

**CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**XML2AIML: APLICAÇÃO DE MOTORES ROBÓTICOS
INTELIGENTES UTILIZANDO PROCESSAMENTO DE LINGUAGEM
NATURAL INTEGRADOS COM REPOSITÓRIO INSTITUCIONAL**

RAFAEL LUIZ DE MACEDO

Marília, 2013

**CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**XML2AIML: APLICAÇÃO DE MOTORES ROBÓTICOS
INTELIGENTES UTILIZANDO PROCESSAMENTO DE LINGUAGEM
NATURAL INTEGRADOS COM REPOSITÓRIO INSTITUCIONAL**

Monografia apresentada ao
Centro Universitário Eurípides de
Marília como parte dos requisitos
necessários para a obtenção do
grau de Bacharel em Ciência da
Computação
Orientador: Prof. Dr. Elvis Fusco

Marília, 2013



CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO – AVALIAÇÃO FINAL

Rafael Luiz de Macedo

**IMPLEMENTAÇÃO DE MOTORES ROBÓTICOS INTELIGENTES UTILIZANDO
PROCESSAMENTO DE LINGUAGEM NATURAL E INTEGRADOS COM A PLATAFORMA
DSPACE**

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da
Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da
Computação.

Nota: 10 (Def)

Orientador: Elvis Fusco

1º. Examinador: Rodolfo Barros Chiamonte

2º. Examinador: Mauricio Duarte

Marília, 04 de dezembro de 2013.

“Seja quem você for, seja qualquer posição que você tenha na vida, nível altíssimo ou mais baixo social, tenha sempre como meta muita força, muita determinação e sempre faça tudo com muito amor e com muita fé em Deus, que um dia você chega lá. De alguma maneira você chega lá.” (Ayrton Senna).

Em memória ao meu pai, José Carlos de Macedo Junior.

Em memória da minha avó Neide Castanho de Macedo.

A minha mãe Marli de Fátima Luiz de Macedo, por sempre estar ao meu lado me apoiando nas minhas decisões.

A minha irmã Taisa Cristina de Macedo, pela ajuda e apoio durante estes quatro anos de graduação.

Aos meus fiadores José dos Santos Coutinho e Fátima Maria de Oliveira Coutinho, pela confiança que depositaram em mim.

AGRADECIMENTOS

Primeiramente, agradeço ao nosso senhor Deus, pelo dom da vida e por esta conquista.

Dedico esta monografia em memória do meu melhor amigo, meu herói, meu pai, José Carlos de Macedo Junior.

Em memória da pessoa que me deu de presente a oportunidade de estar cursando um ensino superior, minha avó Neide Castanho de Macedo.

A minha rainha, minha mãe Marli de Fátima Luiz de Macedo, por estar sempre ao meu lado, ajudando nos momentos que mais precisava de uma orientação.

Agradeço a minha irmã Taisa Cristina de Macedo, sem você eu não estaria escrevendo essa monografia, obrigado por estar sempre ao meu lado.

Agradeço imensamente aos meus fiadores, José dos Santos Coutinho e Fátima Maria de Oliveira Coutinho, obrigado pelo voto de confiança e acreditarem na minha capacidade.

Agradeço ao meu orientador, chefe, patrão, professor Dr. Elvis Fusco, por me apoiar durante estes quatro anos nas salas de aula, durante os dois anos de estágio, de iniciação científica e trabalho de conclusão de curso.

Agradeço aos meus amigos, ao Allan Mariano de Souza por ter dado o nome do meu software XML2AIML, e também ao Rodolfo Rosa Miranda e o Matheus Pellegrine Fernandes, a amizade de vocês levarei por toda a minha vida.

Agradeço aos meus queridos professores Danielle, Elton, Fábio Dacêncio, Fábio Lucio, Ildeberto Bugatti, José Celso, Juliana, Jussara, Leonardo, Mauricio, Paulo Rogerio, Paulo Nardi, Renata, Ricardo (pela ajuda na escrita da minha monografia) e Rodolfo (espero não ter esquecido ninguém). Obrigado pelos ensinamentos passados durante as aulas.

Agradeço especialmente aos professores que se tornaram amigos, Dr. Elvis Fusco, Dr. Fábio Dacêncio Pereira, Ms. Leonardo Castro Botega e Ms. Rodolfo Barros Chiaramonte, por me darem a oportunidade de estagiar no laboratório de pesquisa COMPSI, graças ao estágio realizei o sonho meu e dos meus pais, que é de ver o filho se formando no ensino superior.

Obrigado pela confiança, pelo enorme conhecimento que vocês me passaram durante estes dois anos de estágio, pelas responsabilidades que deixaram na minha mão, por me enviar para um evento em São Paulo, que foi excelente a experiência e outros eventos que participamos.

Agradeço ao professor Dr. Fábio Dacêncio Pereira, por estar sempre me ajudando com oportunidades em participar nos projetos, hoje me formo com uma bagagem grande no currículo.

Agradeço ao professor Ms. Leonardo Castro Botega, pela ajuda em levar a mesa tangível nos eventos, que não é fácil desmontar a mesa, obrigado por me ensinar e me ajudar nas horas que precisei de orientação.

(Esta conquista não é só minha, é sua mãe, do pai, da minha irmã, dos professores e de todos que me ajudaram).

Rafael Luiz de Macedo

RESUMO

O AIML (*Artificial Intelligence Markup Language*) é um conjunto de elementos (TAG) XML (eXtensible Markup Language) capaz de representar e relacionar expressões em linguagem natural permitindo a criação de motores robóticos inteligentes capazes de manter um diálogo simples, porém, os motores robóticos são limitados na quantidade de perguntas em que possam responder, assim, não conseguindo manter um diálogo simples por muito tempo. Esse trabalho tem o objetivo de estender a linguagem AIML na incorporação de novos elementos padrões e personalizados utilizando um interpretador Java para realizar buscas Web semântica em repositório DSpace e com a integração de um software analisador de código-fonte desenvolvido neste trabalho, aplicará um tratamento nas informações armazenadas no repositório gerando novas perguntas e respostas para os motores robóticos desenvolvidos na linguagem AIML.

Palavra chave: AIML; Inteligência Artificial; Motores Robóticos; Web Semântica; Repositório; DSpace;

ABSTRACT

The Artificial Intelligence Markup Language (AIML) is a set of elements (TAG) eXtensible Markup Language (XML) to represent and relate expressions in natural language allowing the creation of intelligent robotic engines capable of maintaining a simple dialog, however, robotic engines are limited in the amount of questions they may respond, thus, failing to keep a simple dialogue for a long time. This work aims to extend the AIML language in the incorporation of new elements and custom patterns using a Java interpreter to search Semantic Web in DSpace repository and with the integration of a source code Analyzer software developed in this work, apply a treatment on the information stored in the repository generating new questions and answers to the robotic engines developed in AIML language.

Keywords: AIML; Artificial Intelligence; Engines Robotic; Semantic Web; Repository; DSpace;

LISTA DE ILUSTRAÇÕES

Figura 1: Registro do repositório digital Univem Aberto do Centro Universitário Eurípides de Marília.	24
Figura 2: Repositório digital Univem Aberto.	24
Figura 3: Modelo de funcionalidade de uma busca de arquivo em repositório digital por meio de um chatterbot AIML.	34
Figura 4: Resposta XML vinda do servidor do repositório Univem Aberto.	38
Figura 5: Metadados contidos no XML.....	40
Figura 6: Nova base de conhecimento gerado pelo software XML2AIML.	41

LISTA DE ABREVIATURAS E SIGLAS

AIML	Artificial Intelligence Markup Language
DC	Dublin Core
PLN	Processamento de Linguagem Natural
TAG	Elemento de Linguagem de Marcação
XML2AIML	Software Analisador de Metadados e AIML
XML	eXtensible Markup Language

SUMÁRIO

AGRADECIMENTOS	6
RESUMO	8
ABSTRACT	9
LISTA DE ILUSTRAÇÕES	10
LISTA DE ABREVIATURAS E SIGLAS	11
INTRODUÇÃO.....	14
CAPITULO 1 – AIML	16
1.1. Estruturas do Arquivo AIML.....	18
1.2. Reconhecimento de palavra.....	20
1.3. Processamento de Linguagem Natural	21
CAPITULO 2 – REPOSITÓRIOS DIGITAIS	23
CAPITULO 3 – METADADOS.....	27
3.1. Dublin Core	29
CAPITULO 4 – XML2AIML	33
4.1. Software XML2AIML.....	33
4.2. Coletando dados do usuário.....	36
4.3. Criando URL para requisição de busca de arquivo	37
4.4. Pergunta e Resposta Pré-Definidas.....	39
4.5. Criando nova base de conhecimento	39
4.6. Outros cenários de teste.....	41
CONCLUSÃO.....	43
TRABALHOS FUTUROS	44
PRODUÇÃO BIBLIOGRÁFICA	45
Trabalhos completos publicados em anais de congresso.....	45

Resumos expandidos publicados em anais de congressos.....	46
Resumos publicados em anais de congressos.....	46
REFERÊNCIAS BIBLIOGRÁFICAS	47
Apêndice A	50
Apêndice B	52
Apêndice C	52
Apêndice D	53
Apêndice E	54
Apêndice F.....	56
Apêndice G	57

INTRODUÇÃO

As tecnologias atuais têm permitido cada vez mais a percepção que é possível fazer uma máquina ser inteligente a ponto de responder perguntas feitas por meio de motores robóticos construídos com inteligência artificial.

A linguagem destina-se à comunicação sobre o mundo. Quando se estuda a linguagem, passa-se a conhecer mais o mundo. Podem-se testar teorias sobre o mundo se conhecem que sustentação elas dão à tentativa de compreender a linguagem. E, se conseguirmos criar um modelo computacional de linguagem, pode-se ter uma ferramenta poderosa para a comunicação sobre o mundo. (RICH, Elaine e KNIGHT, Kevin, 1993, p.433).

Por um lado, as formas em que os seres humanos e computadores normalmente comunicam são muito diferentes. Os seres humanos tendem há gastar se muito tempo em bate-papos e diálogo informal com pouco ou nenhum efeito. Computadores são conhecidos por dar respostas precisas, verdadeiras e lógicas. A taxa de troca de informação de mais diálogo humano é muito baixa, não mais do que 1kbit por segundo, mas a comunicação de computador é mais rápida. Alicebot / AIML é uma tentativa de moldar esta divisão. (WALLACE, Dr. Richard S., 2003, “The Element of AIML Style”).

Uma das linguagens que possibilita a criação e desenvolvimento desses motores robóticos inteligentes capazes de se manter em um diálogo é a linguagem de marcação de inteligência artificial AIML.

Os motores robóticos desenvolvidos na linguagem AIML têm uma grande limitação na quantidade de perguntas que possam responder, por principal motivo de ser desenvolvido em cima de um ou mais contextos e não podendo abranger dinamicamente a quantidade de perguntas e respostas em seu banco de informação.

Com o surgimento da *Internet*, as informações estão cada vez mais acessíveis, de forma que, apenas com uma máquina que tenha acesso a rede *Internet* é possível realizar buscas para obter novas informações.

