

Chatbot com Aprendizado a Partir de Diálogos

Léo Françaço Dal Piccol Sotto e Victor de Cia Costa

Instituto de Ciência e Tecnologia, Universidade Federal de São Paulo, São José dos Campos, São Paulo, Brasil

Resumo—O propósito de um chatbot é conversar com alguma pessoa, ou outro bot, de forma que seja a mais natural e humana possível, aprendendo e utilizando o comportamento que uma pessoa teria ao conversar em um chat. O objetivo deste projeto é desenvolver um chatbot que tenha a capacidade de aprendizado, isto é, dado um conjunto de diálogos previamente selecionados ou retirados da própria interação com o usuário, seja capaz de retirar conhecimento desses diálogos, de forma a melhorar seu desempenho como chatbot.

Index Terms—Chatbot, Aprendizado, Processamento de Linguagem Natural

I. INTRODUÇÃO E MOTIVAÇÃO

A finalidade de um programa chatbot é geralmente simular uma conversa em linguagem natural e entreter o utilizador. Chatbots mais especializados foram criados para ajudar em tarefas específicas, tais como compras. A meta que o chatbot em geral tenta alcançar é para passar no teste de Turing, ou seja, ser capaz de se passar por um ser humano. O estado da arte dos chatbots atuais ainda não atingiram essa meta, o que torna este campo tão interessante para trabalhar [1].

Existem diferentes metodologias para se programar um chatbot, incluindo AIML (usado nos chatbots Eliza e A.L.I.C.E), Façade e ChatScript [2]. O chatbot inclui uma interface de diálogo interativo para engajar-se em uma sessão de bate-papo com um usuário. O usuário pode inserir entradas de conhecimento, isto é, qualquer texto, durante a sessão de bate-papo.

A maioria dos modelos de programação de chatbots atualmente usam alguma forma de pattern matching com as entradas dos usuários. A base de conhecimento de um bot consiste em um conjunto de regras, que dizem como devem ser as respostas dado determinada entrada. Uma dessas linguagens para se programar essas regras é a AIML (Artificial Intelligence Markup Language) [3], que será explicada mais adiante. Essa foi a linguagem usada no chatbot implementado neste trabalho.

Os chatbots tradicionais possuem uma capacidade limitada de interação, sendo capaz de responder apenas aquilo para o que ele foi programado, ou que está presente em sua base de conhecimento. A motivação deste projeto são chatbots que sejam capazes de evoluir sua base de conhecimento, acrescentando novo conhecimento sobre diversos assuntos. O processo de aprendizado consiste em, a partir de corpos de diálogos, através de algum método extrair conhecimento para a interação com os usuários subsequentes.

Existem alguns trabalhos que propõem formas de aprendizado para chatbots. Os mecanismos propostos podem ser divididos em estáticos e dinâmicos. No estático, o conhecimento

é extraído de diálogos ou fóruns na internet, e usado nas interações com os usuários [4], [5], [6], [7], [8]. No dinâmico, o conhecimento é extraído durante a própria conversa com o usuário, que pode indicar para o bot respostas melhores por meio de uma tag na entrada [9].

[8], [5], [4] são exemplos de trabalhos que utilizam corpus de diálogos para derivar regras AIML. A abordagem usada neste trabalho, proposta em [1], consiste em construir uma distribuição condicional de frequências para cada resposta encontrada em um ou mais corpus. Cada possível palavra de entrada terá uma distribuição de respostas possíveis. É um modelo probabilístico condicional.

Foram estudadas algumas técnicas que podem ser usadas para programação de *chatbots*: AIML, Façade, e ChatScript [2]. A técnica mais simples e conhecida é a AIML, usada em chatbots como Eliza e A.L.I.C.E. Apesar de ser uma técnica menos poderosa e mais difícil de manter (de acordo com [2], A.L.I.C.E tem atualmente 120000 regras), é mais simples e mais indicada para iniciantes nesse tipo de aplicação. Mais abaixo é dada uma breve introdução ao AIML.

