

UM CHATTERBOT PARA AUXILIAR NO APRENDIZADO EM QUALIDADE DE SOFTWARE

Marcello Thiry
thiry@univali.br

Diego Eduardo Rios
drios@univali.br

Universidade do Vale do Itajaí – UNIVALI Campus São José
LQPS – Laboratório de Qualidade e Produtividade de Software
Rodovia SC 407, km 4
88122-000 São José-SC Brasil
Fone: (048) 281-1505 Fax: (048) 281-1506

Resumo

Este artigo procura fazer uma contribuição na utilização de *chatterbots* para auxiliar no aprendizado em qualidade de software. Grande parte dos ambientes existentes atualmente para educação à distância está focada no uso de ferramentas tradicionais como *chats* e fóruns, os *chatterbots* funcionam como uma alternativa a essas ferramentas, nele o usuário obtém as informações através de uma conversa, com a vantagem de ter acesso às informações a qualquer momento. *Chatterbots* são programas que possuem a capacidade de interagir com o usuário por meio de textos. O *chatterbot* criado utiliza uma base de conhecimento desenvolvida na linguagem AIML e possui conhecimentos em qualidade de *software*.

Palavra-chave: Chatterbots, educação à distância, AIML.

1. Introdução

Desde a antiguidade, os homens tentam reproduzir em pinturas, estátuas e máquinas as características que os tornam seres humanos, como formas, raciocínio, interpretação, fala, entre outras. Pensando nisso, em 1950, o matemático inglês Alan Turing, baseado na teoria de Descartes, na qual sugeria que para se diferenciar máquinas de humanos era necessário obter um diálogo entre ambos, formulou a pergunta “Máquinas podem pensar?”, propondo assim o Teste de Turing, que era chamado de Jogo da Imitação (CHEONG, 1996, p. 250). Esse jogo baseia-se em um juiz, um jogador humano e um jogador máquina. O juiz deve interagir com ambos por meio de um sistema de conversação, onde a máquina tentará enganar o juiz imitando o comportamento de um ser humano. O juiz então deverá descobrir qual dos jogadores é a máquina.

Para uma máquina ter a capacidade de compreensão pode-se utilizar agentes inteligentes, “que são programas capazes de realizar atividades interativas que o usuário necessita ou deseja. Com certo grau de independência e autonomia.” (LESNICK e MOORE, 1996, p. 09).

Agentes com a capacidade de conversação são chamados de *chatterbots*, “programas destinados a demonstrar a capacidade racional de um computador, ou seja, programas que buscam imitar o comportamento da inteligência humana.” (LEONHARDT et al, 2003, p. 01).

Os primeiros *chatterbots* utilizavam pequenas bases de conhecimento, e não haviam linguagens e modelos apropriados (LAVEN, 2001). A chamada segunda geração de *chatterbots* utilizava novas técnicas de Inteligência Artificial como redes neurais, possibilitando assim também a capacidade de aprendizado durante os diálogos (GARNER, 1999). Na terceira geração iniciou-se o uso de tecnologias próprias para conversação como a AIML (*Artificial Intelligence Markup Language*), tecnologia baseada em XML (*Extensible Markup Language*) (MENEZES e TEIXEIRA, 2003).

A AIML foi desenvolvida por Dr. Richard S. Wallace em conjunto com a comunidade de software livre Alicebot, no período de 1995 até 2000. A linguagem AIML descreve como será o comportamento do programa que a está processando, as *tags* (marcações) da linguagem contém os conhecimentos do *chatterbot*, possuindo em cada *tag* uma pergunta e a respectiva resposta.(WALLACE, 200-). AIML é uma linguagem simples, e que facilita a criação de uma extensa base de conhecimentos (ALICE, 200-).

Os *chatterbots* podem possuir diversas utilidades, como o assistente virtual disponibilizado pela Philips que tira dúvidas sobre seus produtos (PHILIPS, 200-), o assistente do Microsoft Office que auxilia os usuários na utilização da ferramenta utilizada (MICROSOFT, 2003), a Professora Elektra, que é utilizada como um instrumento de complementação no aprendizado de estudantes de cursos à distância (LEONHARDT et al, 2003), entre outros.