A quantidade de informação que são disponibilizadas diariamente na Web tem facilitado cada vez mais o acesso para as pessoas. Consequentemente tem crescido o número de repositórios online que possuem artigos, monografias, jornais, revistas e outros tipos de arquivos digitais. Uma plataforma que disponibiliza esse tipo de repositório é o DSpace, que é bastante utilizado por ser uma plataforma open source e ter várias ferramentas disponíveis a

ser utilizado no repositório, como ferramenta de busca, acesso aos arquivos por meio de requisições de buscas aos servidores *Web*.

O principal objetivo desse trabalho é desenvolver um software analisador de metadados dos repositórios digitais da plataforma DSpace, que realizará buscas de arquivos digitais nos metadados do repositório para encontrar os arquivos relacionados aos assuntos que os usuários estão buscando. O software analisador gerará novas bases de conhecimentos de perguntas e respostas para os motores robóticos em cima desses metadados do repositório, para fins, de solucionar a limitação dos motores robóticos em apenas responder perguntas que estão contidas no contexto que foram desenvolvidos.

Para o desenvolvimento do software analisador de metadados será utilizado o repositório digital do Centro Universitário Eurípides de Marília – UNIVEM denominado Univem Aberto (Univem Aberto, <http://aberto.univem.edu.br/>, Novembro 2013), por o repositório Univem Aberto ser da plataforma DSpace.

Estruturalmente esta monografia é composta por cinco capítulos.

O primeiro capítulo apresenta a história da linguagem de marcação de inteligência artificial AIML, apresentando a estrutura de um arquivo AIML em que os motores robóticos fazem uso para poderem reconhecer perguntas e responde-las.

O segundo capítulo trata os principais motivos da existência de repositório digital institucional e sobre repositórios digitais da plataforma DSpace.

O terceiro capítulo apresenta o que são metadados, qual o objetivo de utilizar metadados em repositórios digitais e o padrão de metadados que é utilizado no repositório.

O quarto capítulo é o resultado obtido neste trabalho, são descritos como um chatterbot coleta informações do usuário, com as informações coletadas a realização da busca de arquivos no repositório digital Univem Aberto, o tratamento dos metadados vindo do repositório para o software analisador de metadados e a geração das novas bases de conhecimentos de perguntas e respostas AIML.

No último capítulo é tratada a conclusão do trabalho sobre os resultados obtidos.

CAPITULO 1 – AIML

A linguagem AIML é um conjunto de elementos (TAG) XML (eXtensible Markup Language) capaz de representar e relacionar expressões em linguagem natural, permitindo a criação de motores capazes de manter um diálogo simples. Cada conjunto de elementos AIML possui um ou mais elementos chamadas de categoria, sendo que as categorias são desenvolvidas em cima de um contexto; uma categoria é formada pelo conjunto dos elementos *template* e *pattern*, que são os elementos responsáveis por interpretar a mensagem pelo usuário e enviar uma mensagem de resposta ao usuário.

Os *Chatterbots* (*Chat* = Conversa e *bot* = Robô), como conhecidos atualmente são motores robóticos inteligentes que interage com usuários por meio de perguntas e respostas. Atualmente vem crescendo a quantidade de motores robóticos inteligentes utilizados em atendimentos online e por meio de dispositivos de comunicação por voz.

As empresas têm utilizado esses motores robóticos, para realizarem um auto - atendimento ao cliente para fins de melhorarem os atendimentos e economizando tempo, o uso desses motores robóticos é por motivo desses motores fazerem o reconhecimento do assunto que o cliente quer tratar com a empresa, para assim, redirecionar o atendimento ao setor competente. Os motores robóticos são utilizados para atendimentos online e atendimentos por telefonia.

Um motor robótico bastante conhecido e é utilizado para conscientizar as pessoas em cuidarem do meio ambiente é o robô Ed da empresa CONCEPT junto com o apoio da PETROBRAS.

O surgimento da linguagem AIML foi resultado do desenvolvimento do software de PLN conhecida A.L.I.C.E. (*Artificial Linguistic Internet Computer Entity*), a licença do AIML está sob a GNU GPL.

Os motores robóticos desenvolvidos na linguagem AIML, possuem dois módulos, linguagem e motor. A linguagem é todo o conhecimento que esses motores possuem, da linguagem natural e das informações desenvolvidas na linguagem AIML. O motor realiza a comunicação entre as duas linguagens, natural e AIML, ocorrendo o reconhecimento das informações contidas nas duas linguagens, para assim, os motores robóticos poderem reconhecer perguntas escritas feitas pelos humanos e responder essas perguntas.

Existe uma limitação nos módulos dos motores robóticos, os mesmos não

conseguem reconhecer perguntas realizadas fora do contexto em que foram desenvolvidos, por não poder reconhecer perguntas de outros contextos que não tenha em seu banco de conhecimento, às conversas não durão muito tempo.

Alan Turing (1912-1954), em seu famoso ensaio “*Computing Machinery and Intelligence*” (Turing 1950), sugeriu que, em vez de perguntar se as máquinas podem pensar, deve-se perguntar se máquinas podem passar por um teste de inteligência comportamental, que veio a ser chamado de teste de Turing (Stuart Russell e Peter Norvig, 2004, p.916).

O Teste de Turing é realizado por meio de perguntas e respostas feitas para uma máquina com motor robótico inteligente e para um humano, assim como, máquina e o humano, cada um é alocado em uma sala separadamente. Um avaliador realizará as perguntas para essas duas salas, mas é feito as perguntas sem o avaliador saber em qual das salas está a máquina com o motor robótico.

Conforme as respostas que o avaliador vai recebendo, o mesmo poderá descobrir em qual das salas se encontra a máquina e na outra o humano. Os motores robóticos inteligentes que reconhecem a linguagem natural, não conseguem criar novas perguntas e respostas de forma autônoma, fazendo com que ocorra uma perda de não conseguirem reconhecer uma pergunta que está fora do seu conhecimento ou repetirem uma resposta já utilizada em perguntas diferentes.

A máquina que conseguir passar no teste de Turing é considerada inteligente, atualmente não existem máquinas capazes de passar por esse teste.

A linguagem AIML é uma linguagem livre (*open source*), assim, possibilitando o uso em pesquisas de melhoria nas criações dos motores robóticos inteligentes capazes de reconhecer informações escrita na linguagem natural.

Uma linguagem de marcação possui várias plataformas gratuitas e pagas para a utilização da linguagem e plataformas desenvolvidas em linguagens de programação como Java, Python, C, C++ e etc.

A plataforma ProgramD é a mais utilizada quando se fala em desenvolvimento de motores robóticos na linguagem AIML e a mais completa em recurso da linguagem, por motivo da plataforma ser desenvolvida na linguagem Java e ser *open source*.

1.1. Estruturas do Arquivo AIML

A criação de uma base de conhecimentos (Arquivo) da linguagem AIML possui regras que devem ser seguidas para que um motor robótico possa fazer o uso dessa nova base. As regras da linguagem definem os elementos padrões que devem conter um arquivo AIML, que esses elementos são:

- Aiml

São utilizados para definir qual versão do AIML que é utilizado na estrutura do arquivo, quais são os padrões HTML e xsi. Esse elemento é o último a ser finalizado no arquivo, por ser o responsável de agrupar todas as categorias (category) e outros elementos.

- XML

Este elemento é utilizado para definir qual versão do XML que é utilizado na estrutura da base de conhecimento, quais são os padrões do XML.

- Category

A categoria é o conjunto de um pattern e template, que representam uma pergunta e resposta. A base de conhecimento AIML pode conter um ou mais conjuntos de categorias, que a utilização desses conjuntos pelos motores robóticos é possível se manterem em um dialogo. Uma categoria pode fazer o uso de pattern e template pertencente de outra categoria, para poder fazer a junção de respostas. No apêndice A é apresentando várias formas de category que são utilizados neste trabalho.

- Pattern

É responsável por definir uma pergunta que o motor robótico poderá reconhecer, toda pergunta que o usuário fazer ao robô será verificado pelo motor robótico para encontrar a pattern exatamente igual à pergunta realizada.

- Template

É responsável por definir uma resposta que o motor robótico utilizará quando o usuário realizar uma pergunta, não é necessário apenas utilizar a resposta que está definida em uma única categoria, pode ser usada varias outras categorias, por meio do elemento srail.

- Srail

Toda a resposta definida no elemento template pode fazer o uso do elemento srail, que redireciona pra uma categoria já criada, esse tipo de redirecionamento é utilizado para perguntas com o mesmo intuito. Como por exemplo, perguntar para uma pessoa como ela está

pode ser perguntada de várias maneiras e utilizar uma única resposta, para assim, se evita redundância de informações.

- Set

Este elemento set é utilizado quando é necessário gravar informações vindas na pergunta do usuário, em uma variável ou no banco de dados. Neste trabalho é utilizado para gravar as informações sobre os arquivos que o usuário está buscando no repositório. A palavra ou mensagem digitada pelo usuário é armazenada pelo uso do elemento star ou “*” dentro do elemento pattern e no elemento set é criada a variável, assim, como é mostrado no apêndice B.

- Get

O elemento get realiza a leitura de uma variável já criada pelo elemento set, neste trabalho será utilizado para confirmação das informações que o usuário digitou.

- Star

Se uma mensagem de entrada contiver alguma palavra ou uma mensagem que esteja dentro de uma categoria que esta sendo usado o símbolo “*” ou “_”, no começo ou no fim ou até mesmo no começo e no fim da mensagem, o elemento star faz com que as palavras que esteja antes ou depois da mensagem contida nesta categoria sejam irrelevantes, assim podendo executar essa categoria e fazer a função contida nela. O símbolo “*” representa a <star> que pode ser escrita na forma de tag <star> ou o símbolo “*”.

- Substitutions

É responsável por criar um conjunto de elemento substitute, que é um conjunto de palavras que foram selecionadas por conterem acentos ou “ç” em suas estruturas. O elemento substitutions possui vários outros elementos como input, person e person2, mas neste trabalho não será utilizado os elementos person e person2.

- Substitute

Como a linguagem natural utilizada para a criação da linguagem de marcação é a linguagem universal Inglês, os motores robóticos possuem uma limitação em reconhecer palavras de outras linguagens naturais, como palavras com acentos e “ç” que são da linguagem natural português.

O elemento substitute é responsável pelo reconhecimento de palavras com acentos e “ç” que serão digitadas pelos usuários do repositório, durante o desenvolvimento deste trabalho foi encontrado um grande problema por não conseguir fazer um motor robótico

reconhecer essas palavras, com as pesquisas em trabalhos correlatos foi encontrado uma solução que é o uso desse elemento substitute.

- Input

O elemento input tem a principal de função de reconhecer o que o usuário do repositório está digitando, para que seja possível pegar cada palavra (TOKEN como é definido na análise léxica do PLN) e utilizar o elemento substitute.

Na linguagem AIML existem vários outros elementos que podem ser utilizados na estrutura da base de conhecimento, foram apresentados os elementos utilizados neste trabalho para desenvolver as bases de conhecimento de forma autônoma pelo software analisador de metadados XML2AIML.