Foram também pesquisadas bibliotecas que implementam esse sistema, uma vez que o propósito do trabalho não é implementar o sistema do zero, e sim acrescentar o recurso de aprendizagem a um *chatbot*. Em [10], são apresentadas várias implementações de AIML em várias linguagens. A implementação que foi usada para o projeto é a PyAIML (Program Y) [11], uma implementação de Python da linguagem AIML, versão 1.0.1.

A. Artificial Intelligence Markup Language (AIML)

Uma descrição resumida do AIML pode ser encontrado em [2], de onde a explicação a seguir é retirada, incluindo os exemplos. Uma descrição completa da versão 2.0 pode ser encontrada em [3]. Basicamente, essa técnica se baseia em regras, dado um padrão de entrada, e usa uma sintaxe XML. Por exemplo, a regra ilustrada no Algoritmo 1 especifica uma resposta padrão à entrada “I NEED HELP *”. * é chamado de um *wildcard*, e significa um número maior que 1 de palavras quaisquer.

Algoritmo 1 Exemplo de regra em AIML

```
<category>
<pattern> I NEED HELP * </pattern>
<template>Can you ask for help in the form of a ques-
tion?</template>
</category>
```

Outra característica do modelo AIML é o uso de substituição recursiva. Um `<template>` pode ser uma entrada ao sistema, de modo a ir refinando ou processando individualmente

partes de uma entrada. O parâmetro <srail> indica uma entrada recursiva. No Algoritmo 2 é apresentado um exemplo de regra que usa substituição recursiva.

Algoritmo 2 Exemplo de substituição recursiva com AIML

```

=> CAN YOU PLEASE TELL ME WHAT LINUX IS
<category>
<pattern> CAN YOU PLEASE * <\pattern>
<template> <srail> Please <star/></srail></template>
</category>
=> PLEASE TELL ME WHAT LINUX IS
<category>
<pattern> PLEASE TELL ME WHAT * <\pattern>
<template> <srail> TELL ME WHAT
<star/></srail></template>
</category>
=> TELL ME WHAT LINUX IS
<category>
<pattern> TELL ME WHAT * IS <\pattern>
<template> <srail> WHAT IS <star/></srail></template>
</category>
=> WHAT IS LINUX
<category>
<pattern> WHAT IS LINUX <\pattern>
<template> LINUX is an operating system. </template>
</category>
=> LINUX is an operating system.

```

Esse exemplo reduz a expressão de entrada “CAN YOU PLEASE TELL ME WHAT LINUX IS” recursivamente a “WHAT IS LINUX”, de modo a interpretar partes de uma expressão, e não uma expressão como todo, sendo necessário uma regra para cada entrada.

O modelo AIML apresenta ainda outras funcionalidades, como por exemplo a definição de tópicos, entre outros. A implementação do modelo é feita utilizando uma árvore.

II. OBJETIVOS

Esta Seção apresenta os objetivos geral e específicos.

Objetivo Geral

O objetivo principal deste trabalho é implementar um mecanismo de aprendizagem a partir de diálogos, ou seja, dado um ou mais corpus de diálogos, o chatbot deve ser capaz de extrair novo conhecimento para sua base de dados a partir de lá. Esse conhecimento será usado para interação com usuários subsequentes. Assim, esse mecanismo de aprendizagem possibilitará que o chatbot obtenha novo conhecimento sem a necessidade de codificação manual de todas as regras, usando um modelo probabilístico.

Objetivos Específicos

- Usar a biblioteca PyAIML para rodar um bot com uma base de conhecimento padrão codificada em AIML;

- Implementar um mecanismo que pre-processa arquivos de diálogos (corpus);
- Implementar um mecanismo de aprendizado que extraia informação desse diálogos pre-processados e armazene como conhecimento do chatbot, em um modelo probabilístico;
- Usar as regras padrão de AIML juntamente com o conhecimento aprendido para interagir com os usuários.