Dentro deste contexto, este artigo procura fazer uma contribuição no estudo sobre a utilização de *chatterbots* para auxiliar no aprendizado em qualidade de software. É implementado um sistema capaz de recriar um ambiente de conversação, onde o usuário poderá inserir através de texto suas perguntas em linguagem natural e receber as respostas.

2. Chatterbots

Chatterbots são “programas destinados a demonstrar a capacidade racional de um computador, ou seja, programas que buscam imitar o comportamento da inteligência humana.” (LEONHARDT et al, 2003, p. 01).

Para simular o entendimento de textos, os *chatterbots* geralmente utilizam uma técnica da inteligência artificial chamada processamento da linguagem natural (NORVIG e RUSSEL, 1995). Graças ao processamento da linguagem natural, os *chatterbots* conseguem “conversar” com o usuário como uma pessoa. Isso representa um grande potencial em relação a educação à distância, pois possibilita a autonomia e desenvoltura em dialogar naturalmente sobre um assunto de estudo, sem estar preso a perguntas e respostas fixas.

2.1. Base de conhecimento

A base de conhecimento é o local onde é armazenado todo o conhecimento do *chatterbot* (WALLACE, 200-).

O *chatterbot* apresentado neste trabalho irá utilizar a linguagem AIML para o desenvolvimento de sua base de conhecimento.

A linguagem AIML descreve como será o comportamento do programa que a está processando, as *tags* (marcações) da linguagem contém os conhecimentos do *chatterbot*, possuindo em cada *tag* uma pergunta e a respectiva resposta (WALLACE, 200-).

Segundo Menezes e Teixeira (2003), a experiência concebida com A.L.I.C.E. mostra que aproximadamente 2.000 palavras atendem a 95% das opções escolhidas pelas pessoas como a primeira palavra do início de uma frase. O número de opções para a segunda e terceira em diante diminui consideravelmente. Com aproximadamente 41.000 unidades de conhecimento é possível estabelecer um bom diálogo.

2.2. Obtenção de conhecimento

A obtenção de conhecimento é de grande importância para o bom funcionamento do *chatterbot*. Programas como A.L.I.C.E. não adquirem conhecimento através de conversas com usuários, mas sim, de edições feitas diretamente na sua base de conhecimento (WALLACE, 2001). Normalmente,

essa inclusão de conhecimento é feita baseada em diálogos que o *chatterbot* teve com alguns usuários, assim, pode-se identificar assuntos sobre os quais o *chatterbot* não possui ou necessita de mais conhecimentos.

2.3. Modelo funcional

De acordo com Ferrari, Geyer e Sganderla (2003), os *chatterbots* necessitam de uma base de conhecimento, onde é armazenado todo o conhecimento do programa, uma aplicação que é responsável por realizar o tratamento da linguagem natural das frases de entrada e também de saída, além de alguma arquitetura que possibilite a utilização remota do sistema, propiciando aos usuários acesso ao programa a qualquer momento de qualquer local.

