna j pertence ao indivíduo i , ou seja, ao menos uma linha (gene) do indivíduo é coberta pela coluna j . O c_j corresponde ao custo da coluna (objeto de aprendizagem) j . O cálculo do c_j é realizado pela Eq. (12).

4.5.3.4 Técnica de seleção

O AG que resolve o PROA utiliza o método de seleção por torneio. São realizados quatro torneios com dois indivíduos em cada um. Esses indivíduos são escolhidos aleatoriamente na população. O indivíduo mais apto (com menor *fitness*) de cada torneio é selecionado, obtendo-se assim quatro indivíduos.

4.5.3.5 Cruzamento

Os quatro indivíduos selecionados pelo método do torneio são agrupados para formar duas duplas. Em cada dupla, aplica-se o cruzamento incremental para criar um novo indivíduo, conforme a Figura 13.

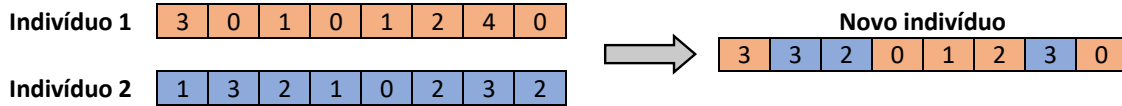


Figura 13 – Crossover incremental de indivíduos do AG que resolve o PROA

Dados dois indivíduos 1 e 2, cada qual com *fitness*, respectivamente, F_1 e F_2 , o *crossover* entre eles gera um novo indivíduo do qual cada gene corresponde ao valor do gene de mesma posição de um dos pais. O gene do indivíduo 1 é escolhido com uma probabilidade $P = F_2/(F_1 + F_2)$. Consequentemente, a chance do gene do indivíduo 2 ser escolhido é $1 - P$.

Esse cruzamento é inspirado no crossover incremental apresentado por Beasley e Chu (1996). A principal diferença está na representação dos indivíduos nos quais esse operador é aplicado. No AG apresentado pelos autores, os cromossomos são representados por um vetor binário, conforme a Figura 2, em que cada gene corresponde a uma coluna (objeto de aprendizagem). Em contrapartida, cada gene dos cromossomos do AG que resolve o PROA corresponde a uma linha (conceito), conforme a Figura 12.

Esse operador de fusão leva em conta a estrutura e a aptidão relativa das soluções pais. A vantagem desse tipo de *crossover*, em relação ao *crossover* de um ponto e de dois pontos, está na criação de novas soluções quando os pais são estruturalmente semelhantes.

4.5.3.6 Mutação

Após alguns experimentos com o AG que resolve o PROA, constatou-se que uma taxa de mutação de 10% é o valor mais adequado para a expansão aleatória do espaço de busca do AG. Em um indivíduo com 100 genes, por exemplo, 10 genes são escolhidos aleatoriamente para sofrer mutação. Esse processo de mutação é tradicional e bem mais simples que o esquema de mutação proposto por Beasley e Chu (1996).

O valor do gene que sofre a mutação é substituído aleatoriamente por um outro valor inteiro que indexa alguma coluna. Esse processo pode gerar soluções inviáveis, pois a coluna escolhida pode não cobrir o gene (linha) ao qual ela foi associada. Na próxima seção, apresenta-se um procedimento que é executado após a mutação para tornar as soluções viáveis e eliminar as colunas redundantes.

4.5.3.7 Soluções viáveis

Assim como no trabalho de Beasley e Chu (1996), utiliza-se operadores adicionais para tornar todas as soluções viáveis. Nas soluções viáveis não há colunas descobertas. Um operador heurístico é utilizado para garantir que todas as colunas sejam cobertas e mais um operador adicional é implementado para eliminar as colunas redundantes, que são aquelas que após serem eliminadas de uma dada solução, essa permanece viável.

O procedimento que torna as soluções viáveis substitui os valores inteiros (colunas) das linhas (genes) descobertas por colunas que cobrem essas linhas. A coluna escolhida para cobrir uma linha descoberta sempre é a coluna mais à esquerda na matriz e, consequentemente, a de menor custo, dado que as colunas são ordenadas em ordem crescente de custo, o que torna essa heurística gulosa.

Esse processo pode tornar algumas colunas redundantes e para eliminar esse tipo de coluna utiliza-se de uma estratégia de otimização local para tornar o AG mais eficaz. O procedimento que implementa essa estratégia trabalha eliminando cada coluna (em ordem decrescente de custo) que não cobre ao menos um conceito de maneira única. Em uma dada solução (indivíduo), a coluna que cobre apenas conceitos já cobertos por outras colunas da solução é substituída pela coluna de menor custo da solução. Note que esse operador utiliza também uma heurística gulosa, pois as colunas que podem ser eliminadas são escolhidas em ordem decrescente de custo e são substituídas pela coluna de menor custo da solução.

4.5.3.8 Visão geral do AG que resolve o PROA

As etapas do AG que resolve o PROA (linha 15 do ARP) são apresentadas a seguir.

1. Gere uma população inicial de N soluções randômicas. Faça $t = 0$.
2. Selecione quatro soluções P_1 , P_2 , P_3 e P_4 da população usando a seleção por torneio.
3. Combine P_1 e P_2 para formar uma nova solução C_1 usando o operador de *crossover* por fusão. Aplique o mesmo processo em P_3 e P_4 para formar uma nova solução C_2 .
4. Proceda com a mutação em C_1 e C_2 .
5. Torne C_1 e C_2 viáveis e remova as colunas redundantes em C_1 e C_2 aplicando o operador heurístico.
6. Se C_1 ou C_2 é idêntico a qualquer uma das soluções na população, vá para a etapa 2; caso contrário, faça $t = t + 1$ e vá para a etapa 7.
7. Substitua duas soluções escolhidas randomicamente com uma *fitness* acima da média na população por C_1 e C_2 .
8. Repita as etapas 2-7 até que $t = M$ soluções não duplicadas tenham sido geradas. A melhor solução encontrada é aquela com a menor *fitness* na população.

A população inicial é gerada aleatoriamente com N soluções, todas viáveis. Os parâmetros configuráveis, tais como o N e o número de gerações M , são considerados nos experimentos. Assim que uma nova solução filha é gerada, ela substitui algum indivíduo da população com *fitness* acima da média. Vale lembrar que quanto menor a *fitness*, melhor o desempenho do indivíduo. Esse é o mesmo método de substituição incremental utilizado no AG que resolve o PCC (BEASLEY; CHU, 1996). A vantagem desse método é que as melhores soluções são mantidas na população e a solução mais recentemente criada é prontamente disponibilizada para seleção e reprodução, acelerando a convergência do AG.

Experimentos e Análise dos Resultados

Neste capítulo, apresentam-se os métodos de validação das hipóteses, os experimentos e a avaliação dos resultados. O protótipo do SCROA implementado em Python foi utilizado para a execução dos experimentos e verificação da eficiência da abordagem. Os testes foram realizados em um notebook com 4GB de RAM e processador Intel Core i5 de 2.6 GHz.

5.1 Métodos e Experimentos

No ARP, há três situações de recomendação de OAs que se caracterizam pelos conceitos do OA ideal. Na primeira situação, o OA ideal não possui conceitos e, portanto, não há um PROA (baseado em conceitos) para ser resolvido, então um processo de recomendação mais simples é executado com base nos demais parâmetros do OA ideal. As outras duas situações correspondem às duas tentativas de recomendação do SCROA, mostradas na Figura 5, nas quais o OA ideal possui conceitos e, neste caso, utiliza-se um AG para resolver o PROA. Para validar a abordagem nessas três situações, executam-se dois grupos de testes:

1. Grupo de testes A: O OA ideal não possui conceitos, então a recomendação é executada pelo procedimento da linha 7 do ARP, que compara os parâmetros do OA ideal com os parâmetros dos demais OAs;
2. Grupo de testes B: O OA ideal possui conceitos, os quais podem ser cobertos pelos OAs sugeridos (primeira tentativa), ou o OA ideal possui ao menos um conceito que não é coberto pelos OAs sugeridos na primeira tentativa de recomendação, então executa-se a segunda tentativa, na qual seções wiki são convertidas em OAs que cobrem os conceitos descobertos.

Os parâmetros utilizados nos dois grupos de testes são apresentados na Tabela 5.