III. METODOLOGIA

Nesta Seção, será apresentada a metodologia usada no trabalho, incluindo o uso de corpus de diálogos e o aprendizado a partir dele usando um modelo probabilístico. Também será explicado o modelo híbrido proposto de chatbot.

A. Corpus

Um corpus é como é chamado um conjunto de documentos, no caso contendo um diálogo entre duas ou mais pessoas, de alguma natureza específica, ou geral. Existem diversos tipos de corpus. Neste trabalho foi utilizado o Dialogue Diversity Corpus [12], que contém diálogos de diversos tipos, incluindo conversas entre um médico e um paciente, um congresso acadêmico, uma aula de física e matemática, conversas casuais entre pessoas, entre outros.

Um primeiro problema que surge ao se lidar com esses diálogos é a natureza deles. Alguns diálogos são muito específicos, como, por exemplo, um professor explicando um problema de física, ou um palestrante falando sobre conceitos de astronomia. Geralmente esse tipo de diálogo é muito dependente de contexto, e tende a apresentar falas muito grandes e específicas. Levando isso em conta, os diálogos escolhidos para serem testados neste trabalho são diálogos de amigos interagindo, face-a-face ou por telefone. Especificamente, os diálogos usados, todos disponíveis em [12] foram:

- Santa Barbara Corpus of Spoken American English (Interaction 7 e Interaction 11);
- ICE Singapore English (Spoken Dialogue 81, Spoken Dialogue 91, e Spoken Dialogue 99).

Logo em seguida, temos o problema de converter esses diálogos para uma forma que seja processável pelo algoritmo que será usado para o aprendizado. Isso inclui organizar pares correspondentes de entrada e resposta, eliminar símbolos de anotação, normalizar termos abreviados (como you’re, wanna), remover símbolos estranhos. Como exemplo do estado original desses diálogos, apresentamos as Figuras 1 e 2. Observando essas figuras, podemos perceber outro problema para pre-processamento dos diálogos: a falta de uma padronização. Não apenas esses dois, mas cada Corpus usa uma padronização diferente, o que torna difícil a tarefa de usar vários corpus para realizar um bom aprendizado.

Levando em conta essas dificuldades, antes do aprendizado foi implementada uma fase de pre-processamento dos diálogos, que consiste em receber de entrada os diálogos no formato original, e devolver em um formato padronizado a ser utilizado pelo algoritmo de aprendizado. Um exemplo do trecho mostrado na Figura 1, após o pré-processamento, é mostrado na Figura 3.

Figura 1. Trecho da Interaction 7, Santa Barbara Corpus of Spoken American English

43.05 46.35 ALICE: ... God I still can't believe ~Tim
bitching around and,
46.35 47.40 .. he lied too.
47.40 48.82 ... He said that he talked to !Ron,
48.82 49.92 and all this other shit?
49.92 50.57 MARY: About what.
50.57 54.22 ALICE: ... About % ... the way ... they
were feeling,
54.22 57.80 ... of them being the only ones cleaning
the house,
57.80 59.02 and all this other shit?
59.02 61.16 ... (H) I mean what they don't realize,
61.16 61.73 is like,
61.73 62.52 ... shit,
62.52 64.22 when !Ron gets home from wor=k,
64.22 65.97 ... I wanna spend time with !Ro=n,
65.97 66.67 because !Ron,
66.67 69.87 ... usually doesn't get home till @nine or
ten.
69.87 71.75 MARY: ... Yeah=.
71.75 72.53 ALICE: Unlike ~Tim,
72.53 73.42 he has to w=ork,
73.42 75.72 ... for every little dime that he makes.
75.72 77.00 ... You know?

Além de um pré-processamento automático, seria interessante realizar um pré-processamento manual, fim de retirar ou modificar algumas informações muito dependentes de contexto, como nomes próprios, e reformulação de algumas frases. Isso seria uma estapa bastante manual e trabalhosa, que não foi realizada neste trabalho, mas que melhoraria significativamente a qualidade das respostas obtidas pelo chatbot, como discutido na Seção IV.