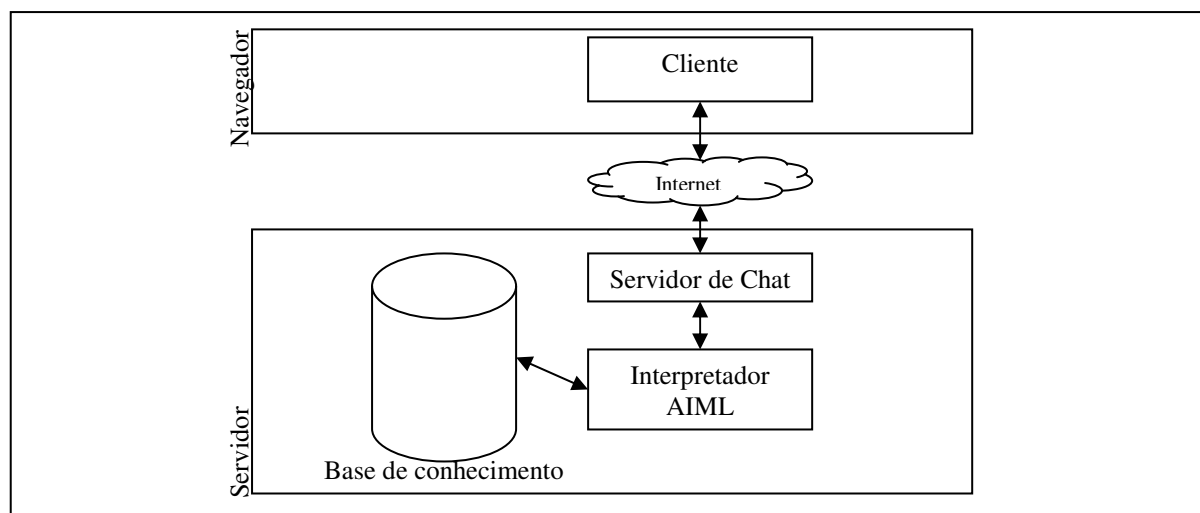


Figura 1 - Modelo funcional do sistema

Para o tratamento das mensagens, neste trabalho optou-se em utilizar uma alternativa equivalente ao processamento de linguagem natural. O processamento da linguagem natural é feito através de arquivos AIML que são baseados no formato XML. O arquivo AIML é processado por um programa de interpretação, que é responsável por identificar as palavras chaves da frase enviada e também responsável por dar uma resposta a mesma.

3. Processamento da linguagem natural

O processamento de linguagem natural é o conjunto de métodos formais para analisar textos e gerar frases escritas em um idioma humano (OLIVEIRA, 200-). O processamento da linguagem natural faz com que os seres humanos possam se comunicar com os computadores de uma forma mais natural, ou seja, a forma com que os humanos estão acostumados a utilizar. Desta forma extingui-se a necessidade de adaptação a linguagem de um equipamento. Ao invés de utilizar comandos específicos, pode-se ordenar a execução de um procedimento por meio de frases como as utilizadas no dia a dia dos usuários.

Segundo Beardon, Holmes e Lunsden (1991), pode-se dividir a linguagem em:

- Linguagem natural - algo que já existe e desempenha várias funções no diálogo com outras pessoas;
- Linguagem artificial - algo que é definido, com propósito específico e de aplicação mais restrita.

A linguagem natural é a linguagem utilizada pelas pessoas para se comunicarem. A linguagem artificial possui algumas restrições em relação à linguagem natural, pois possui regras fixas, procedimentos definidos que fazem com que o computador execute uma tarefa. Na linguagem natural existe ambigüidade, necessidade de interpretação de acordo com o contexto, conhecimento do mundo, regras gramaticais, entre outras características.

3.1. Artificial Intelligence Markup Language (AIML)

AIML foi desenvolvido por Dr. Richard S. Wallace em conjunto com a comunidade de software livre Alicebot no período de 1995 até 2000 (ALICE, 200-). A linguagem AIML é baseada em XML (*Extensible Markup Language*) e descreve como será o comportamento do programa que a está processando. As *tags* (marcações) da linguagem contém os conhecimentos do *chatterbot*, possuindo em cada *tag* uma pergunta e a respectiva resposta (WALLACE, 200-). AIML é uma linguagem simples, facilitando assim a criação de uma extensa base de conhecimento (ALICE, 200-).

AIML é formada por padrões, conhecidos como categorias (*category*). São nessas categorias em que o conhecimento do *chatterbot* é inserido. Cada categoria consiste em uma pergunta de entrada, uma resposta de saída, e um texto opcional. A pergunta ou estímulo é chamada de “*pattern*” (padrão). A resposta é chamada de “*template*” (molde). Os textos opcionais são chamados de “*that*” ou “*topic*”. Encontra-se um exemplo de uma categoria em AIML na Tabela 1.