Tabela 5 – Parâmetros utilizados nos testes

OA ideal	Grupo de Testes A	Grupo de Testes B
Conceitos (Palavras-chave)	-	Reprodução, Mitose, Meiose, Célula, Tecido adiposo, Tecido conjuntivo, Epitélio, Protista, Animalia, Plantae
Título	Reprodução nos Seres Vivos	Reprodução nos Seres Vivos
Tipos de recursos de aprendizagem	<i>additional reading, forum activity, animation, exercise, reflection quiz, self assessment</i>	<i>additional reading, forum activity, animation, exercise, reflection quiz, self assessment</i>
Densidade semântica	<i>low</i>	<i>low</i>
Grau de dificuldade	<i>easy</i>	<i>easy</i>
Qualidade	1.0	1.0

Os parâmetros “descrição”, “tipo de interatividade” e “nível de interatividade” foram desativados (peso = 0) nos testes, simulando, assim, uma situação em que esses parâmetros não foram preenchidos pelo usuário.

Todos os parâmetros do OA ideal são definidos pelo usuário, exceto os tipos de recursos de aprendizagem que são inferidos por regras SWRL usando os estilos de aprendizagem do estudante. O perfil do estudante, utilizado para encontrar os tipos de recursos de aprendizagem do OA ideal (Tabela 5), é dado por:

1. Entrada: Verbal;
2. Organização: Sequencial;
3. Percepção: Intuitivo;
4. Processamento: Ativo.

Para a execução dos testes no protótipo do SCROA, a ontologia foi enriquecida com 200 OAs. Alguns métodos do SCROA foram aproveitados para o povoamento da ontologia. O algoritmo de conversão de seções wiki em OAs, por exemplo, foi utilizado para a criação de 114 dos 200 OAs. Eles estão associados a 114 seções wiki derivadas de 10 páginas da Wikipédia resultantes da busca pelos conceitos Reprodução, Mitose, Meiose, Célula, Tecido adiposo, Tecido conjuntivo, Epitélio, Protista, Animalia e Plantae. Os outros 86 OAs foram criados aleatoriamente. De fato, foram criados apenas seus metadados, pois o conteúdo não é necessário para os testes.

Dentre os 114 OAs, os metadados dos parâmetros título e conceitos são preenchidos automaticamente, utilizando-se respectivamente o título da seção wiki e o procedimento de extração de conceitos. Em relação aos 86 OAs, o parâmetro título é preenchido aleatoriamente com um dos títulos dos 114 OAs e o parâmetro conceito é preenchido aleatoriamente com um dos conceitos do OA ideal. Dos 200 OAs, 190 cobrem apenas um conceito e os outros 10 OAs cobrem dois conceitos. Para a execução dos testes, é interessante que os OAs cubram um número de conceitos parecido para que eles tenham a mesma chance de ser recomendados.

Os parâmetros “tipos de recursos de aprendizagem”, “densidade semântica”, “grau de dificuldade” e “qualidade” dos 200 OAs foram preenchidos com dados aleatórios. Para preencher a densidade semântica, por exemplo, gera-se um número inteiro e randômico no intervalo de $[0, 5]$ para indexar no vetor $[very\ low, low, medium, high, very\ high]$ um valor para a densidade semântica.

Para diversificar os testes, o tipo “*wiki content*” (da Figura 8) não foi utilizado para representar os OAs derivados de seções wiki. Para isso, utilizou-se diversos outros tipos de recursos de aprendizagem escolhidos aleatoriamente. Contudo, em cenários reais, parece conveniente o uso do tipo “*wiki content*” para identificar os OAs correspondentes a seções wiki.

Os 200 OAs permanentes foram inferidos como OAs sugeridos, pois possuem ao menos um parâmetro (em geral um conceito) similar ao OA ideal. Esses OAs são utilizados nos dois grupos de testes. Para garantir a variabilidade da amostra de 200 OAs e também a possibilidade de se obter boas soluções, cada um dos três parâmetros, quais sejam, tipo de recurso, densidade semântica e grau de dificuldade, tem 50% de chance de ter um valor igual ao do OA ideal. Isso significa que o valor de cada um desses parâmetros do OA ideal é contemplado, probabilisticamente, por 100 OAs.

O OA ideal ativo e a lista dos 200 OAs sugeridos são utilizados pelo procedimento da linha 7 do ARP (na execução dos testes do grupo A) ou pelo procedimento da linha 14 para a criação da instância do PROA, formada por uma matriz m e o vetor de custos dessa matriz. Essa instância é a entrada do AG clássico (linha 15) utilizado na solução do PROA.

5.1.1 Testes do grupo A

Os testes do grupo A são executados para testar o procedimento da linha 7 do ARP, o qual é executado quando não há conceitos no OA ideal. Neste caso, o processo de recomendação é realizado com base na similaridade dos parâmetros do OA ideal em relação aos demais OAs. O OA ideal (Tabela 5) e os 200 OAs disponíveis para recomendação são utilizados como entrada do procedimento da linha 7 do ARP.

Foram executados quatro testes diferentes variando os pesos dos parâmetros do OA ideal. Na Tabela 6, exibe-se o valor do peso atribuído a cada parâmetro. No teste 1, todos os parâmetros possuem o mesmo peso. Os parâmetros densidade semântica, grau de dificuldade e qualidade recebem, respectivamente, nos testes 2, 3 e 4, um peso bem maior que o peso dos demais parâmetros. O parâmetro “conceitos” é desabilitado porque o OA ideal dos testes do grupo A não possui conceitos.

Os resultados dos quatro testes executados são apresentados na Tabela 7. Nela, apresenta-se os 10 OAs mais similares ao OA ideal em cada teste executado. O score é calculado pela Eq. (13). Quanto maior o score de um dado OA, maior o grau de similaridade entre ele e o OA ideal.

Tabela 6 – Pesos atribuídos ao OA ideal nos quatro testes do grupo A

Parâmetro	Teste 1	Teste 2	Teste 3	Teste 4
Palavras-chave (Conceitos)	desabilitado	desabilitado	desabilitado	desabilitado
Título	1	0.2	0.2	0.2
Tipos de recursos	1	0.2	0.2	0.2
Densidade semântica	1	1	0.2	0.2
Grau de dificuldade	1	0.2	1	0.2
Qualidade	1	0.2	0.2	1

Tabela 7 – Os 10 melhores OAs ranqueados em cada teste

TESTE 1		TESTE 2		TESTE 3		TESTE 4	
OAs	Scores	OAs	Scores	OAs	Scores	OAs	Scores
OA_189	0.7931	OA_189	0.3186	OA_189	0.3186	OA_189	0.3131
OA_133	0.7912	OA_133	0.3182	OA_133	0.3182	OA_133	0.3112
OA_44	0.7909	OA_44	0.3181	OA_44	0.3181	OA_84	0.3103
OA_84	0.7903	OA_84	0.3180	OA_84	0.3180	OA_150	0.3077
OA_150	0.7877	OA_150	0.3175	OA_18	0.3167	OA_111	0.3071
OA_18	0.7839	OA_18	0.3167	OA_3	0.3157	OA_18	0.3039
OA_3	0.7786	OA_20	0.3149	OA_20	0.3149	OA_134	0.3032
OA_20	0.7746	OA_46	0.3113	OA_46	0.3113	OA_79	0.2961
OA_46	0.7569	OA_111	0.3106	OA_102	0.3105	OA_105	0.2960
OA_111	0.7533	OA_102	0.3105	OA_73	0.3093	OA_20	0.2946

O teste 1 é comparado com os testes 2, 3 e 4, respectivamente, nas Figuras 14, 15 e 16. Na Figura 14 (a), o OA_3 possui densidade semântica *very low* (0), que é diferente da densidade semântica do OA ideal (*low* = 0.25). Na Figura 14 (b), atribui-se um peso maior ao parâmetro densidade semântica priorizando a similaridade entre o OA ideal e os demais OAs em relação a esse parâmetro. Note que, neste caso, os 10 OAs melhor ranqueados possuem a mesma densidade semântica do OA ideal.

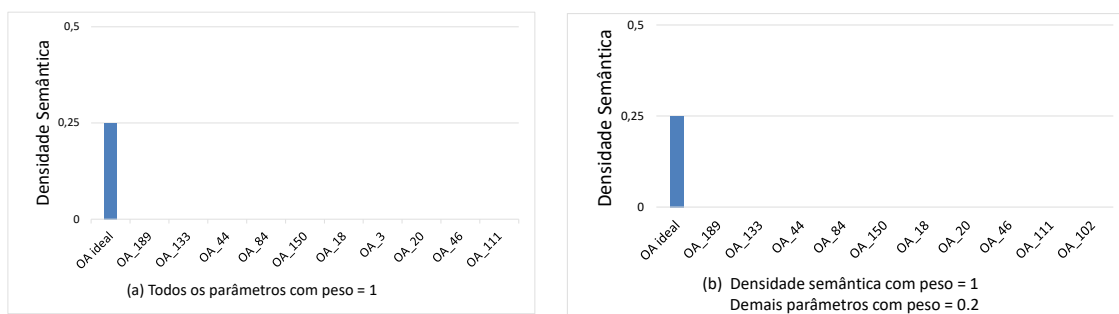


Figura 14 – Comparação da densidade semântica do OA ideal em relação aos demais OAs

Se um parâmetro possui um peso maior que o peso dos demais parâmetros, então os OAs recomendados tendem a ser iguais ao OA ideal com relação a esse parâmetro. Os 10 OAs mais similares ao OA ideal na Figura 15 (b) têm o mesmo grau de dificuldade do OA ideal. Note que quando todos os pesos são iguais, Figura 15 (a), isso não ocorre, o OA_150 e o OA_111 possuem um grau de dificuldade maior que o do OA ideal.