A linguagem AIML possui uma grande limitação na quantidade de perguntas e respostas em que os motores robóticos possam estar reconhecendo e respondendo, por motivo dos mesmos motores não serem capazes de abranger seus conhecimentos de forma autônoma. Por esse motivo, neste trabalho é apresentado um meio de abranger o conhecimento desses motores robóticos com novas bases de conhecimentos gerados de forma autônoma por um software analisador de metadados XML2AIML.

Este software analisador fará o uso das informações dos arquivos que estão submetidos nos repositórios digitais da plataforma DSpace, para assim, gerar as novas bases de conhecimentos para os motores robóticos.

1.2. Reconhecimento de palavra.

Os motores robóticos desenvolvidos na linguagem AIML fazem o reconhecimento de palavras da linguagem natural que foi construído a linguagem de marcação AIML, ou seja, reconhecem palavras da linguagem natural inglês. Por esse motivo, ocorre o problema de que quando são desenvolvidos esses motores utilizando uma linguagem natural diferente da linguagem que reconhecem, não conseguem fazer um reconhecimento em determinadas palavras que não pertence à linguagem universal.

A linguagem natural que está sendo utilizada para estrutura dos motores robóticos utilizados neste trabalho é a língua portuguesa, porém, os motores não reconhecem palavras pertencentes a esta língua que possuem acentos ou “ç”.

Com as novas regras ortográficas algumas palavras da linguagem não possuem

mais acentos ou hífen (Educação, “Reforma Ortográfica: Conheça as novas regras”). As novas regras ortográficas diminuíram as palavras que devem ser tratadas quando os motores robóticos realizarem o reconhecimento de palavra.

Uma solução para o reconhecimento das palavras com acentos e “ç” é o uso do elemento substitutions, esse elemento faz o reconhecimento da palavra original digitada pelo usuário do repositório e substitui por outra palavra que pertence ao mesmo significado da palavra original. Como no exemplo abaixo, o motor robótico recebe a pergunta digitada pelo usuário do repositório já com a palavra substituído.

Mensagem de como o usuário digitou.

Usuário do repositório: Você tem algum artigo AIML?

Mensagem após receber o tratamento: Voce tem algum artigo AIML?

No apêndice C é apresentado o arquivo substitutions criado neste trabalho para o reconhecimento das possíveis palavras com acentos e “ç” que os usuários do repositório digital possam estar utilizando. Os principais motivos de estar criando em um arquivo separado das bases de conhecimento são de concentrar em um único lugar todas as palavras a serem tratadas e de facilitar futuras correções que possa estar ocorrendo para a melhoria dos motores robóticos.

1.3. Processamento de Linguagem Natural

O PLN é o estudo da linguagem, sendo linguagem falada ou linguagem de texto escrito. O estudo da linguagem nos possibilita uma melhor compreensão sobre a linguagem e, se as máquinas compreendem a linguagem, é possível criar motores robóticos inteligentes capazes de reconhecer e compreender informações em forma de perguntas e respostas na linguagem natural.

A linguagem escrita faz o uso de conhecimentos léxicos, sintáticos e semânticos da linguagem e também de todas as informações necessárias sobre o mundo real. (RICH, Elaine e KNIGHT, Kevin, 1993, p.434).

O PLN é dividido em vários componentes, sendo que cada componente tem sua função no reconhecimento da linguagem natural. Os componentes, Análise Morfológica, Análise Sintática, Análise Semântica, Integração de Discurso e Análise Pragmática, são o

grupo de componentes que compõem o corpo do PLN.

O software analisador de metadados desenvolvido neste trabalho utiliza técnicas de reconhecimento léxicas e sintáticas para realizar o reconhecimento de cada token (palavra) e a forma sintática em que esses tokens estão posicionados corretamente.

Os motores robóticos desenvolvidos na linguagem AIML fazem o uso dessas PLN durante o reconhecimento de uma pergunta feita pelo usuário e durante a leitura de uma base de conhecimento para encontrar uma possível resposta.

Todas as bases de conhecimentos dos motores robóticos aplicam técnicas de reconhecimento de palavra de PLN, como visto nesta monografia os motores robóticos não conseguem reconhecer palavras de outras línguas naturais que possuem caracteres fora da linguagem natural que foram desenvolvidos o inglês.

Neste trabalho será desenvolvido um software analisador que fará o uso das regras técnicas do PLN, para fins de reconhecer informações que estão contidas nos metadados do repositório DSpace.

No próximo capítulo são descritos as funcionalidades dos repositórios digitais institucionais e quais os motivos das instituições estarem aderindo os repositórios.

CAPITULO 2 – REPOSITÓRIOS DIGITAIS

Com a necessidade de divulgar os trabalhos produzidos por instituições, sem depender de uma editora para divulgar os trabalhos, as instituições começaram a divulgar na *internet* por conta própria e com o passar, foram surgindo várias ferramentas chamadas de repositórios digitais.

Os primeiros repositórios digitais voltados para as instituições começaram a serem desenvolvidos 2002, com os repositórios digitais as instituições começavam a fazer o papel de editora.

Os repositórios digitais têm as principais funções de armazenamento, disseminação e durabilidade de arquivos digitais, fazendo com que seja fácil o acesso aos arquivos que são submetidos nessas ferramentas.

As instituições têm usados os repositórios de plataformas diferentes, divulgando seus periódicos científicos, monografias, teses e outros trabalhos realizados nas instituições, com o passar dos 11 anos, depois do desenvolvimento dos primeiros repositórios, o repositório digital muito utilizado é da plataforma DSpace.

O DSpace é uma plataforma de software Open Source, responsável pelo armazenamento, disseminação e durabilidade de arquivos digitais. Foi criado pelo MIT (*Massachusetts Institute of Technology*) e *Hewlett-Packard*. O MIT e o *Hewlett-Packard* trabalharam durante dois anos para o desenvolvimento da plataforma de repositório digital DSpace (TANSLEY, Robert, et al.).

A plataforma DSpace é voltada para a área acadêmica, com fins de servir como base para o desenvolvimento futuro de abordar a preservação em longo prazo dos arquivos e problemas de acesso (TANSLEY, Robert, et al.).

No site de registro de provedores de repositórios OPEN ARCHIVES é apresentado uma tabela com informações dos repositórios digitais registrados, atualmente são 2140 repositórios registrados e um registro é o do Centro Universitário Eurípides de Marília, o repositório Univem Aberto. Conforme a figura 1 é apresentada uma parte da tabela disponível no Open Archive, que é mostrado o nome do repositório e a url utilizada para requisições de buscas de arquivos, em destaque está o registro do repositório digital Univem Aberto. (<http://www.openarchives.org/Register/BrowseSites>, Novembro 2013).

Record	Identify	Univ Provence repository	http://orioaiup.univ-provence.fr/ori-oai-repository/OAIHandle
Record	Identify	Univem Aberto	http://aberto.univem.edu.br/oai/request
Record	Identify	Universidad Autonoma del Estado de Hidalgo -Biblioteca Digital-	http://dgsa.uaeh.edu.mx:8080/oai/request

Figura 1: Registro do repositório digital Univem Aberto do Centro Universitário Eurípides de Marília.

[Fonte: Open Archive, <http://www.openarchives.org/Register/BrowseSites>, Novembro 2013].

O Centro Universitário Eurípides de Marília – UNIVEM tem implantado deste o ano de 2012 o repositório digital da plataforma DSpace para fins de divulgar os trabalhos científicos, monografias, teses de mestrado e doutorado do próprio corpo docente e outros trabalhos realizados pelos seus alunos, docentes e pesquisadores.

Atualmente o repositório digital Univem Aberto contém 630 trabalhos disponibilizados. Na figura 2, apresenta o repositório digital Univem Aberto.

Figura 2: Repositório digital Univem Aberto.

[Fonte: Univem Aberto, <http://aberto.univem.edu.br/>, Novembro 2013]

A universidade do Minho implantou o repositório institucional, para disponibilizar um acervo das publicações científicas produzidas na instituição, o repositório é denominado RepositóriUM. O objetivo do RepositóriUM é armazenar, preservar, divulgar e dar acesso à

produção intelectual da Universidade do Minho em formato digital e maximizar a visibilidade, uso e impacto da sua investigação através do Acesso Livre. (RepositorioUM, <http://repositorium.sdum.uminho.pt/>, acessado novembro, 2013).

O Repositório Institucional da UnB é um conjunto de serviços oferecidos pela Biblioteca Central para a gestão e disseminação da produção científica e acadêmica da Universidade de Brasília. Todos os seus conteúdos estão disponíveis publicamente, e por estarem amplamente acessíveis proporcionam maior visibilidade e impacto da produção científica da instituição. (Repositório Institucional da Universidade de Brasília, <http://repositorio.unb.br/>, Novembro 2013).

Integram seu acervo, além das teses e dissertações defendidas na Universidade de Brasília, artigos científicos, livros eletrônicos, capítulos de livros e trabalhos apresentados em eventos pelos professores e pesquisadores.

Para disponibilizar sua produção científica no Repositório, os professores, pesquisadores e alunos de pós-graduação da UnB devem preencher e assinar um termo de autorização e entregá-lo no setor de Gerenciamento da Informação Digital (GID), 2º andar da BCE, juntamente com o arquivo do trabalho. Este documento poderá ser assinado, digitalizado e enviado junto com o arquivo por e-mail.

Nos repositórios digitais institucionais, todos os seus conteúdos estão disponíveis publicamente, e por estarem amplamente acessíveis proporcionam maior visibilidade e impacto da produção científica da instituição.

Com a enorme quantidade de arquivos que são disponibilizados nos repositórios digitais institucionais, é necessário padronizar regras para poder gerenciar e identificar cada arquivo que é submetido nos repositórios.

Quando um arquivo é submetido no repositório digital da plataforma DSpace, são necessárias algumas informações sobre o arquivo, por motivo de tratar-se de uma plataforma de organização de arquivos digitais, são necessários inserir informações em determinados campos de metadados.

Todas as informações que foram inseridas nos campos de metadados são utilizadas para catalogar um arquivo no repositório, é registrada em campos denominados metadados, que no repositório digital da plataforma DSpace é utilizado o padrão de metadados Dublin Core (DC). O DC é um conjunto de metadados, que com esse conjunto é possível catalogar todos os arquivos digitais contidos nos repositórios do DSpace.

Para integrar os motores robóticos inteligentes desenvolvidos na linguagem de marcação AIML, serão definidas perguntas com a funcionalidade de obter informações sobre os arquivos que o usuário do repositório esteja buscando. Com essas informações coletadas, os motores robóticos integrados com o software analisador de metadados desenvolvido neste trabalho realizarão requisições de buscas de arquivos no repositório Univem Aberto. O software analisador de metadados denominado de XML2AIML aplicará um tratamento nas informações que estão contidas na resposta vinda do repositório digital Univem Aberto, para, gerar novas bases de conhecimentos aos motores robóticos inteligentes, no formato de perguntas e respostas.

No próximo capítulo é descrito o que são metadados, a necessidade de criar padrões para catalogar arquivos digitais e o que é o padrão de metadados Dublin Core.