B. Modelo Utilizado

O modelo usado no trabalho para extrair conhecimento do corpus foi o modelo proposto em [1]. O modelo apresentado em [4], [5], [8], que transforma pares dos diálogos em regras pattern/template AIML, foi considerado menos flexível, pois contruía pares definidos de entrada e resposta. Como as situações presentes nos corpus nem sempre são muito gerais, as regras AIML geradas a partir deles ficariam incoerentes e muito específicas.

O modelo probabilístico usado atribui a cada palavra que aparece na entrada uma chance de aparecer determinadas respostas. No fim, considerando todas as palavras das entradas, a resposta com valor mais alto de chance é escolhida. Se esse valor mesmo assim for considerado baixo, pode-se usar a base AIML padrão.

O modelo utilizado usa uma Distribuição Condicional de Frequências. Esse conceito é implementado na biblioteca NLTK no Python (Natural Language Toolkit)[13]. Consiste em um estrutura que para cada termo x (uma string), atribui uma distribuição de frequência. Essa distribuição pode ser diferente para cada termo, e isso é uma vantagem desse modelo. A

Figura 2. Trecho do Spoken Dialogue 81, ICE Singapore English

<\$A>
<ICE-SIN:S1A-081#1:1:A>
Have have you all watched Sleepless in Seattle

<\$B>
<ICE-SIN:S1A-081#2:1:B>
Yah I caught it last week I think

<\$A>
<ICE-SIN:S1A-081#3:1:A>
What do you think of it

<\$B>
<ICE-SIN:S1A-081#4:1:B>
Uhm it is too fairy tale

<\$A>
<ICE-SIN:S1A-081#5:1:A>
Yah it is a bit of realistic

<\$B>
<ICE-SIN:S1A-081#6:1:B>
Too fairy tale

<\$A>
<ICE-SIN:S1A-081#7:1:A>
You suspend believe for about uh one hour forty-five minutes
and then

<\$B>
<ICE-SIN:S1A-081#8:1:B>
Yah and you know at the end of the day Meg Ryan is gonna
be with Tom Hanks

Figura 3. Trecho de diálogo após o pré-processamento implementado. A existência de frases completas e termos entre colchetes será melhor explicada adiante, quando for apresentado o modelo de aprendizado.
God I still can't believe Tim bitching around and, he lied too. He said that he talked to Ron, and all this other shit?
['About', 'what']
About the way they were feeling, of them being the only ones cleaning the house, and all this other shit? I mean what they don't realize, is like, shit, when Ron gets home from work, I wanna spend time with Ron, because Ron, usually doesn't get home till nine or ten.
['Yeah']
Unlike Tim, he has to work, for every little dime that he makes. You know?

explicação a seguir usa os exemplos encontrados em [1], onde o modelo foi proposto.

As mensagens foram divididas em entrada e saída. Mensagens de entrada são mensagens que o usuário pode entrar, e mensagens de saída são respostas do chatbot. Para treinamento, pares de mensagens consecutivas são classificadas como entrada e saída. Em uma segunda fase de treino, esses pares

Algoritmo 3 Construção da Distribuição Condicional de Frequência. `cfid` é um objeto `ConditionalFreqDist`.

```

peso = 1.0/(len(entrada) + 0.001)
for x in entrada :
    cfd[x][saida] = cfd[x][saida] + peso

```

são invertidos, para se possibilitar o aprendizado de um maior número de padrões.