A linguagem *pattern* é constituída de palavras-chave, espaços em branco e pelos símbolos _ e *. As palavras podem conter somente letras e numerais, não importando se serão maiúsculas ou minúsculas, e são separadas por espaços em branco. Os símbolos funcionam como palavras.

```
<category>
  <pattern>ENTRADA</pattern>
  <template>
    resposta
  </template>
</category>
```

Tabela 1 – Formato de uma categoria em AIML

4. Utilização da técnica de Stemmer

Um dos grandes problemas encontrados na construção e manutenção do *chatterbot* está na implementação da base de conhecimento. Para o *chatterbot* funcionar de maneira satisfatória, é necessário uma base que atenda o objetivo do *chatterbot*. Para que isso seja feito, será preciso conhecimentos específicos, no caso deste trabalho, conhecimento em qualidade de software, e também irá ser necessária a capacidade de interagir com o usuário de uma maneira humana, através de conversas como “olá”, “tudo bem”, entre outros. A língua portuguesa possui inúmeras variações morfológicas de palavras, portanto, aumentando a dificuldade na construção da base de conhecimento. Devido a essa variação, o *chatterbot* poderia deixar de identificar inúmeras frases, pois não são escritas de maneira idênticas as contidas na base de conhecimento.

Conforme o exemplo encontrado na Tabela 2, um simples erro de conjugação verbal pode acarretar um erro de reconhecimento em uma categoria em AIML.

<p>Categoria contida na base de conhecimento</p> <pre> <category> <pattern>TU PODES ENTENDER</pattern> <template> Claro que posso. </template> </category> </pre> <p>Entrada do usuário e resposta do <i>chatterbot</i></p> <p>Usuário: Tu pode entender.</p> <p><i>Chatterbot</i>: Não pude entender sua última frase.</p>

Tabela 2 – Exemplo de um problema ocorrido pela variação morfológica das palavras.

Com o objetivo de facilitar a construção da base de conhecimento e diminuir as chances de não reconhecimentos de categorias, aplicou-se neste trabalho uma técnica chamada *stemmer* como uma contribuição ao modelo já utilizado pelo grupo que desenvolveu a AIML.

A técnica de *steemer* consiste em extrair sufixos e prefixos das palavras permanecendo assim somente a raiz (*stem*) da mesma (PANESSI e BORDIGNON, 2003, p. 01). Na Tabela 3 encontra-se um exemplo de cinco palavras que possuem o mesmo *stem*.

<p>Construir</p> <p>Construído</p> <p>Construindo</p> <p>Construiremos</p> <p>Construirá</p>	<p>} Constr</p>
--	-----------------

Tabela 3 – Exemplo de extração do *stem* das palavras

Um exemplo de utilização do algoritmo de *stemmer* é em sistemas de buscas na *Web* como o *Altavista*, que além de pesquisar a palavra solicitada pelo usuário o sistema também busca as palavras com o mesmo *stem* (BRAHM *et al*, 2004, p. 08).

Considerando a capacidade do algoritmo de *stemmer* em reduzir a um mesmo radical, palavras diferentes, a sua utilização juntamente com o algoritmo do *chatterbot* aumentou a capacidade de reconhecimento do *chatterbot*. Ou seja, uma mesma categoria pode atender a diferentes variações de escrita e tempos verbais.

4.1. Aplicação do *Stemmer* no algoritmo do *chatterbot*

O algoritmo do *stemmer* é aplicado em duas regiões do *chatterbot*: leitura das categorias e entrada de dados.

- **Leitura das categorias** – ao se iniciar o *chatterbot* o programa carrega em memória toda a sua base de conhecimento, ou seja, as categorias serão lidas uma a uma formando uma lista de objetos contendo cada um, uma categoria diferente. Em cada *pattern* das categorias, será aplicado o algoritmo de *stemmer* na frase inteira. Assim, a criação da base de conhecimento não é comprometida, já que pode-se escrever *patterns* com as palavras em seus formatos normais.
- **Entrada de dados** – Quando o usuário realiza alguma pergunta para o *chatterbot* o sistema captura essa mensagem e antes de compará-la com as categorias existentes é

aplicado o algoritmo de *stemmer*. Após a aplicação do *stemmer* a frase irá ser formada apenas por *stems*, podendo então ser comparada com os *patterns* existentes na lista de objetos, já que esses *patterns* já sofreram a aplicação do algoritmo *stemmer*.

A Figura 2 mostra os locais onde o algoritmo *stemmer* é aplicado no modelo funcional do *chatterbot*. A aplicação realizada dessa forma faz com que o sistema e a base de conhecimento não necessitem de alterações para que continuem funcionando de maneira correta.