CAPITULO 3 – METADADOS

Com o grande aumento de arquivos digitais postados na *Internet*, houve a necessidade de desenvolver padrões que identificam a descrição exata de cada informação dos arquivos, ou seja, desenvolver padrões de Metadados.

Metadados significa dados sobre dados. Os metadados são formas de catalogar todas as informações de um arquivo, da mesma forma que são feitas em bibliotecas reais, cada livro tem um tombo de identificação, são organizados por área e em ordem alfabética pelo título. Os metadados têm a finalidade de documentar e organizar os arquivos digitais de forma estruturada, facilitando a identificação do arquivo por meio de dados padronizados como autor, título, área e resumo.

Metadado é definido aqui como dado que descrevem atributos de um recurso, caracteriza suas relações, possibilita a sua recuperação e uso efetivo e sua existência no ambiente eletrônico. Metadados normalmente consistem em um jogo de elementos de dados onde cada elemento descreve um atributo do recurso, sua administração ou uso. (FUSCO, Elvis, 2010, p.62. VELLUCCI, 1998, p.192).

Segundo Milstead e Feldman (1999, p.3), “todos os motivos pelos quais a indexação e a catalogação são necessárias para fontes impressas se aplicam ainda mais enfaticamente aos metadados para documentos eletrônicos”.

Os repositórios digitais possuem metadados para catalogar todos os arquivos que são submetidos, o padrão utilizado nos repositórios digitais é o padrão Dublin Core que são conjuntos de metadados.

Segundo Weibel, o Dublin Core não tem a intenção de substituir modelos mais ricos com o código AACR2/MARC, mas apenas fornece conjuntos básicos de elementos de descrição, que podem ser usados por catalogadores ou não-catalogadores para simples descrição de recursos de informações.

Com os arquivos catalogados sobre a utilização dos metadados, os mesmos arquivos acabam sendo mais utilizados pelos usuários do que os arquivos que não possuem nenhuma catalogação de metadados, por motivo da facilidade em que os metadados disponibilizam as ferramentas de buscas da *Internet* em encontrar estes arquivos.

O conceito de metadados não é algo novo, mas o uso desse termo em ambientes digitais e a variedade de padrões e as formas de utilização o são. Os registros bibliográficos

que tem sido criado por profissionais da informação há muito tempo devem ser considerados essencialmente como metadados, pois proporcionam informação descritiva e analítica sobre um objeto de informação. (FUSCO, Elvis, 2010).

Logo abaixo são apresentados alguns padrões diferentes de metadados para identificar determinados arquivos:

- Government Information Locator Service (GILS) – informações governamentais.
- Federal Data Geographic Committee (FDDC) – descrição de dados geo-espaciais.
- Machine Readable Card (MARC) – catalogação bibliográfica.
- Dublin Core (DC) – dados sobre páginas da *Web*.
- Consortium for the Interchange of Museum Information (CIMI) – informações sobre Museus.

Novos padrões de metadados podem ser criados de acordo com a necessidade de organização e contribuir para documentar os dados do arquivo digital.

Esses padrões podem ser vistos como padrões de conteúdo dos metadados, padrões de intercâmbio de dados por meio eletrônico e, numa última instância, padrões para modelos de dados. (Ribeiro, 1995).

Os metadados são estruturados por blocos de dados, cada bloco contém informações como autor, título, onde foi publicado e etc., mas uma informação é um campo que pode ser definido por nome do campo, tipo de informação do campo, o formato que é aceito pelo campo e outras descrições que identifica a informação passada neste campo.

A criação de um esquema de metadados deve estabelecer uma estrutura padrão e terminologia. Declarações de rótulos como "criador", "autor", "escultor", ou "compositor" têm pouca representatividade se esses campos, que todos têm a mesma função, não poderem ser mapeados para o mesmo conceito único. Um formulário deverá ser estabelecido, quer através de uma lista de autoridade e assuntos, quer de um vocabulário padrão controlado e então, os relacionamentos precisarão mapear formas alternativas para a forma estabelecida. (FUSCO, Elvis, p.64, 2010).

Nos repositórios digitais, é possível desenvolver softwares capazes de buscar arquivos por meio de requisições feitas aos repositórios, essas requisições fazer o uso dos

conjuntos de metadados DC que são passados pela url de requisição de busca. Em seguida é descrito as funcionalidades dos conjuntos de metadados DC.

3.1. Dublin Core

A plataforma DSpace possibilita ao servidor web do repositório digital receber requisições vindo de meios externos, fora do servidor. Com as requisições é possível ter acesso a todos os arquivos digitais que estão submetidos no repositório.

No apêndice D é apresentada uma url que é utilizada para fazer uma requisição ao servidor web do repositório aberto, que buscará os arquivos de um determinado período. Esse período foi determinado por meio das datas que o usuário passou para o robô AIML desenvolvido nesse trabalho. E no apêndice E é apresentado o resultado XML dessa requisição de busca, vindo do servidor Web.

Quando se realiza uma requisição por url, é utilizado conjuntos de elementos de metadados, ou seja, é usado Dublin Core para definir quais são os parâmetros que serão passados pela url, os valores passados por todos os parâmetros serão utilizados para identificar qual arquivo, grupo de arquivo ou período de submissão do arquivo está sendo buscado no repositório.

A requisição feita ao servidor web é analisada pelo repositório e são buscados todos os arquivos que se encaixarem nas informações passadas pelos DCs, para assim, quando encontrado todos os arquivos é gerado um XML como resposta que é enviado ao agente (software) que realizou a requisição. O software desenvolvido neste trabalho é denominado de XML2AIML, o mesmo ficará responsável por fazer as requisições aos servidores web dos repositórios, que nesse caso é do servidor web do repositório aberto.

Um XML de resposta do repositório contém informações do catalogo do arquivo, como autor (es), título, data de publicação, data em que foi submetido ao repositório, resumo, palavras chaves e entre outras informações que foram utilizadas para catalogar. Essas informações são registradas em campos definidos do DC.

Os DC são utilizados para acrescentarem na plataforma DSpace palavras chaves (metadados) que referenciam as principais informações do catálogo de um arquivo, ou seja, é possível localizar um determinado arquivo por meio de seus próprios metadados.

A plataforma digital DSpace possui vários DC, a seguir são apresentado os DC

padrões que já são criados na instalação da plataforma.

- Título (Title) – Define o título do arquivo digital, quando é realizada uma pesquisa no repositório não é necessário saber como está descrito o título, por meio de palavras-chave é possível localizar o arquivo.
- Assunto e Palavras-chave (Subject) – É descrito as palavras-chaves que referenciam o assunto abordado ou título do arquivo digital.
- Descrição (Description) – Não é um simples resumo, podem conter informações completas do arquivo digital. Por se tratar de um DC rico em informação, esse campo ajuda o usuário que está procurando um arquivo no repositório selecionar quais arquivos podem estar dentro da área de sua pesquisa.
- Tipo (Type) – É o gênero ou tipo do arquivo digital que está sendo submetido no repositório. Com esse DC é possível fazer uma seleção por meio de cada tipo de grupo de arquivo, como arquivos de monografias, jornais, dissertações e etc.
- Fonte (Source) – São referências do arquivo, se for um artigo publicado em um periódico é necessários informações como página inicial e final que o artigo se encontra no periódico e edição do periódico.
- Relação (Relation)
- Cobertura (Coverage)
- Autor (Creator) – É o autor ou são os autores responsáveis pelos direitos do arquivo que está sendo submetido no repositório.
- Editor (Publisher) – Pode ser uma editora responsável pela publicação do arquivo ou até mesmo o próprio autor, mas quando é o autor responsável como editor não é necessário digitar o nome do mesmo neste campo.
- Colaborador (Contributor) – É responsável pela contribuição do conteúdo do arquivo, pode ser um autor, uma universidade ou qualquer tipo de entidade responsável por apoiar o material (arquivo).
- Direitos (Rights) – São os direitos que o autor ou editor tem sobre o

arquivo, que acaba englobando Direitos de Propriedade Intelectual (DPI), direitos autorais e diversos direitos de propriedades.

- Data (Date) – É a data de publicação do arquivo, não é necessário definir o dia, apenas o mês + ano ou somente o ano, a plataforma validará a data.
- Formato (Format) – Pode ser o tipo do arquivo, determinar o equipamento necessário para operar o arquivo, como o software ou hardware.
- Identificador (Identifier) – É um número de identificação do arquivo, pode ser o número ISBN (*International Standard Book Number*) e do DOI (Identificador Digital do Objeto).
- Linguagem (Language) – É a linguagem natural em que o arquivo pertence, como por exemplo, essa monográfica pertence à linguagem natural português.
- Público (Audience) – Todos os arquivos são voltados para um determinado grupo de pessoas, como por exemplo, um arquivo voltado para área de PLN, é voltado para grupo de pessoas que trabalham com reconhecimento de palavras.
- Proveniência (Provenance)
- Títulos de direitos autorais (RightsHolder)
- Método de instrução (Instructional Method)
- Método de competência (Accrual Method)
- Periodicidade de competência (Accrual Periodicity)
- Política de competência (Accrual Policy)

Todos os DC acima estão descritos em português para poder ser explicado a funcionalidade de que cada DC na linguagem natural deste trabalho, e está escrito em inglês por causa das variáveis de parâmetros (DC).

Na plataforma DSpace é possível criar novos DCs direto de uma função que é disponibilizado após a instalação do repositório. Com essa funcionalidade, cada instituição pode definir vários metadados de acordo com a sua necessidade para catalogar os arquivos,

mas para que o software XML2AIML que é desenvolvido neste trabalho possa fazer requisição de buscas aos servidores dos repositórios, serão utilizados os metadados padrões já criados pela plataforma DSpace.

No quinto capítulo é apresentado o desenvolvimento do software analisador e a sua funcionalidade, apresentado suas etapas de processamento na geração de um novo arquivo de extensão AIML.

CAPITULO 4 – XML2AIML

Nos capítulos anteriores são apresentados de como é estruturado um chatterbot, sua base de conhecimento de perguntas e respostas e quais são as suas funcionalidades de quando é implantado em empresas. Também é descrito nos capítulos o que são repositórios digitais institucionais, principalmente repositórios digitais da plataforma DSpace, o motivo das instituições implantarem repositórios digitais em suas universidades e o impacto que é causado após a implantação dos repositórios digitais, como é possível fazer uma busca de arquivos nos repositórios digitais por meios externos e não direto pelas páginas dos repositórios, como funciona os conjuntos de metadados Dublin Core e a estrutura de uma url de requisição de busca de arquivo.

Como visto também, para fazer o reconhecimento das informações vinda do arquivo XML, que é dado como resposta da requisição de busca de arquivo do repositório digital, é necessário os usos das técnicas de PLN.

Neste capítulo são descritos os resultados obtidos neste trabalho, este capítulo fará o uso de toda a bibliografia descrita nos capítulos anteriores.

4.1. Software XML2AIML

O principal objetivo deste trabalho é de abranger o conhecimento dos motores robóticos desenvolvidos na linguagem AIML, com novas bases de conhecimento criadas de forma autônoma por um software responsável por buscar informações em repositórios digitais, por meio de metadados, e tratá-las para gerar as novas bases.