A mesma situação ocorre para o parâmetro qualidade. Comparando-se as partes (a)

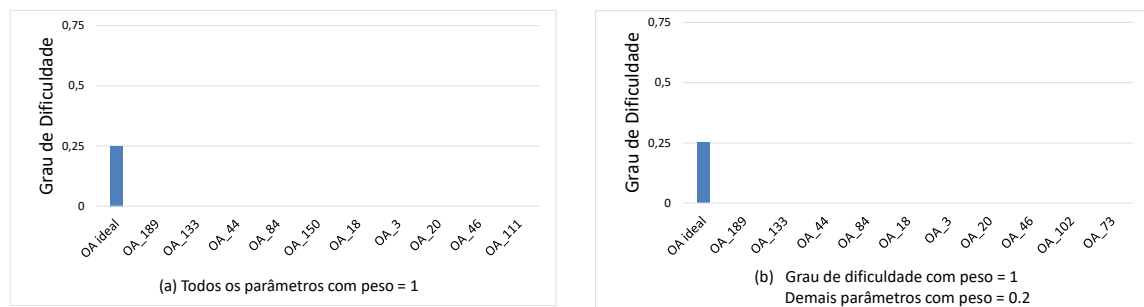


Figura 15 – Comparação do grau de dificuldade do OA ideal em relação aos demais OAs

e (b) da Figura 16, observa-se que a qualidade dos 10 OAs recomendados é maior (mais próxima da qualidade do OA ideal) quando esse parâmetro tem um peso maior que os demais parâmetros.

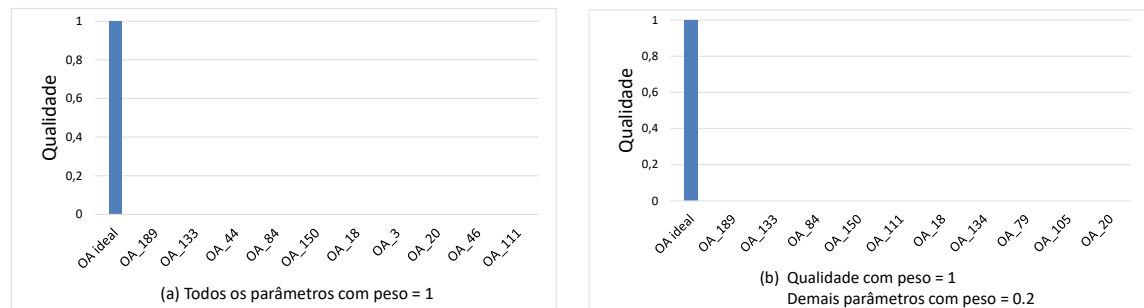


Figura 16 – Comparação do parâmetro qualidade do OA ideal em relação aos demais OAs

O tempo de execução do procedimento da linha 7 do ARP é mostrado na Figura 17. Nela, observa-se que esse procedimento tem um tempo linear em relação ao tamanho da entrada (número de OAs). O gráfico da figura representa o tempo médio de 10 execuções deste procedimento para cada tamanho de entrada.

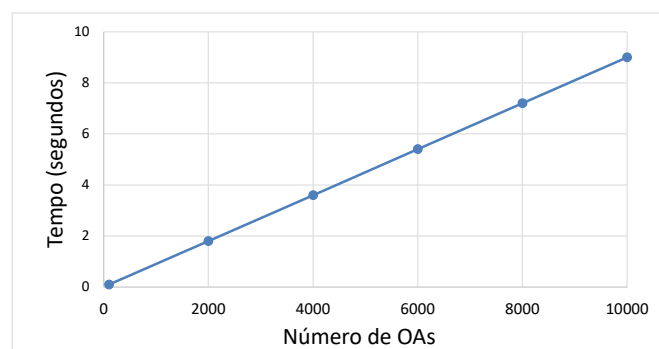


Figura 17 – Tempo do processo que compara o OA ideal com os demais OAs

5.1.2 Testes do grupo B

Os quatro testes do grupo B correspondem à situação em que o OA ideal possui conceitos e, neste caso, uma ou duas tentativas de recomendação podem ser realizadas:

- a) Tentativa 1: Todos os conceitos do OA ideal podem ser cobertos pelos OAs sugeridos na ontologia;
- b) Tentativa 2: O OA ideal possui ao menos um conceito que não é coberto pelos OAs sugeridos na primeira tentativa de recomendação, então o procedimento que converte seções wiki em OAs é executado, favorecendo a cobertura dos conceitos descobertos.

Em ambos os casos, utiliza-se um AG para resolver o PROA. O PROA a ser resolvido está vinculado ao OA ideal apresentado na coluna “Grupo de Testes B” da Tabela 5. Assim, a instância do PROA criada na linha 14 do ARP corresponde a uma matriz de 10 linhas (uma para cada um dos 10 conceitos do OA ideal) e 200 colunas (uma para cada um dos 200 OAs criados para a execução dos testes) mais o vetor de custos dessa matriz. Na Tabela 8, apresentam-se os pesos assumidos pelos parâmetros em cada teste executado.

Tabela 8 – Pesos atribuídos ao OA ideal nos quatro testes do grupo B

Parâmetro	Teste 1	Teste 2	Teste 3	Teste 4
Palavras-chave (Conceitos)	1	0.2	0.2	1
Título	1	0.2	0.2	1
Tipos de recursos	1	0.2	0.2	1
Densidade semântica	1	1	0.2	1
Grau de dificuldade	1	0.2	1	1
Qualidade	1	0.2	0.2	Desabilitado

O AG foi executado para a instância do PROA criada. Realizou-se 100 mil iterações (com a geração de dois novos indivíduos em cada uma), considerando uma taxa de mutação de 10% e uma população com 100 indivíduos. Na Tabela 9, apresentam-se as 5 melhores soluções de cada teste do grupo B. A primeira solução do Teste 1, por exemplo, é composta pelos OAs 90, 194, 44, 200, 133 e 198. Quanto menor a *fitness*, maior a similaridade dos OAs recomendados em relação ao OA ideal.

Para mostrar como o peso de cada parâmetro influencia na escolha dos OAs que compõem cada solução, o Teste 1 foi comparado com os Testes 2, 3 e 4. Comparando-se as partes (a) e (b) da Figura 18, percebe-se que a densidade semântica média das 5 melhores soluções é maior quando o peso desse parâmetro é maior que o dos demais parâmetros. Esse valor médio é obtido pela média aritmética dos valores assumidos pelo referido parâmetro nos OAs de uma dada solução.

Esse mesmo raciocínio pode ser aplicado aos outros parâmetros. O grau de dificuldade médio da 5 melhores soluções, por exemplo, é maior quando esse parâmetro tem um peso maior que o dos demais parâmetros, conforme a Figura 19.

Tabela 9 – As 5 melhores soluções em cada teste do grupo B

TESTE 1		TESTE 2		TESTE 3		TESTE 4	
Soluções	Fitness	Soluções	Fitness	Soluções	Fitness	Soluções	Fitness
OA_90	2.4550	OA_90	4.5243	OA_197	4.5434	OA_197	2.3010
OA_194		OA_194		OA_194		OA_194	
OA_44		OA_44		OA_44		OA_167	
OA_200		OA_200		OA_62		OA_62	
OA_133		OA_133		OA_196		OA_196	
OA_198		OA_198		OA_198		OA_198	
OA_197	2.4679	OA_90	4.5286	OA_197	4.5477	OA_197	2.3122
OA_194		OA_194		OA_194		OA_194	
OA_44		OA_46		OA_46		OA_167	
OA_200		OA_200		OA_62		OA_200	
OA_133		OA_133		OA_196		OA_133	
OA_198		OA_198		OA_198		OA_198	
OA_90	2.4764	OA_90	4.5289	OA_197	4.5480	OA_197	2.3122
OA_194		OA_194		OA_194		OA_194	
OA_46		OA_167		OA_167		OA_167	
OA_200		OA_200		OA_62		OA_200	
OA_133		OA_133		OA_196		OA_192	
OA_198		OA_198		OA_198		OA_198	
OA_90	2.4782	OA_90	4.5342	OA_197	4.5548	OA_197	2.3243
OA_194		OA_194		OA_194		OA_194	
OA_167		OA_44		OA_199		OA_167	
OA_200		OA_200		OA_62		OA_49	
OA_133		OA_192		OA_196		OA_196	
OA_198		OA_198		OA_92		OA_198	
OA_197	2.4893	OA_90	4.5365	OA_197	4.5553	OA_197	2.3330
OA_194		OA_194		OA_194		OA_194	
OA_46		OA_40		OA_44		OA_167	
OA_200		OA_200		OA_50		OA_200	
OA_133		OA_133		OA_196		OA_67	
OA_198		OA_198		OA_198		OA_198	