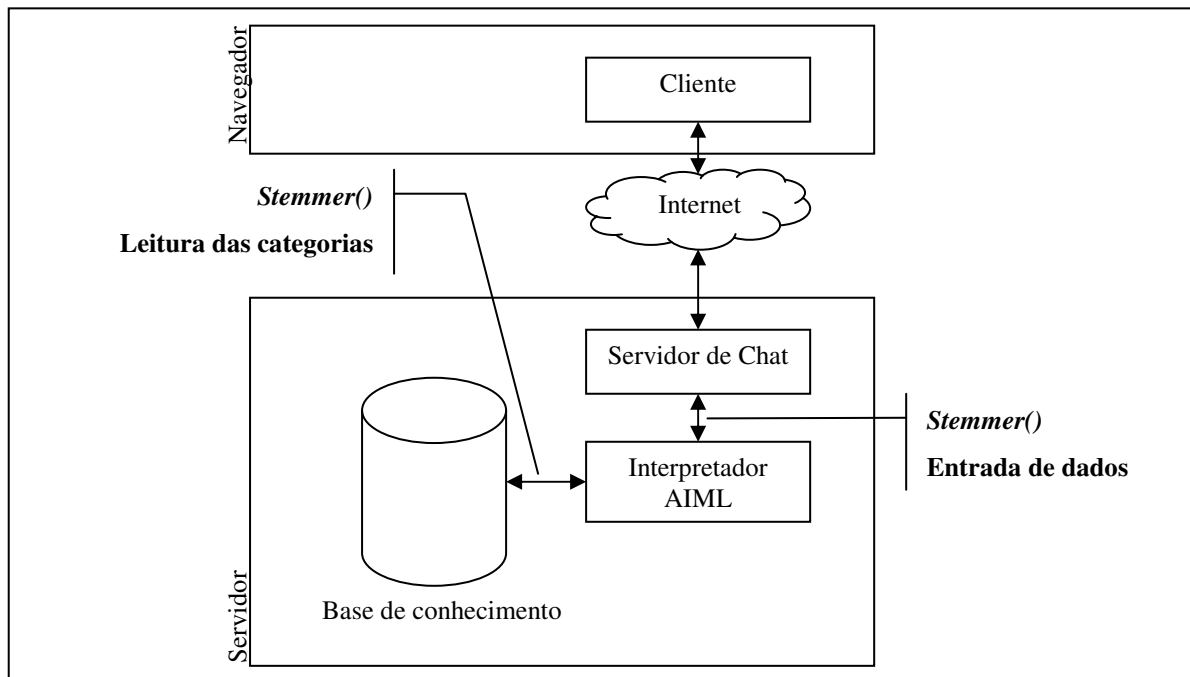


Figura 2 – Locais onde o algoritmo *stemmer* é aplicado no modelo funcional do *chatterbot*

5. Aplicação do modelo proposto

Para realizar o teste de funcionamento da ferramenta foi construída uma base de conhecimento contendo teorias sobre qualidade de *software*. Essa base foi implementada tentando direcionar a conversa para assuntos que o *chatterbot* possua conhecimento, conforme exemplo retirado do *log* do *chatterbot* (Tabela 4).

[24/05/2005 16:54:07][oi][Olá Desconhecido! Sou um Chatterbot que possui conhecimentos em Qualidade de Software.]
[24/05/2005 16:54:15][tudo bem][Tudo, e com você?]
[24/05/2005 16:54:16][bem][Possuo conhecimentos na área de Qualidade de Software. Você está precisando de alguma ajuda?]
[24/05/2005 16:54:23][sim][Tentarei responder da melhor maneira possível.]

Tabela 4 – Exemplo de direcionamento de conversa

Além de possuir conhecimentos em qualidade de *software*, o *chatterbot* deve possuir a capacidade de interagir com o usuário, ou seja, ele deve poder conversar de uma forma humana. Sendo assim, a base de conhecimento do *chatterbot* deve conter categorias para conversação, como categorias de saudação “Oi”, “Tudo bem?” e categorias de despedidas “Tchau!”, “Até mais”, entre outros tipos de categorias que dêem suporte para uma conversação humana.