Os repositórios digitais possuem uma limitação em sua ferramenta de busca de material, por não ser muito preciso em encontrar os arquivos quando um usuário utiliza o repositório. Então neste trabalho foi desenvolvido para solucionar os dois problemas, fazendo a utilização de Chatterbots para estarem ajudando os usuários do repositório nas buscas dos arquivos e com a implantação dos Chatterbots nos repositórios, as informações contidas nos DC (metadados) dos repositórios serão utilizados pelo software para abranger com novas bases de conhecimento AIML.

Foi criado um teste de validação do software analisador de metadados XML2AIML, para que sejam encontrados possíveis erros durante o dialogo de um usuário do

repositório digital Univem Aberto com o chatterbot, o teste avalia a geração das novas bases de conhecimento e na resposta do chatterbot com os arquivos já encontrados no repositório digital.

A criação de novas perguntas e respostas para os motores robóticos da linguagem AIML, são necessários utilizar novas bases de informação para poder analisar e tratar, para assim, gerar novas perguntas e respostas em cima dessas novas informações.

Na figura 3 é apresentado um modelo por background, de como funciona a parte dos chatterbot coletarem informações sobre os arquivos que o usuário do repositório está buscando, o tratamento que é aplicado pelo software XML2AIML nessas informações coletadas, a geração da url de requisição de busca, o envio da requisição ao repositório digital Univem Aberto, o reconhecimento das informações contidas no XML vinda como resposta servidor do Univem Aberto, a geração da nova base de conhecimento de perguntas e respostas, para no final, o chatterbot responder quais arquivos foram encontrados no repositório e disponibilizar os links de acesso a esses arquivos.

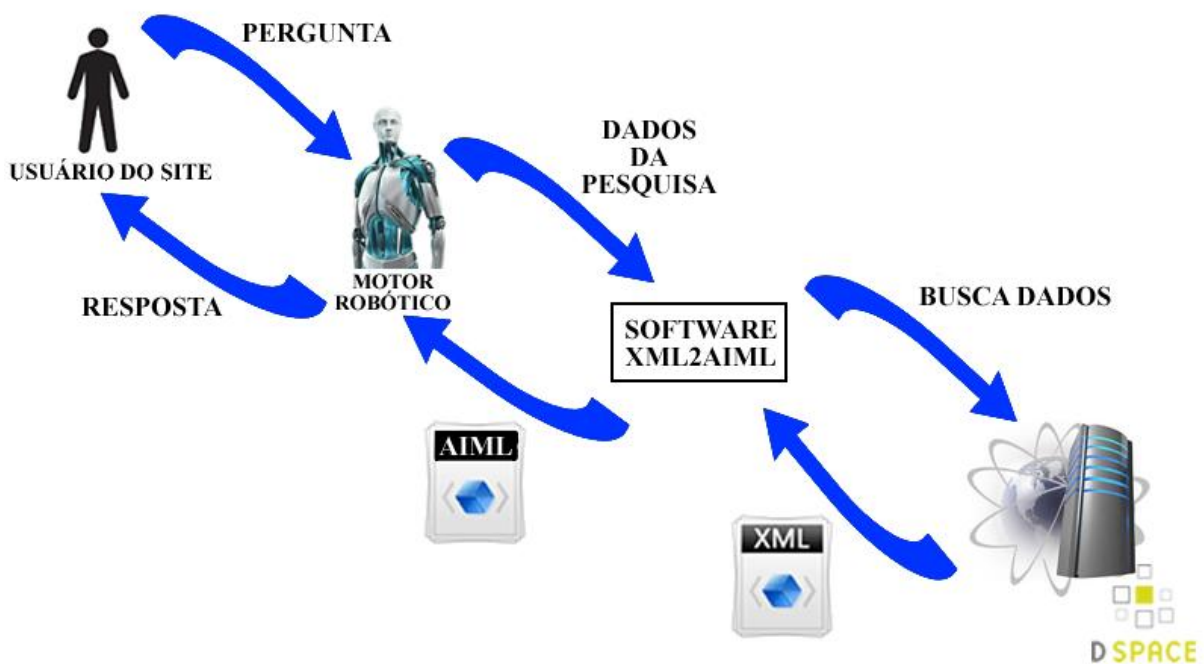


Figura 3: Modelo de funcionalidade de uma busca de arquivo em repositório digital por meio de um chatterbot AIML.

[Fonte: Própria]

Para uma melhor explicação de como é a funcionalidade dos chatterbot e do software XML2AIML foram criados alguns cenários de teste, todos com vários tipos de diálogo entre o usuário com o chatterbot. No cenário que será utilizado para descrever as funcionalidades do software, o usuário está buscando um tutorial de desenvolvimento AIML no repositório aberto.

As funcionalidades estão divididas em quatro etapas: Coletando dados do usuário; Criando URL para requisição da busca dos arquivos, Perguntas e Respostas Pré-definida e Criando nova base de conhecimento. O cenário criado para teste de uma conversação de busca de arquivo no repositório é apresentado abaixo, para cada etapa de funcionalidade, são utilizados determinadas partes do diálogo.

- Diálogo para iniciar uma busca

ROBÔ => Olá, tudo bem?

Usuário => Olá, tudo e com você?

Robô => Estou bem. Caso você esteja procurando um material, pode perguntar para mim que procurarei no repositório aberto do Centro Universitário Eurípides de Marília - UNIVEM.

Usuário => Estou precisando do material de Inteligência Artificial.

Robô => A informação digitada sobre o material que você procura pertence a alguns dos seguintes grupos: AREA, AUTOR, DATA DA PUBLICAÇÃO OU TÍTULO. Por favor, Escreva o tipo da informação e em seguida digite o que você sabe do material que procura. Exemplo: Autor Rafael de Macedo ou Título Implementação de Motores Robóticos.

Usuário => Área Desenvolvimento em AIML.

Robô => Certo, você tem mais alguma informação sobre a área Desenvolvimento em AIML?

Usuário => Eu não tenho.

Robô => Certo. Aguarde enquanto procuro algum material nessa área.

Robô => Foi encontrado um arquivo sobre esta área Desenvolvimento em AIML, este arquivo tem o título Interpretador AIML Alimentado com TAGS HTML5 - Manual de Comandos do AIML, você pode ver o resumo, o autor e o link de acesso para visualizar todo

o arquivo. Se você quiser ver o resumo, digite resumo mais o título do arquivo. Por Exemplo: Resumo Interpretador AIML Alimentado com TAGS HTML5 - Manual de Comandos do AIML.

Usuário => Resumo Interpretador AIML Alimentado com TAGS HTML5 - Manual de Comandos do AIML.

Robô => Interpretador AIML Alimentado com TAGS HTML5 - Manual de Comandos do AIML ----- Resumo: O AIML (Artificial Intelligence Markup Language, Linguagem de Marcação da Inteligência Artificial) é um conjunto de tags XML (eXtensible Markup Language, Linguagem de Marcação Extensível) capaz de representar e relacionar expressões em linguagem natural permitindo a criação de motores capazes de manter um diálogo simples em linguagem natural. Cada conjunto de tags pode possuir uma ou várias tags chamadas de categorias, pois as tags categorias possuem vários tipos de tags dentro de si que fazem a interpretação de uma mensagem escrita por um usuário e assim dar uma resposta através de funções que estão definidas nas tags dentro da tag categoria. AIML é um chatterbot, ou seja, um robô com Inteligência Artificial que interage com usuário através de conversas. Chatterbots significa: Chat = Conversa e bot = Robô.

Usuário => Eu quero poder acessar o arquivo.

Robô => ----- Pagina: <http://hdl.handle.net/11077/804>

Usuário => Muito obrigado.

Robô => De nada. Caso queira mais algum arquivo estou à disposição.

4.2. Coletando dados do usuário

A primeira etapa é coletar o máximo de informação do usuário sobre o material que está procurando, para poder pegar essas informações o chatterbot já contém perguntas que foram criadas para fazer essa coleta. Como as buscas serão feitas pelas variáveis DC, o chatterbot tem que coletar informação que pode ser usada nessas variáveis, ou seja, informações como autor, título, área, data de publicação ou um período de publicação.

No cenário de teste o chatterbot explica para o usuário que se quiser procurar um material, deve seguir algumas regras. Essas regras foram definidas para que o chatterbot possa compreender que tipo de informação que o usuário está passando.

O usuário está procurando materiais da área de desenvolvimento em AIML, o chatterbot salvará essa informação em uma variável e depois pergunta para o usuário se possui mais alguma informação sobre essa área. Como o usuário não possui mais nenhuma

informação, então o chatterbot aciona um elemento script, que é responsável por chamar o método “coletado_informacao (autor, titulo, data_inicial, data_final, area)”, esse método passa todas as informações do chatterbot para o software XML2AIML.

Na seção 4.3 Criando URL para requisição de busca dos arquivos, o software construirá uma url com parâmetro que contem as informações passada do usuário.

4.3. Criando URL para requisição de busca de arquivo

As informações do usuário foram passadas para o software XML2AIML, com essas informações o software analisa quais são os tipos de informações recebidas, para assim, de acordo com o tipo da informação será chamado o método correto.

Neste cenário o usuário passa informação para o chatterbot sobre a área do material, então o software chamará o método responsável por criar uma URL que contém as variáveis DC de parâmetro autor, área e título, mesmo que o chatterbot não tenha passado informações sobre o autor e título. O XML2AIML já está programado para colocar DC na URL apenas as variáveis que foram atribuídas algum valor na hora de chamar o método, essas variáveis são criados no escopo da passagem de parâmetro do método.

O software com a url pronta chamará uma função da biblioteca Java InputStreamReader e BufferedReader, que são responsáveis por fazer a requisição ao servidor do repositório e receber a resposta que o servidor enviará.

O servidor recebe os valores das variáveis de parâmetros passados na url e procura no banco de dados do repositório algum arquivo relacionado a aquelas informações, após procurar, o mesmo gera um XML para enviar como resposta há máquina que fez a requisição. Na figura 4 é apresentado um XML vindo como resposta do servidor do repositório digital Univem Aberto.

```

<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2013-12-08T21:57:41Z</responseDate>
  <request verb="ListRecords" from="2012-11-22T00:00:01Z" metadataPrefix="oai_dc"
  until="2012-11-22T23:59:01Z">http://aberto.univem.edu.br/oai/request</request>
  ▼<ListRecords>
    ▼<record>
      ▶<header>...</header>
      ▼<metadata>
        ▼<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
        xmlns:dc="http://purl.org/dc/elements/1.1/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
        http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
          ▼<dc:title>
            Interpretador AIML Alimentado com TAGS HTML5 - Manual de Comandos do AIML
          </dc:title>
          <dc:creator>Macedo, Rafael Luiz de</dc:creator>
          ▶<dc:description>...</dc:description>
          <dc:date>2012-11-21T11:56:02Z</dc:date>
          <dc:date>2012-11-21T11:56:02Z</dc:date>
          <dc:date>2012-11-21</dc:date>
          <dc:type>Dissertação</dc:type>
          <dc:identifier>http://hdl.handle.net/11077/804</dc:identifier>
          <dc:language>pt_BR</dc:language>
        </oai_dc:dc>
      </metadata>
    </record>
  </ListRecords>
</OAI-PMH>

```

Figura 4: Resposta XML vinda do servidor do repositório Univem Aberto.