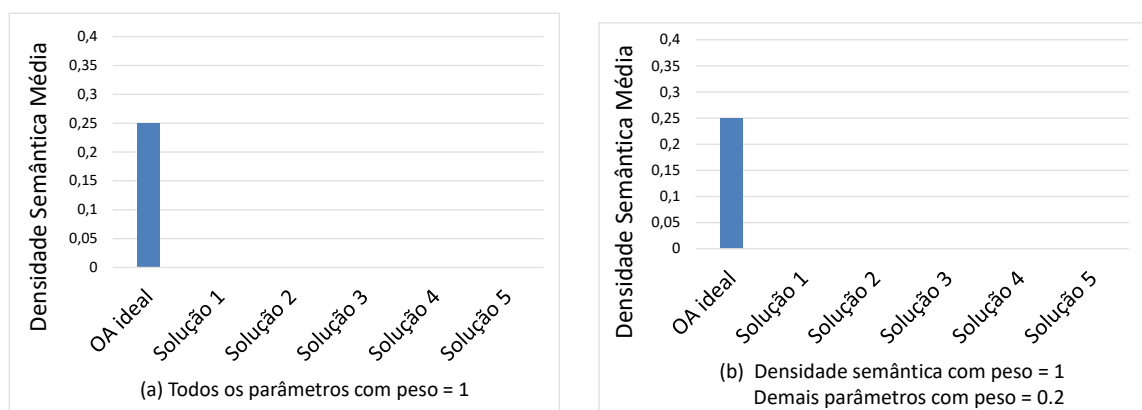


Figura 18 – Comparação da densidade semântica média das 5 melhores soluções em relação ao OA ideal

Para mostrar a importância do parâmetro qualidade, na Figura 20, compara-se a qualidade média das 5 melhores soluções em relação ao OA ideal quando o parâmetro qualidade é considerado, parte (a), e quando ele está desabilitado, parte (b). Comparando-

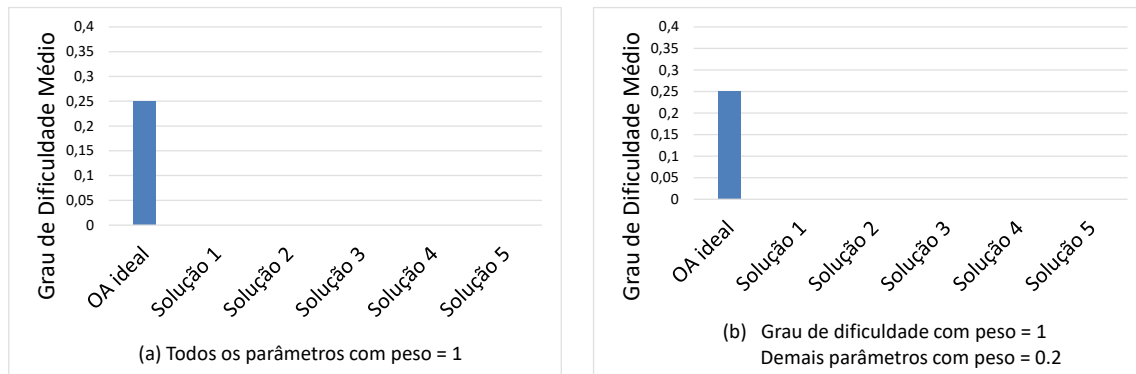


Figura 19 – Comparação do grau de dificuldade médio das 5 melhores soluções em relação ao OA ideal

se as duas partes da Figura 20, observa-se que considerar esse parâmetro eleva a qualidade dos OAs melhor recomendados.

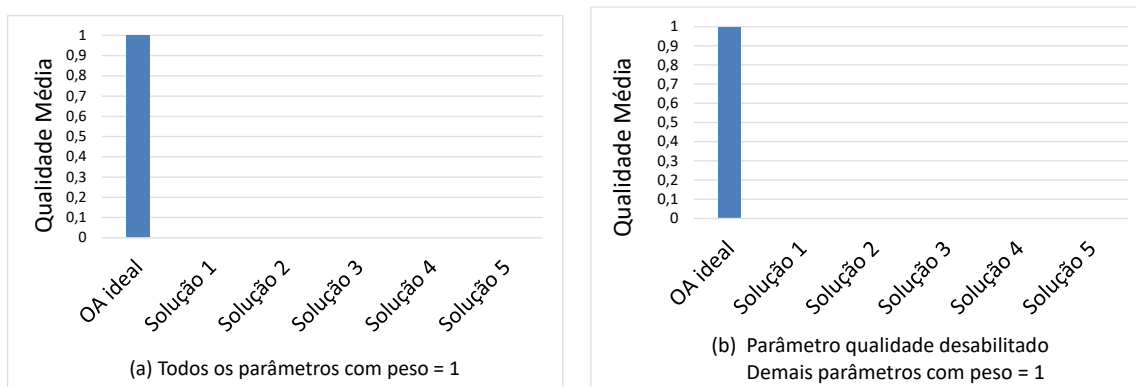


Figura 20 – Comparação da qualidade média das 5 melhores soluções em relação ao OA ideal

A eficiência do AG em termos de tempo de execução é mostrada na Tabela 10. Cada tempo apresentado corresponde à média aritmética de dez execuções do AG. Em geral, quando o número de OAs aumenta o tempo de execução é maior, mas também pode ser menor. O tempo de execução aumenta significativamente quando o número de conceitos ou iterações aumenta.

O AG também foi testado para a instância 4.10 do *benchmark* OR-Library (citado no início do Capítulo 3). Essa é uma instância grande do PCC que corresponde a uma matriz com 200 linhas e 1000 colunas mais um vetor de custos. Utilizou-se essa instância para a comparação da eficiência do AG em relação a um algoritmo totalmente aleatório (sem os operadores genéticos). O resultado dessa comparação é mostrado na Figura 21. Realizou-se 5001 iterações com a geração de dois novos indivíduos em cada uma, considerando uma população com 100 indivíduos. O AG tem uma taxa de mutação de 10%. O tempo gasto pelo AG para a execução de todas as iterações é de quase 3 minutos.

Tabela 10 – Tempo (em segundos) de execução do AG

Nº de OAs	Nº de Conceitos	Número de iterações			
		500	5000	100000	500000
100	10	0.23s	5.85s	135.84s	649.02s
500	10	0.32s	5.60s	135.67s	605.90s
1000	10	0.37s	6.75s	150.78s	609.60s
100	50	0.77s	8.08s	201.31s	839.74s
500	50	1.25s	9.57s	189.85s	976.82s
1000	50	1.59s	10.43s	206.05s	1177.52s

Nota: Utilizou-se a mesma configuração do Teste 1, variando o número de OAs, de conceitos e de iterações. A cada iteração gera-se dois novos indivíduos.

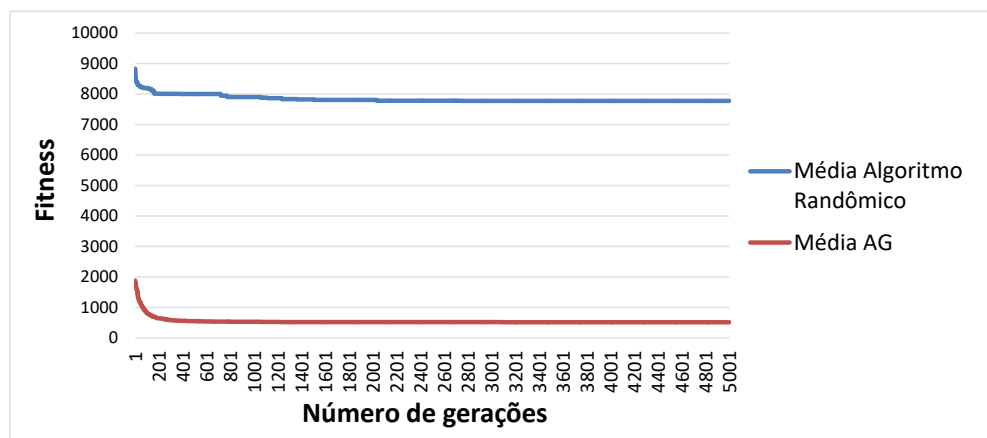


Figura 21 – Comparação entre o AG e um algoritmo totalmente aleatório

O AG que resolve o PROA é executado na primeira ou segunda tentativa de recomendação. A diferença é que, na segunda tentativa, antes da execução do AG, é necessário a criação de novos OAs dado que os OAs sugeridos na ontologia não são suficientes para cobrir todos os conceitos do OA ideal. Neste caso, executa-se o procedimento *wiki_pages_2_LOs* (ver Subseção 4.5.2) para a conversão de seções wiki em OAs. Esse procedimento leva em média 2.2 segundos para a criação de um novo OA. Esse tempo sofre uma pequena variação de acordo com o número de seções da página wiki. Vale lembrar que esse procedimento também gera automaticamente os conceitos para o campo palavras-chave de cada novo OA criado.