Para conversar com o *chatterbot* foram convidadas pessoas com experiências em qualidade de *software* e pessoas sem conhecimentos específicos na área. A cada usuário que utilizava o *chatterbot* recolhiam-se os arquivos de *log* para análise do funcionamento do programa, e para possível inclusão de categorias necessárias para o bom funcionamento do programa.

Toda a análise do funcionamento do *chatterbot* foi realizada através dos *logs* registrados pelo sistema. A cada conversa realizada com o *chatterbot*, um arquivo de *log* da conversa é criado. Cada arquivo armazena todos os dados de entrada inseridos pelo usuário e todas as respectivas respostas do *chatterbot*. A Tabela 5 contém o exemplo de um arquivo gerado pelo *log* do *chatterbot*.

[24/05/2005 16:54:07][oi][Olá Desconhecido! Sou um Chatterbot que possui conhecimentos em Qualidade de Software.]
[24/05/2005 16:54:15][tudo bem][Tudo, e com você?]
[24/05/2005 16:54:16][bem][Possuo conhecimentos na área de Qualidade de Software. Você está precisando de alguma ajuda?]
[24/05/2005 16:54:23][sim][Irei respondê-lo da melhor maneira possível.]
[24/05/2005 16:54:41][o que é qualidade][Você pode entender a qualidade como sendo a totalidade de características de uma entidade que lhe confere a capacidade de satisfazer as necessidades explícitas e implícitas. Esta definição é usada pela norma NBR ISO 8402. Para ajudar no entendimento, uma entidade pode ser uma atividade ou um processo, um produto, uma organização ou uma combinação entre eles.]
[24/05/2005 16:55:05][obrigado][De nada.]
[24/05/2005 16:55:10][tchau][Obrigado por conversar comigo, Desconhecido.]

Tabela 5 – Exemplo de um arquivo gerado pelo *log* do sistema.

Através da análise do arquivo de *log* notou-se que ao realizar uma conversa com o *chatterbot*, o usuário não possui um padrão a ser seguido para formular uma pergunta, isto significa que, existem várias maneiras de se fazer a mesma pergunta. **Erro! A origem da referência não foi encontrada..** Se uma das maneiras da pergunta ser feita ao *chatterbot* não for reconhecida, o programa retornará uma resposta não correspondente à pergunta do usuário. Mesmo que o *chatterbot* possua conhecimento suficiente para responder corretamente, se a maneira com que a pergunta foi formulada não for idêntica à existente na base de conhecimento, o programa não responderá corretamente.

A característica de existir mais de uma forma de se fazer uma pergunta faria com que fosse necessária a criação de várias categorias para se obter uma mesma resposta.

Para que não fosse preciso criar várias categorias para representação de cada um dos conceitos sobre qualidade de *software* inseridos na base de conhecimento do *chatterbot*, foram criadas categorias padrões de ligação. Essas categorias servem como ligação das perguntas dos usuários com os conceitos existentes na base de conhecimento. A Tabela 6 contém dois exemplos de categorias utilizadas como ligação e uma categoria de conceito.

```

<category>
  <pattern>O QUE E * </pattern>
  <template>
    <srai><star/></srai>
  </template>
</category>
<category>
  <pattern>DEFINA * </pattern>
  <template>
    <srai><star/></srai>
  </template>
</category>
<category>
  <pattern>QUALIDADE </pattern>
  <template>
    Você pode entender a qualidade como sendo a totalidade de características de uma entidade que lhe confere a capacidade de satisfazer as necessidades explícitas e implícitas. Esta definição é usada pela norma NBR ISO 8402. Para ajudar no entendimento, uma entidade pode ser uma atividade ou um processo, um produto, uma organização ou uma combinação entre eles.
  </template>
</category>

```

Tabela 6 – Exemplo de categorias utilizadas como ligação

5.1. Análise dos arquivos de *log*

A Tabela 7 contém um trecho de um arquivo de *log*, onde a segunda frase inserida pelo usuário não foi reconhecida pelo *chatterbot*. Analisando a frase, nota-se que a frase não reconhecida é de uso comum, portanto deveria ter sido entendida pelo usuário. Sendo assim, criar-se-á uma categoria que suporte esta frase. A Tabela 8 mostra a categoria criada após a análise do arquivo de *log* contido na Tabela 7.