[Fonte: Repositório digital Univem Aberto,
http://aberto.univem.edu.br/oai/request?verb=ListRecords&from=2012-11-22T00:00:01Z&until=2012-11-22T23:59:01Z&metadataPrefix=oai_dc]

O software XML2AIML já com a resposta da requisição atribuída em uma variável do tipo `BufferedReader`, primeiro analisa se na resposta contém informações de algum arquivo, caso não contenha o software já passa para a próxima etapa que é a de criação da nova base de informação, apenas contendo nessa nova base perguntas e respostas relacionadas ao assunto de que nenhum arquivo foi encontrado no repositório dentro das informações que o usuário passou para o chatterbot.

Quando o XML vindo da resposta da requisição contém informações dos arquivos encontrados no repositório digital, o software passa para a próxima etapa 4.4 Perguntas e Respostas Pré-Definida.

4.4. Pergunta e Resposta Pré-Definidas

Quando o software XML2AIML analisa e gera as novas perguntas e respostas em cima das informações contidas nos metadados da resposta XML, são realizadas consultas a um banco de dados MySQL para que sejam utilizadas perguntas e respostas pré-definidas neste trabalho.

No banco de dados contém partes de perguntas e respostas, que serão concatenadas com determinados tipos de informações dos metadados contidos no XML, para assim, gerar as perguntas e respostas que serão utilizadas nas novas bases de conhecimentos AIML.

O objetivo do uso de um banco de dados no software XML2AIML é de concentrar as novas perguntas e respostas pré-definidas em um único local, com o uso do banco de dados, fica mais ágil em melhorar e aumentar a quantidade dessas pergunta e resposta que serão gerados para cada arquivo encontrado na busca do repositório.

Neste banco de dados também é utilizado uma tabela, para concentrar as perguntas e respostas, que são responsáveis por coletarem informações sobre o material que o usuário do repositório está buscando. Estas perguntas e respostas sofreram bastantes alterações durante o desenvolvimento do software, para fins de fazer o usuário entender claramente as perguntas e respostas que o chatterbot usará durante o diálogo.

Na próxima etapa 4.5 Criando nova base de conhecimento, são aplicadas todas as perguntas e respostas pré-definidas no banco de dados do software analisador em cima dos metadados do XML vindo do repositório digital Univem Aberto.

4.5. Criando nova base de conhecimento

O software XML2AIML analisa o XML em busca dos metadados de cada arquivo que está contido na resposta, como visto no capítulo 2 e 3, os metadados são utilizados para catalogar cada arquivo e no repositório digital da plataforma DSpace é utilizado o padrão de metadados DC. Na figura 5 é apresentada apenas a parte que será utilizada pelo software analisador, que são os metadados encontrados na resposta XML do repositório digital Univem Aberto.

```

<metadata>
▼<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
  ▼<dc:title>
    Interpretador AIML Alimentado com TAGS HTML5 - Manual de Comandos do AIML
  </dc:title>
  <dc:creator>Macedo, Rafael Luiz de</dc:creator>
  ▶<dc:description>...</dc:description>
  <dc:date>2012-11-21T11:56:02Z</dc:date>
  <dc:date>2012-11-21T11:56:02Z</dc:date>
  <dc:date>2012-11-21</dc:date>
  <dc:type>Dissertação</dc:type>
  <dc:identifier>http://hdl.handle.net/11077/804</dc:identifier>
  <dc:language>pt_BR</dc:language>
</oai_dc:dc>
</metadata>

```

Figura 5: Metadados contidos no XML.

[Fonte: Repositório Aberto, <http://aberto.univem.edu.br/>]

Quando o software reconhece os metadados, automaticamente separa os metadados em uma nova variável, para assim, com a nova variável ficar mais ágil o manuseio das informações sobre cada arquivo encontrado no repositório digital.

Na figura 5, pode ser visto os DC que contém as informações sobre o arquivo, como dc:title, dc:creator, dc:date, dc:type e outros tipos de DC. Os DCs são passados por parâmetros para o método responsável por gerar as perguntas e respostas. Com essas informações, o software chamará o método gerarPergunta (autor, título, resumo, link).

O método gerarPergunta realizará uma consulta ao banco de dados MySQL do software, para pegar palavras que farão parte de cada conjunto de pergunta e resposta, com cada uma das informações que o método receber por parâmetro.

As perguntas e respostas já concatenadas com as informações sobre o arquivo são inseridas de forma automática na nova base de conhecimento AIML, mas para cada conjunto de pergunta e resposta o software XML2AIML insere os elementos que fazem parte da estrutura de uma categoria (Na seção 1.1 Estrutura do arquivo AIML, descreve como é a estrutura de uma categoria e no apêndice B é apresentado à estrutura do elemento category) para que o chatterbot possa reconhecer e utilizar essas novas perguntas e respostas.

Cada conjunto de metadados representa um arquivo encontrado na busca no repositório, para cada conjunto o software gera as mesmas perguntas e respostas que foram utilizadas no primeiro conjunto de metadados.

Quando o software termina de fazer toda análise e aplicar o tratamento de

perguntas e respostas, em todos os conjuntos de metadados, o software encerra a nova base de conhecimento AIML, com o elemento `</aiml>`. No apêndice F é mostrado o arquivo de saída contendo as novas perguntas e respostas, que foi gerado pelo software XML2AIML, o arquivo de saída na figura 6 é uma base de conhecimento gerada pelo software no início deste trabalho.