5.2 Avaliação dos Resultados

O SCROA possui uma abordagem que contribui na criação de novos OAs fazendo o reúso de conteúdo wiki de maneira eficiente. O conteúdo wiki pode ser buscado rapidamente e os conceitos de cada seção extraídos de maneira eficiente pelo processo de extração de conceitos apresentado na Seção 4.4.

Embora pareça intuitivo que os conceitos de uma seção possam ser extraídos da hierarquia de títulos, a confirmação da eficiência desse processo depende de questões como: “Até que ponto as palavras-chave podem ser tratadas como conceitos?”. As repostas a essas questões pelas pesquisas futuras são importantes porque a maioria dos padrões de metadados educacionais agregam algum campo do tipo palavra-chave e, geralmente, os diversos recursos educacionais que seguem esses padrões preenchem esse campo.

Em geral, as pesquisas que tratam da recomendação de OAs utilizando conceitos implementam ontologias de domínios do conhecimento criadas por especialistas. Nos trabalhos de Acampora et al. (2011) e Gaeta et al. (2013), por exemplo, os autores implementam esse tipo de ontologia para modelar conceitos e suas relações (como os pré-requisitos). A principal desvantagem desse tipo de abordagem é o alto custo para a criação desses domínios. A vantagem da abordagem do SCROA é que ela não depende de ontologias de domínios do conhecimento. Assim, a abordagem apresentada neste trabalho não limita os assuntos de interesse do usuário por conceitos modelados em ontologias.

Para garantir a qualidade do conteúdo wiki recomendado, o SCROA utiliza um classificador (WARNCKE-WANG; COSLEY; RIEDL, 2013) que com poucas características extraídas facilmente de uma página, procede com a classificação do conteúdo wiki com uma taxa de acerto de 87%. Atribuem-se valores no intervalo de $[0, 1]$ às diversas classes de qualidade do classificador, possibilitando o uso do parâmetro qualidade no escopo do SCROA. O classificador não foi implementado no protótipo do SCROA, mas isso não compromete os resultados, sendo os valores do parâmetro de qualidade gerados de maneira aleatória na execução dos testes.

Nas Figuras 16 e 20, mostra-se que a utilização do parâmetro de qualidade eleva a qualidade média do conteúdo das melhores soluções. A maioria dos testes realizados mostram que a flexibilidade na configuração dos pesos dos parâmetros do OA ideal é importante para quando se pretende valorizar alguns parâmetros mais do que outros.

Quando o parâmetro qualidade tem um peso maior que os demais parâmetros (Figura 16, parte b), a qualidade média dos 10 OAs melhor ranqueados aumenta em relação ao caso em que todos os parâmetros possuem o mesmo peso (parte a).

Quando o parâmetro qualidade é desabilitado (Figura 20, parte b), a qualidade média das melhores soluções diminui em relação ao caso em que todos os parâmetros possuem o mesmo peso (parte a). Assim, comprova-se que a utilização do parâmetro qualidade eleva a qualidade do conteúdo das melhores soluções.

A hipótese de que as seções wiki podem ser convertidas em OAs é confirmada. Esse processo pode ser automatizado de maneira eficiente levando em conta a qualidade do conteúdo wiki.

Em todos os testes, o AG converge no início do processo. Uma característica que torna esse algoritmo eficiente na solução do PROA é que as populações geradas (inclusive a inicial) contêm apenas indivíduos que são soluções viáveis (todo indivíduo é transformado

em uma solução viável antes de ser adicionado à população), acelerando a convergência.

A segunda hipótese deste trabalho é também confirmada, pois o AG que resolve o PROA recomenda OAs que cobrem todos os conceitos do OA ideal e que atendem ao estilo de aprendizagem do estudante.

O tempo gasto com os procedimentos do SCROA pode variar de acordo com o tamanho das páginas wiki utilizadas na construção de OAs e com o tamanho da instância do PROA, ou seja, a quantidade de conceitos e OAs disponíveis para recomendação. Após o usuário entrar com os parâmetros do OA ideal e o perfil do estudante, o processo de inferência é executado para a obtenção de uma lista de OAs sugeridos. Esse processo é rápido, leva poucos segundos.

Dada uma lista de 10 mil OAs sugeridos, se não houver conceitos no OA ideal definido pelo usuário, então o processo de recomendação leva menos que 10 segundos, conforme a Figura 17.

Se há conceitos no OA ideal, então o PROA é resolvido por um AG e, neste caso, o processo de recomendação de OAs pode ser mais demorado. Considerando que o AG converge em aproximadamente 500 iterações (Figura 21), conforme a Tabela 10, em geral, quanto maior o tamanho da instância do PROA (número de OAs e conceitos) maior o tempo gasto. Assim, o tempo desse processo de recomendação pode levar menos que 2 segundos, mas também pode levar alguns minutos.

Pela avaliação dos resultados, o SCROA implementa uma abordagem eficaz para a recomendação de OAs, os quais atendem aos estilos de aprendizagem do estudante e aos parâmetros de busca definidos pelo usuário. Uma atenção deve ser dada ao tempo gasto pelo processo de recomendação. A depender das necessidades de cada contexto, o processo de conversão de seções wiki em OAs (que leva em média 2.2 segundos para cada novo OA criado) pode ser executado antecipadamente para enriquecer a ontologia com os OAs que podem ser sugeridos. O processo de recomendação de OAs também pode ser executado antecipadamente, principalmente em cenários nos quais a instância do PROA é muito grande.

Considerações finais

No contexto do ensino eletrônico, a adaptação do conteúdo de aprendizagem aos estilos de aprendizagem de cada estudante propicia uma aprendizagem mais eficiente (no tempo gasto) e mais eficaz nos resultados alcançados pelo estudante. Essa adaptação é realizada geralmente em Sistemas Hiperídia Adaptativos, que auxiliam professores na potencialização do desempenho dos alunos e também auxiliam estudantes autodidatas a alcançarem seus objetivos sem a intervenção de um professor.

Um problema desse contexto é que a criação de novos OAs por especialistas leva tempo e envolve um alto custo em dinheiro. Este trabalho faz reuso de conteúdo wiki sugerindo que o vasto conteúdo da Web pode ser usado para a criação de recursos educacionais devidamente estruturados. Os conceitos tratados por cada seção wiki são extraídos da hierarquia de títulos da própria seção transformada em OA.

Uma ontologia foi implementada para modelar metadados de OAs conforme o padrão IEEE-LOM e modelar estudantes, especialmente, seus estilos de aprendizagem seguindo o FSLSM. Algumas regras SWRL implementadas na ontologia são utilizadas para inferir os tipos de recursos de aprendizagem apropriados a cada perfil de estudante. Outras regras foram criadas para sugerir OAs com maior chance de serem recomendados ao aprendiz.

O problema de recomendação de OAs, denominado PROA, é baseado em conceitos e, portanto, depende da definição dos conceitos do OA ideal pelo usuário. Quando o usuário atribui valores a alguns parâmetros do OA ideal, mas não define nenhum conceito (palavra-chave), então um procedimento de recomendação mais simples é executado com base apenas nos parâmetros definidos pelo usuário.

O PROA existe quando o usuário define os conceitos que se espera que o aluno aprenda. Neste caso, executa-se um AG que tem como entrada uma matriz (instância do PROA), cujas linhas correspondem aos conceitos do OA ideal e cujas colunas são os OAs sugeridos para recomendação. Como os OAs sugeridos podem não cobrir todos os conceitos, utiliza-se um procedimento de conversão de seções wiki em OAs com o objetivo de se criar uma instância do PCC, garantido que todas as linhas (conceitos) possam ser cobertas.