[11/05/2005 10:21:18][oi][Olá!]

[11/05/2005 10:21:45][você sabe o que é qualidade][Infelizmente não pude entender sua última frase.]

[11/05/2005 10:22:14] [o que é qualidade][Você pode entender a qualidade como sendo a totalidade de características de uma entidade que lhe confere a capacidade de satisfazer as necessidades explícitas e implícitas. Esta definição é usada pela norma NBR ISO 8402. Para ajudar no entendimento, uma entidade pode ser uma atividade ou um processo, um produto, uma organização ou uma combinação entre eles.]

Tabela 7 – Exemplo da falta de uma categoria


```
<category>
  <pattern>VOCE SABE O QUE E * </pattern>
  <template>
    <srai><star/></srai>
  </template>
</category>
```

Tabela 8 – Exemplo de categoria criada após o processo de *targeting*

A base de conhecimento do *chatterbot* sofreu inúmeras modificações de acordo com as necessidades encontradas nos *logs* do sistema. As modificações ocorreram principalmente nas primeiras interações com o usuário, onde a base de conhecimento ainda encontrava-se relativamente pequena.

6. Considerações finais

A utilização de robôs de conversação na educação à distância, é benéfica tanto para a instituição que a está usando quanto para o aluno. Essa técnica permite que o aluno tenha liberdade total em seus horários de estudo, evita que o aluno permaneça com alguma dúvida por ficar intimidado em perguntar ao professor e atrai a atenção das pessoas com a possibilidade de conversar com um robô.

A base de conhecimento do *chatterbot* foi desenvolvida obedecendo ao padrão da linguagem AIML. Esse critério foi utilizado para que se possa utilizar a base de conhecimento em qualquer outro interpretador AIML desenvolvido que atenda aos padrões da linguagem AIML. As categorias inseridas na base podem ser divididas em dois tipos: linguagem natural e conhecimento específico.

As categorias que contém a linguagem natural são as responsáveis pela capacidade de conversação do *chatterbot*. São essas categorias que dão ao *chatterbot* o seu principal atrativo, a possibilidade de se passar por um humano, e também são essas categorias que irão apresentar ao usuário os assuntos dominados pelo *chatterbot*. É através de uma simples troca de frases, que o usuário se sentirá a vontade para conversar com um programa que se comporta como um humano e estará sempre disponível para atender as suas necessidades.

O conhecimento específico do *chatterbot* é baseado totalmente em qualidade de software. As categorias que contém os conhecimentos específicos são atômicas, pois contém simplesmente a palavra chave e o conceito do assunto abordado.

Sendo o *chatterbot* desenvolvido para a educação, ele tenta direcionar a conversa para assuntos nos quais ele possui conhecimento. Essa técnica foi bem sucedida nas experiências realizadas, pois, sempre que o *chatterbot* não reconhecia alguma entrada ou o usuário tentava sair do escopo real do *chatterbot*, ele retornava uma frase que sutilmente chamava a atenção para assuntos aos quais ele conhecia.

A técnica de *Stemmer* mostrou-se satisfatória, pois, aumentou a capacidade da base de conhecimento do *chatterbot* sem que fossem necessárias alterações na mesma.

A implantação do *chatterbot* para teste foi feita através da distribuição do programa. Foram selecionados usuários com conhecimentos em qualidade de software, e usuário que não possuíam tal conhecimento. Após a utilização do *chatterbot*, os usuários enviaram os arquivos contendo o *log* sistema para análise.

Através das análises dos arquivos de *log*, notou-se que alguns dos usuários que possuíam conhecimento na área de qualidade de software tentavam extrair informações do *chatterbot* sem

uma conversa, ou seja, entravam diretamente com um tópico, como “qualidade” e não como era de se esperar “o que é qualidade?”. Como a base de conhecimento do *chatterbot* foi desenvolvida através de tópicos, não houve problemas, pois as respostas eram apresentadas corretamente.

Os usuários que não possuíam conhecimento em qualidade de software voltaram-se para a conversa informal com o *chatterbot*. De acordo com o andamento da conversa, não restava outra opção ao usuário a não ser conversar com o *chatterbot* sobre qualidade de software.