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<aiml version='1.0.1'
  xmlns='http://alicebot.org/2001/AIML-1.0.1'
  xmlns:html='http://www.w3.org/1999/xhtml'
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xsi:schemaLocation='http://alicebot.org/2001/AIML-1.0.1 http://aitools.org/aiml/schema/AIML.xsd'>
  <category>
    <pattern>Qual artigo você tem *</pattern>
    <template>Interpretador AIML Alimentado com TAGS HTML5 -
      Manual de Comandos do AIML</template>
  </category>
  <category>
    <pattern>Quem * o autor *</pattern>
    <template>Macedo, Rafael Luiz de</template>
  </category>
</aiml>
```

Figura 6: Nova base de conhecimento gerado pelo software XML2AIML.

[Fonte: Própria]

O arquivo gerado pelo software XML2AIML que está no apêndice F é passado para o chatterbot para que ele apresente o resultado da busca para o usuário do repositório, o chatterbot continuará com a conversa dizendo ao usuário que encontrou ou não o material.

No cenário de teste o usuário realiza perguntas sobre o material encontrado pelo chatterbot, e o usuário finaliza com a pergunta que deseja acessar o material completo. O chatterbot dá como resposta o link de acesso à página do material.

Todo o cenário de teste está estruturado na linguagem de marcação AIML, conforme é mostrado no apêndice G, que apresenta vários elementos que foram descritos no capítulo 1 AIML.

4.6. Outros cenários de teste

As perguntas que foram definidas para conter nas novas bases de conhecimento foram criadas em cima de cenários reais de buscas que são feitas no repositório Univem Aberto do Centro Universitário Eurípides de Marília.

Um usuário pode em uma única frase passar mais de um tipo de informação para o chatterbot, como por exemplo: Quero os materiais da área de Inteligência Artificial publicados no mês de novembro.

Esse tipo de frase é uma pergunta de forma indireta, que está escrita de forma explícita, como “Você tem os materiais da área de Inteligência Artificial publicados no mês de novembro?”. O chatterbot reconhece como uma pergunta e responde de forma que oriente o usuário, que ele deve seguir uns padrões para que o chatterbot possa coletar corretamente as informações sobre os arquivos que esteja procurando no repositório digital do Univem Aberto.

Segue abaixo um cenário de teste com mais de um tipo de informação sendo coletada pelo chatterbot.

Usuário => Você tem algum material na área de Inteligência Artificial no mês de novembro do ano 2012?

Robô => Por favor, para que eu possa realizar a busca, digite as informações separadamente, para cada informação digite em uma frase. Por exemplo: Área Metodologia Científica. E depois digite Autor Rafael Luiz de Macedo.

Usuário => Área Inteligência artificial.

Robô => Certo, você tem mais alguma informação sobre a área Inteligência Artificial?

Usuário => Data Novembro de 2012.

Robô => Certo. Aguarde enquanto procuro no repositório aberto.

Neste cenário de tese, o usuário foi orientado pelo chatterbot em separar as informações que contém sobre os arquivos que está procurando no repositório digital Univem Aberto.

No próximo capítulo é apresentado à conclusão do trabalho, quais foram os objetivos alcançados e as considerações finais do trabalho.

CONCLUSÃO

A linguagem de marcação AIML possui uma limitação na criação de motores robóticos de conversação, fazendo com que seja limitado o número de perguntas que possam ser feitas a esses motores robóticos. Devido a essa limitação, os motores robóticos não conseguem manter um diálogo simples por muito tempo com um humano, pois as perguntas que são realizadas fora do contexto de conhecimento em que esses motores robóticos foram desenvolvidos, não conseguem ser respondidas.

Outro problema, são os usuários que acessam os repositórios digitais da plataforma DSpace, pois, encontram dificuldades em obterem os materiais que procuram, então, com o auxílio de um chatterbot para interagir com os usuários e coletar informações sobre os materiais que estão procurando, os usuários conseguem saber em um curto tempo se o repositório tem ou não o material.

O objetivo do trabalho foi alcançado, conseguindo abranger o conhecimento dos motores robóticos da linguagem AIML com novas bases de conhecimentos gerados pelo software analisador XML2AIML, utilizando os dados contidos nos metadados de repositórios digitais da plataforma DSpace, além, de melhorar a procura de arquivos ao usuário nesses repositórios digitais.

A contribuição deste trabalho científico é abranger com novas bases de conhecimentos em forma de perguntas e respostas, para os motores robóticos (chatterbot) desenvolvidos na linguagem de marcação AIML. As novas bases de conhecimento são geradas de forma autônoma pelo software analisador XML2AIML. Outra contribuição é fazer o uso do chatterbot desenvolvido neste trabalho, em auxiliar os usuários nas buscas de arquivos nos repositórios institucionais, fazendo com que seja mais eficiente encontrar esses arquivos com o chatterbot, do que utilizar a ferramenta de busca do próprio repositório institucional.

TRABALHOS FUTUROS

Como base nos resultados apresentados tem-se como trabalho futuro a integração de chatterbot com ferramentas de busca; tecnologias de reconhecimento de gestos e voz; e melhorias no software XML2AIML gerado neste trabalho.

- **Ferramentas de busca**

Com este trabalho espero poder abrir novas pesquisas no uso de chatterbot com ferramentas de buscas online. As tecnologias têm possibilitado cada vez mais os usuários interagir com esses chatterbot, por meio de comandos de voz, escritas ou de gestos, então porque não juntar chatterbot com as ferramentas de buscas.

Quando um usuário utiliza uma ferramenta de busca na web, normalmente a ferramenta não consegue obter resultados precisos que o usuário esteja procurando. O problema não está só na ferramenta de busca, mas sim, na forma de como o usuário se expressa, muitas vezes não está claro nas palavras que o usuário digita nestas ferramentas de buscas.

O chatterbot interagindo com o usuário por meio de um diálogo, o chatterbot pode interpretar melhor sobre o que o usuário esta buscando por meio de pergunta que facilitará ao usuário se expressar com palavras mais específicas sobre o assunto que está procurando.

- **Iteração por meio de gestos e voz**

As tecnologias têm possibilitado cada vez mais aos usuários se interagirem com televisores, geladeiras inteligentes, interação de smartphone com televisão e computador, então, porque não integrar chatterbot, por meio de comandos de voz, escritas ou de gestos. Sugere-se fazer a integração de chatterbot com tecnologias de reconhecimento de gestos e voz, para que essa integração possa gerar frutos como ligar uma televisão e ela perguntar ao usuário qual canal deseja assistir, avisá-lo quando uma programação favorita esta passando e pergunta se quer assistir ou gravar para assisti-lo depois, ou seja, por meio de perguntas e resposta o usuário pode utilizar todas as funcionalidades da televisão sem ao menos utilizar o controle remoto. Outras tecnologias para interagirem com o usuário é auxiliá-lo em rodas diferentes do GPS, avisar quando receber uma ligação e perguntar se quer atender.

- **Utilizar mais repositórios institucionais**

Como trabalho futuro, sugere-se a melhoria do software XML2AIML desenvolvido neste trabalho, em fazer o uso de mais de um repositório institucional quando forem realizadas requisições de buscas de arquivos, para assim, aumentar o número de artigos, monografias e outros tipos de arquivos científicos disponíveis aos usuários.

Concentrando em uma única ferramenta de busca, todos os materiais desenvolvidos em várias universidades que têm implantado repositórios digitais da plataforma DSpace.

PRODUÇÃO BIBLIOGRÁFICA

Trabalhos com os resultados de chatterbot autômatos capazes de se auto-alimentar com novas bases de conhecimentos de perguntas e respostas, resultaram em artigos e foram publicados em eventos científicos.

Trabalhos completos publicados em anais de congresso

- MACEDO, R. L. ; FUSCO, E. ; PEREIRA, F. D. . Implementation of Intelligence Robotic Engines Using Ontologies and Process Natural Language. In: World Congress in Computer Science - Worldcomp - ICAI'13, 2013, Las Vegas. ICAI'13 - The 2013 International Conference on Artificial Intelligence, 2013.
- MACEDO, R. L. ; FUSCO, E. . Interpretador AIML alimentado com tags HTML5. In: 2º Congresso de Pesquisa Científica: Inovação, Ética e Sustentabilidade, 2012, Marília. 2º Congresso de Pesquisa Científica: Inovação, Ética e Sustentabilidade - Anais de Trabalhos Premiados. Marília: Centro Universitário Eurípides de Marília - UNIVEM, 2012. v. 1. p. 48-55.

Resumos expandidos publicados em anais de congressos

- MACEDO, R. L. ; FUSCO, E. . Implementação de Motores Robóticos Inteligentes Utilizando Ontologias e Processamento de Linguagem Natural. In: 21º Simpósio Internacional de Iniciação Científica da USP, 2013, São Carlos. 21º SIICUSP, 2013.

Resumos publicados em anais de congressos

- MACEDO, R. L. ; FUSCO, E. . Interpretador AIML alimentado com tags HTML5: Automatização na Geração de Conhecimentos. In: CONIC Semesp, 2012, São Paulo. 12º Congresso Nacional de Iniciação Científica, 2012.
- MACEDO, R. L. ; FUSCO, E. . Interpretador AIML alimentado com tags HTML5: Estudo dos motores inteligentes. In: CONIC Semesp, 2011, Santos. 11º Congresso Nacional de Iniciação Científica, 2011.
- MACEDO, R. L. ; FUSCO, E. . Interpretador AIML alimentado com tags HTML5: Estudo das Linguagens. In: 1º Congresso de Pesquisa Científica: Inovação, Ética e Sustentabilidade, 2011, Marília. 1º Congresso de Pesquisa Científica: Inovação, Ética e Sustentabilidade, 2011.

REFERÊNCIAS BIBLIOGRÁFICAS

ABERTO, Centro Universitário Eurípides de Marília, “Repositório Digital Univem Aberto”, <http://aberto.univem.edu.br/>, Outubro 2013.

ALICE AI Foundation “AIML: The Artificial Intelligence Markup Language” <http://www.alicebot.org/aiml.html>, Agosto 2013.

ALVES, Rachel C. V. (2005) “Web Semântica: Uma Análise focada no uso de metadados”, Dissertação (Mestrado em Ciência da Informação) – Faculdade de Filosofia e Ciências – Universidade Estadual Paulista – UNESP, Marília/SP.

BUSH, Noel. “ProgramD” <http://aitools.org/Programd>, Agosto 2013.

CONCEPT, “Converse com o Robô Ed!” <http://www.ed.conpet.gov.br/br/converse.php>, Agosto 2013.

FUSCO, Elvis, (2010), “Modelos Conceituais de dados como parte do processo da catalogação: perspectiva de uso dos FRBR no desenvolvimento de catálogos bibliográficos digitais”, Teste de Doutorado, UNESP – Marília.

HILLMANN, Diane, (2005), “Using Dublin Core – The Elements”, Making it easier to find information. Copyright © 1995-2012 DCMI. All Rights Reserved.

MILSTEAD, J.; FELDMAN, S. Metadata Projects and Standards.

MENEZES, Paulo, B. (1998), “Linguagens Formais e Autômatos” Segunda Edição. Porto Alegre.

REPOSITÓRIO INSTITUCIONAL, Universidade de Brasília, <http://repositorio.unb.br/>, Novembro 2013.

RIBEIRO, Gilberto P. (1995), “Metadados Geoespaciais Digitais”, Workshop de bancos de dados não convencionais, Anais, Niterói.

RICH, Elaine e KNIGHT, Kevin. (1993) “Inteligência Artificial” – Segunda Edição. São

Paulo/SP.

RUSSELL, Stuart e NORVIG, Peter. (2004) “Inteligência Artificial” – Tradução da Segunda Edição. Rio de Janeiro/RJ.

School of Mathematics and Statistics University of St. Andres. “Historia do Matemático Alan Turing”, <http://apprendre-math.info/portugal/historyDetail.htm?id=Turing>, Escócia, Agosto 2013.

SOUZA, Marcia I. F., VENDRUSCULO, Laurimar G. e MELO, Geane C., (2000) “Metadados para a descrição de recursos de informação eletrônica: utilização do padrão Dublin Core”, Ci. Inf., Brasília, v. 29, n. 1, p. 93-102, jan./abr.

TANSLEY, Robert, et al.. “The DSpace Institutional Digital Repository System: Current Functionality”, Joint Conference on Digital Libraries, pág. 87-97, EUA, 2003.

TOMAÉL, Maria Inês, SILVA, Terezinha E. da, (2007), “Repositórios Institucionais: diretrizes para políticas de informação”, VIII ENANCIB – Encontro Nacional de Pesquisa em Ciência da Informação, Salvador, Bahia, Brasil.

Universidade do Minho, “RepositórioUM”, <http://repositorium.sdum.uminho.pt/>, Novembro 2013.

UOL, Educação (2008) “Reforma Ortográfica: Conheça as novas regras”.

VELLUCCI, S. L. Bibliographic relationships. International Conference On The Principles And Future Development Of AACR. Toronto: American Library Association: Library Association Publishing, 1998. p. 105-147.

VIEIRA, André M. M., DAVID, Gabriel, (2011), “Repositório Digital e Plataforma de Gestão de Processos: uma arquitetura integrada”, Faculdade de Engenharia da Universidade do Porto FEUP.

WALLACE, Dr. Richard S. (2003) “The Element of AIML Style”, © ALICE A. I. Foundation, Inc., Outubro.

WALLACE, Dr. Richard; TAYLOR, Anthony. “AIML Reference Manual”,

<http://www.alicebot.org/documentation/aiml-reference.html>, Agosto 2013.

WEIBEL, S. (1997) “The Dublin Core: a simple content description model for electronic resources.” *Bulletin of the American Society for Information Science*, p.9-11, Oct/Nov. 1997.

Apêndice A

```

<category>
  <pattern>oi tudo bem *</pattern>
  <template>Ola, tudo bem e com você ?</template>
</category>
<category>
  <pattern>Tem algum artigo sobre *</pattern>
  <template><think><set name="area"><star/></set></think><srai>PROCURA POR AREA</srai></template>
</category>
<category>
  <pattern>Qual artigo * tem nessa area *</pattern>
  <template><think><set name="area"><star/></set></think><srai>PROCURA POR AREA</srai></template>
</category>
<category>
  <pattern>Procuo o meu *</pattern>
  <template><think><set name="area"><star/></set></think><srai>PROCURA          PROPRIO
ARQUIVO</srai></template>
</category>
<category>
  <pattern>Procuo a minha *</pattern>
  <template><think><set name="area"><star/></set></think><srai>PROCURA          PROPRIO
ARQUIVO</srai></template>
</category>
<category>
  <pattern>PROCURA POR AREA</pattern>
  <template>Aguarde, enquanto procuro os materiais relacionados nessa area <get name="area" /></template>
</category>
<category>
  <pattern>PROCURA PROPRIO ARQUIVO</pattern>
  <template>Qual o titulo?</template>
</category>
<category>
  <pattern>voce pode me ajudar *</pattern>
  <template>Voce tem alguma desses dados do que busca: Area, Autor, Data, Titulo. Caso tenha algumas dessas
informacoes digite o tipo e o conteudo. Exemplo: Autor Rafael</template>
</category>
<category>

```

```

<pattern>Titulo *</pattern>
<template><think><set          name="titulo"><star/></set></think><srai>PROCURANDO          POR
TITULO</srai></template>
</category>
<category>
  <pattern>Data *</pattern>
  <template><think><set          name="data"><star/></set></think><srai>PROCURANDO          POR
DATA</srai></template>
</category>
<category>
  <pattern>Autor *</pattern>
  <template><think><set          name="autor"><star/></set></think><srai>PROCURANDO          POR
AUTOR</srai></template>
</category>
<category>
  <pattern>Area *</pattern>
  <template><think><set name="area"><star/></set></think><srai>PROCURA POR AREA</srai></template>
</category>
<category>
  <pattern>PROCURANDO POR TITULO</pattern>
  <template>Aguarde, estou procurando o material com o titulo <get name="titulo" /></template>
</category>
<category>
  <pattern>PROCURANDO POR DATA</pattern>
  <template>Aguarde, estou procurando materiais nessa data <get name="data" /></template>
</category>
<category>
  <pattern>PROCURANDO POR AUTOR</pattern>
  <template>Aguarde, estou procurando o material com o autor <get name="autor" /></template>
</category>
<category>
  <pattern>OBSERVACAO TITULO E RESUMO Interpretador AIML Alimentado com TAGS HTML5 -
Manual de Comandos do AIML</pattern>
  <template>Caso queira ver o titulo e o resumo de algum desses autores, digite o nome do autor exetamente
como esta sendo mostrado acima</template>
</category>

```

Apêndice B