O AG que resolve o PROA aceita apenas soluções viáveis dentro da população, ace-

lerando o processo de convergência. Os resultados alcançados sugerem que os OAs recomendados são os mais similares possíveis ao OA ideal, atendendo, portanto, aos estilos de aprendizagem do estudante e cobrindo os conceitos a serem aprendidos.

A seguir, detalham-se as principais contribuições deste trabalho. Na Seção 6.2, apresenta-se uma direção para os trabalhos futuros. Finalmente, citam-se dois artigos produzidos no decorrer desta pesquisa.

6.1 Principais Contribuições

Neste trabalho, apresenta-se uma abordagem para o reúso de conteúdo da Web na criação de OAs por meio das tecnologias da Web Semântica, contribuindo para a solução do problema do alto custo com tempo e dinheiro que comumente se tem na criação de novos OAs. Os OAs recomendados cobrem todos os conceitos que se espera que o aluno aprenda e satisfazem os parâmetros de busca do usuário e os estilos de aprendizagem do estudante. Além disso, garante-se a qualidade do conteúdo wiki associado ao OA.

A ontologia implementada contribui no processo de recomendação de OAs por meio de regras de inferência e garante a modelagem de OAs (conforme o padrão IEEE-LOM) e de estudantes, especialmente seus estilos de aprendizagem (seguindo o FSLSM).

Este trabalho valida a primeira hipótese, mostrando que as seções wiki podem ser convertidas em OAs em poucos segundos, inclusive com a geração automática de alguns metadados, tais como “palavras-chave” e “título”. Além disso, o AG que resolve o PROA prioriza os OAs derivados de páginas wiki com maior qualidade. O AG garante que os OAs recomendados cobrem todos os conceitos do OA ideal e atendem aos estilos de aprendizagem do estudante, confirmando assim, a segunda hipótese.

A validação das hipóteses sugere a eficiência da abordagem do SCROA, o qual pode impactar na popularização de Sistemas Hipermídia Adaptativos que permitam que alunos de todas as camadas sociais tenham um melhor desempenho no processo ensino-aprendizagem.

6.2 Trabalhos Futuros

A abordagem utilizada neste trabalho para a criação de OAs usando conteúdo wiki contribui para automatizar o processo de criação de novos OAs. Porém a Wikipédia não pode ser usada para a criação de qualquer tipo de OA. Embora essa enciclopédia digital contenha conteúdo de diversos domínios do conhecimento, ela certamente não possui uma variedade suficiente de tipos de recursos de aprendizagem. Ela agrega textos, imagens e definições, mas não possui conteúdos dos tipos exercício e avaliação, fundamentais no processo ensino-aprendizagem.

Assim, um desafio para as pesquisas futuras é a exploração do conteúdo da Web visando-se a criação de novos tipos de recursos de aprendizagem, tais como exercícios, avaliações, introduções e resumos, que podem enriquecer a abordagem do SCROA.

O segundo desafio é a extração dos conceitos que cada conteúdo da Web cobre. Esse processo depende de questões como: “Até que ponto as palavras-chave podem ser tratadas como conceitos?”. As repostas a essas questões pelos trabalhos futuros são importantes porque a maioria dos padrões de metadados educacionais possuem o campo “palavra-chave” e a maioria dos recursos educacionais que seguem esses padrões preenchem esse campo.

As pesquisas futuras também podem estender a ontologia com a criação de novas classes, regras e novos atributos para a classe Student com objetivo de implementar uma recomendação de OAs que contemple também a filtragem colaborativa. Assim, o histórico da recomendação de OAs para um dado estudante poderia ser explorado para a recomendação de OAs a outros estudantes com perfis e características similares. Além disso, os campos da categoria “Anotação” do padrão IEEE-LOM podem ser utilizados pelo aluno ou professor para avaliar o OA utilizado, e essa informação pode ser explorada para sofisticar o processo de recomendação.

6.3 Contribuições em Produção Bibliográfica

Durante a realização desta pesquisa, dois artigos foram produzidos. O primeiro artigo (BELIZÁRIO JÚNIOR; DORÇA, 2016) foi publicado no V DesafIE do Congresso da Sociedade Brasileira de Computação durante o primeiro semestre da realização desta pesquisa. Esse artigo contém a ideia inicial desta dissertação, qual seja, a recomendação inteligente de conteúdo da Web na criação de recursos educacionais.

Essa ideia inicial foi amadurecida, o escopo reduzido e o objeto de pesquisa refinado, favorecendo a produção deste trabalho que resultou na produção do segundo artigo (BELIZÁRIO JÚNIOR; DORÇA, 2018) de título “Uma Abordagem para a Criação e Recomendação de Objetos de Aprendizagem usando um Algoritmo Genético, Tecnologias da Web Semântica e uma Ontologia”, publicado no XXIX Simpósio Brasileiro de Informática na Educação.

Referências

- ABECH, M. et al. A model for learning objects adaptation in light of mobile and context-aware computing. **Personal and Ubiquitous Computing**, Springer-Verlag, v. 20, n. 2, p. 167–184, 2016. Disponível em: <<https://doi.org/10.1007/s00779-016-0902-3>>.
- ACAMPORA, G.; GAETA, M.; LOIA, V. Exploring e-learning knowledge through ontological memetic agents. **IEEE Computational Intelligence Magazine**, IEEE, v. 5, n. 2, p. 66–77, 2010. Disponível em: <<https://doi.org/10.1109/MCI.2010.936306>>.
- ACAMPORA, G. et al. Optimizing learning path selection through memetic algorithms. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, 2008, Hong Kong, China. **Proceedings...** Washington, USA: IEEE Computer Society, 2008. p. 3869–3875. Disponível em: <<https://doi.org/10.1109/IJCNN.2008.4634354>>.
- _____. An adaptive multi-agent memetic system for personalizing e-learning experiences. In: INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS, 2011, Taipei, Taiwan. **Proceedings...** Washington, USA: IEEE Computer Society, 2011. p. 123–130. Disponível em: <<https://doi.org/10.1109/FUZZY.2011.6007519>>.
- ADVANCED DISTRIBUTED LEARNING INITIATIVE. **Sharable Content Object Reference Model (SCORM)**. Alexandria, VA, USA, 2001. Disponível em: <<http://www.adlnet.gov>>.
- BEASLEY, J. E.; CHU, P. C. A genetic algorithm for the set covering problem. **European Journal of Operational Research**, Elsevier, v. 94, n. 2, p. 392–404, 1996. Disponível em: <[https://doi.org/10.1016/0377-2217\(95\)00159-X](https://doi.org/10.1016/0377-2217(95)00159-X)>.
- BELIZÁRIO JÚNIOR, C. F.; DORÇA, F. A. Uma abordagem para a recomendação inteligente de conteúdo web na criação de materiais e objetos de aprendizagem. In: WORKSHOP DE DESAFIOS DA COMPUTAÇÃO APLICADA À EDUCAÇÃO, 5., 2016, Porto Alegre, RS. **Anais...** Porto Alegre, RS: PUCRS, 2016. p. 567–576.
- _____. Uma abordagem para a criação e recomendação de objetos de aprendizagem usando um algoritmo genético, tecnologias da web semântica e uma ontologia. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 29., 2018, Fortaleza. **Anais...** Fortaleza: Sociedade Brasileira de Computação, 2018. p. 1533–1542.
- BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The semantic web. **Scientific american**, JSTOR, v. 284, n. 5, p. 34–43, 2001.

BRUSILOVSKY, P. Methods and techniques of adaptive hypermedia. **User modeling and user-adapted interaction**, Springer, v. 6, n. 2-3, p. 87–129, 1996.

CHEN, W. et al. A hybrid recommendation algorithm adapted in e-learning environments. **World Wide Web**, Springer, v. 17, n. 2, p. 271–284, 2014. Disponível em: <<https://doi.org/10.1007/s11280-012-0187-z>>.

CHRISTUDAS, B. C. L.; KIRUBAKARAN, E.; THANGAIAH, P. R. J. An evolutionary approach for personalization of content delivery in e-learning systems based on learner behavior forcing compatibility of learning materials. **Telematics and Informatics**, Elsevier, v. 35, n. 3, p. 520–533, 2018. Disponível em: <<https://doi.org/10.1016/j.tele.2017.02.004>>.

CLEO LAB. PARTNERS. **CLEO Extensions to the IEEE Learning Object Metadata**. Washington, USA, 2003. 65 p.

DANG, Q.-V.; IGNAT, C.-L. An end-to-end learning solution for assessing the quality of wikipedia articles. In: INTERNATIONAL SYMPOSIUM ON OPEN COLLABORATION, 13., 2017, Galway, Ireland. **Proceedings...** New York: ACM, 2017. Disponível em: <<https://doi.org/10.1145/3125433.3125448>>.