A cada arquivo de *log* analisado, várias categorias de conhecimento foram criadas, em sua maioria, categorias de linguagem natural. Através dos *logs* também foi possível perceber uma carência em informações sobre o próprio programa, como nome, sexo e idade, pois já que o *chatterbot* se relacionava como um humano, ele necessita de um mínimo de informações sobre ele mesmo. Várias categorias sobre qualidade de *software* também foram adicionadas.

Pode-se observar que, somente por meio da utilização intensa do *chatterbot*, por pessoas conhecedoras do assunto e pessoas que não possuem tal conhecimento, é que será possível criar uma base de conhecimento extensa o suficiente para possibilitar uma conversa mais agradável e produtiva entre o usuário e o *chatterbot*.

7. Referência bibliográfica

ABED. **Associação Brasileira de Educação a Distância**. 200-. Disponível em: <<http://www.abed.org.br/>>. Acesso em 28 agosto 2004.

ALICE. **The A. L. I. C. E. Artificial Intelligence Foundation**. 200-. Disponível em: <<http://www.alicebot.org>>. Acesso em: 27 agosto 2004.

BEARDON, Colin; HOLMES, Geoff; LUNSDEN, David. **Natural Language and Computational Linguistics**. Tradução 1991. Disponível em: <<http://puhrs.campus2.br/~annes/BEARDON.DOC>>. Acesso em: 8 outubro 2004.

BRAHM et al. **Um Assistente Digital Para Responder Automaticamente Perguntas De Usuários Humanos Em Portais Corporativos**. p. 08, 2003. Disponível em: <<http://www.adsdigital.com.br/artigos.php>>. Acesso em: 12 maio 2005.

CHEONG, Fah-Chun. **Internt Agents: Spiders, Wanderers, Brokers, and Bots**. Indianapolis: New Riders Publishing, 1996.

GARNER, Robby. **CyberMecha Studio**. 1999. Disponível em: <<http://www.robtron.com/>>. Acesso em: 27 agosto 2004.

LAVEN, Simon. **The Simon Laven Page**. 2001. Disponível em: <<http://www.simonlaven.com>>. Acesso em: 28 agosto de 2004.

LEONHARDT et al. **ELEKTRA: Um Chatterbot para Uso em Ambiente Educacional**. 2003. Disponível em: <<http://www.cinted.ufrgs.br/renote/set2003/artigos/elektra-chatterbot.pdf>>. Acesso em: 29 agosto 2004.

LESNICK, L. L.; MOORE, R. E. **Creating Cool Intelligent Agents For The Net**. Foster City: IDG Books Worldwide, Inc.,1996.

MENEZES, Sérgio; TEIXEIRA, Crediné S. **Facilitando o uso de Ambientes Virtuais através de Agentes de Conversação.** 2003. Disponível em: <<http://www.multicast.com.br/sergio/amcorabot.pdf>>. Acesso em: 27 agosto 2004.

NORVIG, Peter; RUSSELL, Stuart. **Artificial Intelligence: A Modern Approach.** New Jersey: Prentice Hall, 1995

OLIVEIRA, Fabio A. D. **Processamento de linguagem natural:** princípios básicos e a implementação de um analisador sintático de sentenças da língua portuguesa. 200-. Disponível em: <<http://www.inf.ufrgs.br/procpar/disc/cmp135/trabs/992/Parser/parser.html>>. Acesso em: 8 outubro 2004.

PANESSI, Walter; BORDIGNON, Fernando Raúl Alfredo. **Procesamiento de Variantes Morfológicas en Búsquedas de Textos en Castellano.** p. 01, 2003. Disponível em: <<http://www.tyr.unlu.edu.ar/TYR-publica/Varia-Morfo.pdf>>. Acesso em 9 julho 2005.

PRIMO et al. **Júnior, um chatterbot para educação a distância.** 200-. Disponível em: <<http://www.c5.cl/ieinvestiga/actas/ribie2000/papers/255/>>. Acesso em: 28 agosto 2004.

WALLACE, Richard S. **AIML Overview.** 200-. Disponível em: <<http://www.pandorabots.com/pandora/pics/wallaceaimltutorial.html>>. Acesso em: 27 agosto 2004.