```
<category>
  <pattern>Area *</pattern>
  <template>Certo, você tem mais alguma informação sobre a área <think><set
name="area"><star/></set></think>?</template>
</category>
```

Apêndice C

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<substitutions xmlns="http://aitools.org/programd/4.6/bot-configuration"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://aitools.org/programd/4.6/bot-configuration
  http://aitools.org/programd/4.6/bot-
configuration.xsd">
  <input>
    <substitute find="além" replace="alem"/>
    <substitute find="Análise" replace="Analise"/>
    <substitute find="análise" replace="analise"/>
    <substitute find="Aquê" replace="Aquem"/>
    <substitute find="aquê" replace="aquem"/>
    <substitute find="Artificial" replace="Artificial"/>
    <substitute find="artificial" replace="artificial"/>
    <substitute find="Está" replace="esta"/>
    <substitute find="está" replace="esta"/>
    <substitute find="História" replace="Historia"/>
    <substitute find="hstória" replace="historia"/>
    <substitute find="mantêm" replace="mantem"/>
    <substitute find="mantém" replace="mantem"/>
    <substitute find="Não" replace="Nao"/>
    <substitute find="não" replace="nao"/>
    <substitute find="Nós" replace="Nos"/>
    <substitute find="nos" replace="nos"/>
    <substitute find="pára" replace="para"/>
    <substitute find="Pré" replace="Pre"/>
```

```

<substitute find="pré" replace="pre"/>
<substitute find="Pró" replace="Pro"/>
<substitute find="pró" replace="pro"/>
<substitute find="pós" replace="pos"/>
<substitute find="Quêm" replace="Quem"/>
<substitute find="quêm" replace="quem"/>
<substitute find="Recém" replace="Recem"/>
<substitute find="recém" replace="recem"/>
<substitute find="Semântica" replace="Semantica"/>
<substitute find="semântica" replace="semantica"/>
<substitute find="só" replace="so"/>
<substitute find="têm" replace="tem"/>
<substitute find="você" replace="voce"/>
<substitute find="Você" replace="Voce"/>
<substitute find="," replace=""/>
<substitute find="á" replace="a"/>
<substitute find="à" replace="a"/>
<substitute find="ã" replace="a"/>
<substitute find="â" replace="a"/>
<substitute find="é" replace="e"/>
<substitute find="è" replace="e"/>
<substitute find="ê" replace="e"/>
<substitute find="í" replace="i"/>
<substitute find="ì" replace="i"/>
<substitute find="ó" replace="o"/>
<substitute find="ò" replace="o"/>
<substitute find="ú" replace="u"/>
<substitute find="ù" replace="u"/>
<substitute find="û" replace="u"/>
<substitute find="ü" replace="u"/>

```

```
</input>
```

```
</substitutions>
```

Apêndice D

http://aberto.univem.edu.br/oai/request?verb=ListRecords&from=2012-11-22T00:00:01Z&until=2012-11-23T08:00:01Z&metadataPrefix=oai_dc

Apêndice E

```

<OAI-
PMH xmlns="http://www.openarchives.org/OAI/2.0/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2013-11-05T03:08:21Z</responseDate>
  <request verb="ListRecords" from="2012-11-22T00:00:01Z" metadataPrefix="oai_dc" until="2012-11-
23T08:00:01Z">http://aberto.univem.edu.br/oai/request</request>
  <ListRecords>
    <record>
      <header>
        <identifier>oai:aberto.univem.edu.br:11077/804</identifier>
        <timestamp>2012-11-22T05:00:06Z</timestamp>
        <setSpec>hdl_11077_180</setSpec>
      </header>
      <metadata>
        <oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/" xmlns:dc="http://
purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
          <dc:title>
            Interpretador AIML Alimentado com TAGS HTML5 - Manual de Comandos do AIML
          </dc:title>
          <dc:creator>Macedo, Rafael Luiz de</dc:creator>
          <dc:description>
            O AIML (Artificial Intelligence Markup Language, Linguagem de Marcação da
            Inteligência Artificial) é um conjunto de tags XML(eXtensible Markup Language,
            Linguagem de Marcação Extensível) capaz de representar e relacionar expressões em
            linguagem natural permitindo a criação de motores capazes de manter um diálogo
            simples em linguagem natural. Cada conjunto de tags pode possui uma ou varias tags
            chamada de categorias, pois as tags categorias possuem varias tipos de tags dentro de si
            que fazem a interpretação de uma mensagem escrita por um usuário e assim dar uma
            resposta através de funções que estão definidas nas tags dentro da tag categoria. AIML é
            um chatterbots, ou seja, um robô com Inteligência Artificial que interage com usuário
            através de conversas. Chatterbots significa: Chat = Conversa e bot = Robô.
          </dc:description>
          <dc:date>2012-11-21T11:56:02Z</dc:date>

```

```
<dc:date>2012-11-21T11:56:02Z</dc:date>  
<dc:date>2012-11-21</dc:date>  
<dc:type>Dissertação</dc:type>  
<dc:identifier>http://hdl.handle.net/11077/804</dc:identifier>  
<dc:language>pt_BR</dc:language>  
</oai_dc:dc>  
</metadata>  
</record>  
</ListRecords>  
</OAI-PMH>
```

Apêndice F

```

<?xml version='1.0' encoding='ISO-8859-1'?>
<aiml version='1.0.1' xmlns='http://alicebot.org/2001/AIML-1.0.1' xmlns:html='http://www.w3.org/1999/xhtml'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://alicebot.org/2001/AIML-1.0.1 http://aitools.org/aiml/schema/AIML.xsd'>
<category>
    <pattern>RESULTADO GERADO</pattern>
    <template>Foi encontrado um arquivo sobre esta área Desenvolvimento em AIML, este arquivo tem o
título: Interpretador AIML Alimentado com TAGS HTML5 - Manual de Comandos do AIML, você pode ver o
resumo, o autor e o link de acesso para visualizar todo o arquivo. Se você quiser ver o resumo, digite resumo
mais o título do arquivo. Por Exemplo: Resumo Interpretador AIML Alimentado com TAGS HTML5 - Manual
de Comandos do AIML.</template>
</category>
<category>
    <pattern>Autor Interpretador AIML Alimentado com TAGS HTML5 - Manual de Comandos do
AIML</pattern>
    <template>Autor Macedo, Rafael Luiz de</template>
</category>
<category>
    <pattern>Resumo Interpretador AIML Alimentado com TAGS HTML5 - Manual de Comandos do
AIML</pattern>
    <template>Resumo: O AIML (Artificial Intelligence Markup Language, Linguagem de Marcação da
Inteligência Artificial) é um conjunto de tags XML(eXtensible Markup Language, Linguagem de Marcação
Extensível) capaz de representar e relacionar expressões em linguagem natural permitindo a criação de motores
capazes de manter um diálogo simples em linguagem natural. Cada conjunto de tags pode possui uma ou varias
tags chamada de categorias, pois as tags categorias possuem varias tipos de tags dentro de si que fazem a
interpretação de uma mensagem escrita por um usuário e assim dar uma resposta através de funções que estão
definidas nas tags dentro da tag categoria. AIML é um chatterbots, ou seja, um robô com Inteligência Artificial
que interage com usuário através de conversas. Chatterbots significa: Chat = Conversa e bot = Robô.</template>
</category>
<category>
    <pattern>LINK Interpretador AIML Alimentado com TAGS HTML5 - Manual de Comandos do
AIML</pattern>
    <template>Segue o link da página do material ----- Pagina: http://hdl.handle.net/11077/804</template>
</category>
</aiml>

```


Apêndice G

<category>

<pattern>Olá, tudo e com voce *</pattern>

<template>Estou bem. Caso você esteja procurando um material, pode perguntar para mim que procurarei no repositório aberto do Centro Universitário Eurípides de Marília - UNIVEM.</template>

</category>

<category>

<pattern>* precisando * material *</pattern>

<template><srail>DADOS DO MATERIAL</srail></template>

</category>

<category>

<pattern>DADOS DO MATERIAL</pattern>

<template>A informação digitada sobre o material que você procura pertence a alguns dos seguintes grupos: AREA, AUTOR, DATA DA PUBLICAÇÃO OU TÍTULO.
Por favor, Escreva o tipo da informação e em seguida digite o que você sabe do material que procura. Exemplo: Autor Rafael de Macedo ou Título Implementação de Motores Robóticos.</template>

</category>

<category>

<pattern>Area *</pattern>

<template>Certo, você tem mais alguma informação sobre a área <think><set name="area"><star/></set></think>?</template>

</category>

<category>

<pattern>Eu nao tenho</patter>

<template><srail>PROCURA POR AREA</srail></template>

</category>

<category>

<pattern>PROCURA POR AREA</pattern>

<template>Certo. Aguarde enquanto procuro algum material nessa área.</template>

</category>

<category>

<pattern>RESULTADO GERADO</pattern>

<template>Foi encontrado um arquivo sobre esta área Desenvolvimento em AIML, este arquivo tem o título: Interpretador AIML Alimentado com TAGS HTML5 - Manual de Comandos do AIML, você pode ver o resumo, o autor e o link de acesso para visualizar todo o arquivo. Se você quiser ver o resumo, digite resumo mais o título do arquivo. Por Exemplo: Resumo Interpretador AIML Alimentado com TAGS HTML5 - Manual de Comandos do AIML.</template>

</category>

<category>

<pattern>Resumo Interpretador AIML Alimentado com TAGS HTML5 - Manual de Comandos do AIML</pattern>

<template>Resumo: O AIML (Artificial Intelligence Markup Language, Linguagem de Marcação da Inteligência Artificial) é um conjunto de tags XML(eXtensible Markup Language, Linguagem de Marcação Extensível) capaz de representar e relacionar expressões em linguagem natural permitindo a criação de motores capazes de manter um diálogo simples em linguagem natural. Cada conjunto de tags pode possuir uma ou várias tags chamadas de categorias, pois as tags categorias possuem vários tipos de tags dentro de si que fazem a interpretação de uma mensagem escrita por um usuário e assim dar uma resposta através de funções que estão definidas nas tags dentro da tag categoria. AIML é um chatterbots, ou seja, um robô com Inteligência Artificial que interage com usuário através de conversas. Chatterbots significa: Chat = Conversa e bot = Robô.</template>

</category>

<category>

<pattern>Eu quero poder acessar o material inteiro</pattern>

<template>Digite o título do material que deseja visualizar acessar, como por exemplo: LINK Motores Robóticos.</template>

</category>

<category>

<pattern>LINK Interpretador AIML Alimentado com TAGS HTML5 - Manual de Comandos do AIML</pattern>

<Template>Segue o link da página do material ----- Pagina: <http://hdl.handle.net/11077/804></template>

</category>

<category>

<pattern>Muito Obrigado</pattern>

<template>De nada. Caso queira mais algum material, estou à disposição.</template>

</category>