DEBORAH, L. J.; BASKARAN, R.; KANNAN, A. Learning styles assessment and theoretical origin in an e-learning scenario: a survey. **Artificial Intelligence Review**, Springer, v. 42, n. 4, p. 801–819, 2014. Disponível em: <<https://doi.org/10.1007/s10462-012-9344-0>>.

FELDER, R. M.; SILVERMAN, L. K. Learning and teaching styles in engineering education. **Engineering education**, v. 78, n. 7, p. 674–681, 1988.

GAETA, M. et al. An approach to personalized e-learning. **Journal of Education, Informatics & Cybernetics**, v. 11, n. 1, 2013.

GAREY, M. R.; JOHNSON, D. S. **Computers and intractability**: A guide to the theory of np-completeness. New York: W. H. Freeman & Co., 1979. 340 p. (Series of books in the mathematical sciences).

GENNARI, J. H. et al. The evolution of protégé: An environment for knowledge-based systems development. **Int. J. Hum.-Comput. Stud.**, Elsevier, v. 58, n. 1, p. 89–123, 2003.

GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear**: Modelos e algoritmos. 2. ed. Rio de Janeiro: Campus, 2005. 518 p.

GRAF, S.; KINSHUK; IVES, C. A flexible mechanism for providing adaptivity based on learning styles in learning management systems. In: IEEE INTERNATIONAL CONFERENCE ON ADVANCED LEARNING TECHNOLOGIES, 10., 2010, Washington, USA. **Proceedings...** Washington, USA: IEEE Computer Society, 2010. p. 30–34.

GRUBER, T. R. A translation approach to portable ontology specifications. **Knowledge Acquisition**, Academic Press Limited, v. 5, n. 2, p. 199–220, 1993. Disponível em: <<https://doi.org/10.1006/knac.1993.1008>>.

- JÁCOME JÚNIOR, L.; MENDES NETO, F. M.; SILVA, L. C. N. Uma abordagem baseada em algoritmo genético para recomendação de objetos de aprendizagem sensível ao contexto do estudante. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 23., 2012, Rio de Janeiro. **Anais...** Rio de Janeiro: Sociedade Brasileira de Computação, 2012.
- KLAŠNJA-MILIĆEVIĆ, A.; VESIN, B.; IVANOVIĆ, M. Social tagging strategy for enhancing e-learning experience. **Computers & Education**, Elsevier, v. 118, p. 166–181, 2018. Disponível em: <<https://doi.org/10.1016/j.compedu.2017.12.002>>.
- LEARNING TECHNOLOGY STANDARDS COMMITTEE. **IEEE 1484.12.1**: Draft standard for learning object metadata. Nova York, 2002. 44 p.
- LIMONGELLI, C.; GASPARETTI, F.; SCIARRONE, F. Wiki course builder: A system for retrieving and sequencing didactic materials from wikipedia. In: INFORMATION TECHNOLOGY BASED HIGHER EDUCATION AND TRAINING, 14., 2015, Lisbon, Portugal. **Proceedings...** Washington, USA: IEEE Computer Society, 2015. Disponível em: <<https://doi.org/10.1109/ITHET.2015.7218041>>.
- MENOLLI, A.; MALUCELLI, A.; REINEHR, S. Criação semi-automática de objetos de aprendizagem a partir de conteúdos da wiki. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 22., 2011, Aracaju, Sergipe. **Anais...** Porto Alegre, RS: Sociedade Brasileira de Computação, 2011. p. 214–223.
- NASCIMENTO, P. et al. Recomendação de objetos de aprendizagem baseados em modelos de estilos de aprendizagem: Uma revisão sistemática da literatura. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 28., 2017, Recife, Pernambuco. **Anais...** Porto Alegre, RS: Sociedade Brasileira de Computação, 2017. p. 213–222.
- NILASHI, M.; IBRAHIM, O.; ITHNIN, N. Hybrid recommendation approaches for multi-criteria collaborative filtering. **Expert Systems with Applications**, Elsevier, v. 41, n. 8, p. 3879–3900, 2014. Disponível em: <<https://doi.org/10.1016/j.eswa.2013.12.023>>.
- NOY, N. F.; MCGUINNESS, D. L. **Ontology Development 101**: A guide to creating your first ontology. Stanford, USA, 2001. (Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880).
- OUF, S. et al. A proposed paradigm for smart learning environment based on semantic web. **Computers in Human Behavior**, Elsevier, v. 72, p. 796–818, 2017. Disponível em: <<https://doi.org/10.1016/j.chb.2016.08.030>>.
- PASHLER, H. et al. Learning styles: Concepts and evidence. **Psychological science in the public interest**, SAGE Publications Sage CA: Los Angeles, CA, v. 9, n. 3, p. 105–119, 2008.
- PEREIRA, C. K. et al. Broad-rsi-educational recommender system using social networks interactions and linked data. **Journal of Internet Services and Applications**, SpringerOpen, v. 9, n. 7, p. 1–28, 2018.

- RAMIREZ-ARELLANO, A.; BORY-REYES, J.; HERNÁNDEZ-SIMÓN, L. M. Learning object retrieval and aggregation based on learning styles. **Journal of Educational Computing Research**, SAGE Publications Sage CA: Los Angeles, CA, v. 55, n. 6, p. 757–788, 2017. Disponível em: <<https://doi.org/10.1177/0735633116681303>>.
- SOLOMAN, B. A.; FELDER, R. M. **Index of learning styles questionnaire**. 2005. NC State University. Disponível em: <<https://www.webtools.ncsu.edu/learningstyles/>>.
- TARUS, J.; NIU, Z.; KHADIDJA, B. E-learning recommender system based on collaborative filtering and ontology. **International Scholarly and Scientific Research & Innovation**, World Academy of Science, Engineering and Technology, v. 11, n. 2, p. 256–261, 2017.
- TARUS, J. K.; NIU, Z.; KALUI, D. A hybrid recommender system for e-learning based on context awareness and sequential pattern mining. **Soft Computing**, Springer, v. 22, n. 8, p. 2449–2461, 2018. Disponível em: <<https://doi.org/10.1007/s00500-017-2720-6>>.
- VICCARI, R. et al. The obaa proposal for learning objects supported by agents. In: MASEIE WORKSHOP–AUTONOMOUS AGENTS AND MULTI-AGENT SYSTEMS (AAMAS), 9., 2010, Toronto, Canadá. **Proceedings...** Toronto, Canadá: IFAAMAS, 2010.
- WARNCKE-WANG, M.; COSLEY, D.; RIEDL, J. Tell me more: An actionable quality model for wikipedia. In: INTERNATIONAL SYMPOSIUM ON OPEN COLLABORATION, 9., 2013, Hong Kong, China. **Proceedings...** New York: ACM, 2013. Disponível em: <<https://doi.org/10.1145/2491055.2491063>>.
- WHITLEY, D. A genetic algorithm tutorial. **Statistics and computing**, Springer, v. 4, n. 2, p. 65–85, 1994.
- XIAO, J. et al. A personalized recommendation system with combinational algorithm for online learning. **Journal of Ambient Intelligence and Humanized Computing**, Springer, v. 9, n. 3, p. 667–677, 2018.

Apêndices

Regras SWRL implementadas na ontologia

Enumeram-se, a seguir, todas as 32 regras SWRL implementadas na ontologia do SCROA.

1. $\text{Student}(\text{?student}), \text{IdealLOs}(\text{?idealLO}), \text{hasState}(\text{?idealLO}, \text{activeIdealLO}), \text{isRecommendedFor}(\text{?idealLO}, \text{?student}), \text{Profile}(\text{?profile}), \text{hasProfile}(\text{?student}, \text{?profile}), \text{hasProcessing}(\text{?profile}, \text{activeProcessing}) \rightarrow \text{ListResourcesIdealLO}(\text{animation})$
2. $\text{Student}(\text{?student}), \text{IdealLOs}(\text{?idealLO}), \text{hasState}(\text{?idealLO}, \text{activeIdealLO}), \text{isRecommendedFor}(\text{?idealLO}, \text{?student}), \text{hasProfile}(\text{?student}, \text{?profile}), \text{hasPerception}(\text{?profile}, \text{sensing}) \rightarrow \text{ListResourcesIdealLO}(\text{animation})$
3. $\text{Student}(\text{?student}), \text{IdealLOs}(\text{?idealLO}), \text{hasState}(\text{?idealLO}, \text{activeIdealLO}), \text{isRecommendedFor}(\text{?idealLO}, \text{?student}), \text{Profile}(\text{?profile}), \text{hasProfile}(\text{?student}, \text{?profile}), \text{hasProcessing}(\text{?profile}, \text{reflective}) \rightarrow \text{ListResourcesIdealLO}(\text{reflectionQuiz})$
4. $\text{Student}(\text{?student}), \text{IdealLOs}(\text{?idealLO}), \text{hasState}(\text{?idealLO}, \text{activeIdealLO}), \text{isRecommendedFor}(\text{?idealLO}, \text{?student}), \text{Profile}(\text{?profile}), \text{hasProfile}(\text{?student}, \text{?profile}), \text{hasPerception}(\text{?profile}, \text{intuitive}) \rightarrow \text{ListResourcesIdealLO}(\text{additionalResource})$
5. $\text{Student}(\text{?student}), \text{IdealLOs}(\text{?idealLO}), \text{hasState}(\text{?idealLO}, \text{activeIdealLO}), \text{isRecommendedFor}(\text{?idealLO}, \text{?student}), \text{Profile}(\text{?profile}), \text{hasProfile}(\text{?student}, \text{?profile}), \text{hasProcessing}(\text{?profile}, \text{reflective}) \rightarrow \text{ListResourcesIdealLO}(\text{example})$
6. $\text{Student}(\text{?student}), \text{IdealLOs}(\text{?idealLO}), \text{hasState}(\text{?idealLO}, \text{activeIdealLO}), \text{isRecommendedFor}(\text{?idealLO}, \text{?student}), \text{Profile}(\text{?profile}), \text{hasProfile}(\text{?student}, \text{?profile}), \text{hasUnderstanding}(\text{?profile}, \text{sequential}) \rightarrow \text{ListResourcesIdealLO}(\text{additionalResource})$
7. $\text{Student}(\text{?student}), \text{IdealLOs}(\text{?idealLO}), \text{hasState}(\text{?idealLO}, \text{activeIdealLO}), \text{isRecommendedFor}(\text{?idealLO}, \text{?student}), \text{Profile}(\text{?profile}), \text{hasProfile}(\text{?student}, \text{?profile}), \text{hasProcessing}(\text{?profile}, \text{activeProcessing}) \rightarrow \text{ListResourcesIdealLO}(\text{selfAssessment})$

8. Student(?student), IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), isRecommendedFor(?idealLO, ?student), Profile(?profile), hasProfile(?student, ?profile), hasInput(?profile, verbal) \rightarrow ListResourcesIdealLO(additionalResource)
9. Student(?student), IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), isRecommendedFor(?idealLO, ?student), Profile(?profile), hasProfile(?student, ?profile), hasProcessing(?profile, activeProcessing) \rightarrow ListResourcesIdealLO(forumActivity)
10. Student(?student), IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), isRecommendedFor(?idealLO, ?student), Profile(?profile), hasProfile(?student, ?profile), hasPerception(?profile, sensing) \rightarrow ListResourcesIdealLO(realLifeApplication)
11. Student(?student), IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), isRecommendedFor(?idealLO, ?student), Profile(?profile), hasProfile(?student, ?profile), hasProcessing(?profile, reflective) \rightarrow ListResourcesIdealLO(additionalResource)
12. Student(?student), IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), isRecommendedFor(?idealLO, ?student), Profile(?profile), hasProfile(?student, ?profile), hasInput(?profile, verbal) \rightarrow ListResourcesIdealLO(forumActivity)
13. Student(?student), IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), isRecommendedFor(?idealLO, ?student), Profile(?profile), hasProfile(?student, ?profile), hasProcessing(?profile, activeProcessing) \rightarrow ListResourcesIdealLO(exercise)
14. Student(?student), IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), isRecommendedFor(?idealLO, ?student), Profile(?profile), hasProfile(?student, ?profile), hasPerception(?profile, sensing) \rightarrow ListResourcesIdealLO(exercise)
15. Student(?student), IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), isRecommendedFor(?idealLO, ?student), Profile(?profile), hasProfile(?student, ?profile), hasPerception(?profile, sensing) \rightarrow ListResourcesIdealLO(example)
16. Student(?student), IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), isRecommendedFor(?idealLO, ?student), Profile(?profile), hasProfile(?student, ?profile), hasUnderstanding(?profile, sequential) \rightarrow ListResourcesIdealLO(exercise)
17. Student(?student), IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), isRecommendedFor(?idealLO, ?student), Profile(?profile), hasProfile(?student, ?profile), hasPerception(?profile, intuitive) \rightarrow ListResourcesIdealLO(exercise)
18. Student(?student), IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), isRecommendedFor(?idealLO, ?student), Profile(?profile), hasProfile(?student, ?profile), hasUnderstanding(?profile, sequential) \rightarrow ListResourcesIdealLO(selfAssessment)
19. Student(?student), IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), isRecommendedFor(?idealLO, ?student), Profile(?profile), hasProfile(?student, ?profile), hasUnderstanding(?profile, global) \rightarrow ListResourcesIdealLO(realLifeApplication)
20. Student(?student), IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), isRecommendedFor(?idealLO, ?student), Profile(?profile), hasProfile(?student, ?profile), hasPerception(?profile, intuitive) \rightarrow ListResourcesIdealLO(reflectionQuiz)

21. Student(?student), IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), isRecommendedFor(?idealLO, ?student), Profile(?profile), hasProfile(?student, ?profile), hasUnderstanding(?profile, global) \rightarrow ListResourcesIdealLO(example)
22. Student(?student), IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), isRecommendedFor(?idealLO, ?student), Profile(?profile), hasProfile(?student, ?profile), hasInput(?profile, visual) \rightarrow ListResourcesIdealLO(animation)
23. Student(?student), IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), isRecommendedFor(?idealLO, ?student), Profile(?profile), hasProfile(?student, ?profile), hasUnderstanding(?profile, sequential) \rightarrow ListResourcesIdealLO(reflectionQuiz)
24. Student(?student), IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), isRecommendedFor(?idealLO, ?student), Profile(?profile), hasProfile(?student, ?profile), hasUnderstanding(?profile, sequential) \rightarrow ListResourcesIdealLO(animation)
25. IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), ListResourcesIdealLO(?res_type), PermanentLOs(?lo), Educational_5(?edu), hasEducationalData(?lo, ?edu), hasLearningResourceType(?edu, ?res_type) \rightarrow SuggestedLOs(?lo)
26. IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), General_1(?gen_ideal), hasGeneralData(?idealLO, ?gen_ideal), hasKeyword(?gen_ideal, ?keyword), PermanentLOs(?lo), General_1(?gen), hasGeneralData(?lo, ?gen), hasKeyword(?gen, ?keyword) \rightarrow SuggestedLOs(?lo)
27. IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), General_1(?gen_ideal), hasGeneralData(?idealLO, ?gen_ideal), hasTitle(?gen_ideal, ?title), PermanentLOs(?lo), General_1(?gen), hasGeneralData(?lo, ?gen), hasTitle(?gen, ?title) \rightarrow SuggestedLOs(?lo)
28. IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), Educational_5(?edu_ideal), hasEducationalData(?idealLO, ?edu_ideal), hasSemanticDensity(?edu_ideal, ?sem_den), PermanentLOs(?lo), Educational_5(?edu), hasEducationalData(?lo, ?edu), hasSemanticDensity(?edu, ?sem_den) \rightarrow SuggestedLOs(?lo)
29. IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), Educational_5(?edu_ideal), hasEducationalData(?idealLO, ?edu_ideal), hasDifficulty(?edu_ideal, ?diff), PermanentLOs(?lo), Educational_5(?edu), hasEducationalData(?lo, ?edu), hasDifficulty(?edu, ?diff) \rightarrow SuggestedLOs(?lo)
30. IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), Educational_5(?edu_ideal), hasEducationalData(?idealLO, ?edu_ideal), hasInteractivityLevel(?edu_ideal, ?int_lev), PermanentLOs(?lo), Educational_5(?edu), hasEducationalData(?lo, ?edu), hasInteractivityLevel(?edu, ?int_lev) \rightarrow SuggestedLOs(?lo)
31. IdealLOs(?idealLO), hasState(?idealLO, activeIdealLO), Educational_5(?edu_ideal), hasEducationalData(?idealLO, ?edu_ideal), hasInteractivityType(?edu_ideal, ?int_type),

- PermanentLOs(?lo), Educational_5(?edu), hasEducationalData(?lo, ?edu), hasInteractivityType(?edu, ?int_type) \rightarrow SuggestedLOs(?lo)
32. TemporaryLOs(?tempLO), General_1(?gen_temp), hasGeneralData(?tempLO, ?gen_temp), Identifier(?ide_temp), hasIdentifier(?gen_temp, ?ide_temp), hasEntry__(?ide_temp, ?uri), PermanentLOs(?lo), General_1(?gen), hasGeneralData(?lo, ?gen), Identifier(?ide), hasIdentifier(?gen, ?ide), hasEntry__(?ide, ?uri) \rightarrow SuggestedLOs(?lo)