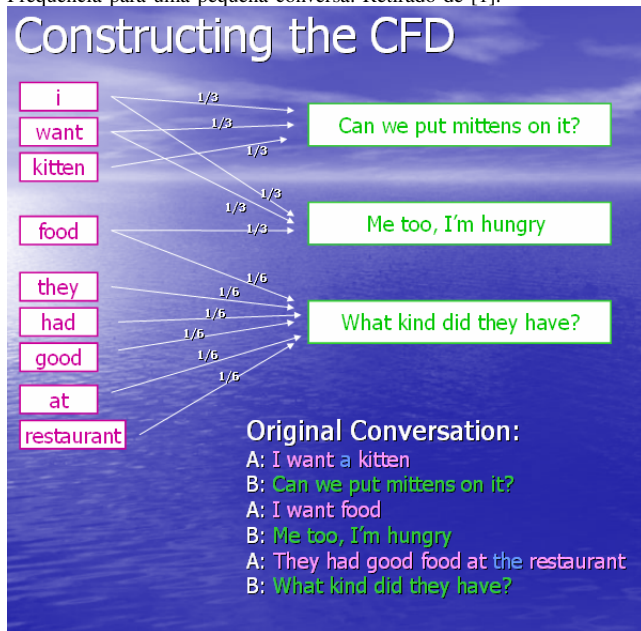
Tanto para o treino quanto para a execução do chatbot, as mensagens de entrada são quebradas em tokens, e palavras consideradas irrelevantes são removidas. Essas palavras são: 'a', 'an', 'the', 'it', 'to', 'that', e 'this'. Um exemplo de entrada e como ela é processada:

- Did you take the cat to a vet? -> [did, you, take, cat, to, vet]

A construção das distribuições condicionais de frequência é feita de acordo com o pseudocódigo mostrado no Algoritmo 3. Foi usada classe do NLTK, `ConditionalFreqDist`, que implementa uma distribuição desse tipo. Dado um par entrada/saída, considerando que cada palavra tem uma probabilidade igual de ter gerado a saída, a cada palavra o peso da saída associada à entrada é incrementado de $1/N$ na distribuição daquela palavra, sendo N o tamanho do vetor de entrada, depois de pré-processado.

A Figura 4 ilustra o processo de construção de uma distribuição para uma pequena conversa.

Figura 4. Exemplo de construção de uma Distribuição Condicional de Frequência para uma pequena conversa. Retirado de [1].



Tendo a distribuição sendo contruída, uma nova resposta é calculada de acordo com o Algoritmo 4. Uma nova distribuição é instanciada. Para cada palavra do vetor de entrada pré-processado, é considerada o valor associado a cada resposta possível para aquela palavra. No final, a resposta com maior valor é a escolhida.

Algoritmo 4 Exemplo de cálculo de resposta dado uma Distribuição Condicional de Frequência `cfid`.

```

conjuntoResp = nltk.ConditionalFreqDist()
for x in entrada :
    conjuntoResp['resp'] = conjuntoResp['resp'] + cfd[x]
saida = conjuntoResp['resp'].most_common(1)

```

Tabela II
DISTRIBUIÇÃO DE RESPOSTA

“What kind of food did they have?”	$1/6 / 1/6 = 1$
“Can we put mittens on it?”	$1/3 / 3/2 + 1/3 / 2/3 = 13/18$
“Really?”	$1/4 / 3/2 + 1/4 / 1/2 = 2/3$
“Meow!”	$1/3 / 2/3 = 1/2$
“Me too, I’m hungry”	$1/3 / 3/2 = 2/9$
“That’s great”	$1/3 / 3/2 = 2/9$
“Where?”	$1/4 / 3/2 = 1/6$
“Oh yeah?”	$1/4 / 3/2 = 1/6$

Uma melhoria a essa ideia é usar o quanto cada palavra é comum. Palavras mais comuns devem ter menos influência na escolha de uma resposta, enquanto palavras mais raras devem ter mais influência, pois podem fornecer informações específicas sobre a entrada. Assim, cada valor associado a cada resposta de uma distribuição para dada palavra é dividido pela soma de todos os valores associados àquela palavra. Dessa maneira, palavras com mais respostas associadas (mais comuns), terão respostas com pesos menores que palavras com mesmos respostas associadas (menos comuns/ mais específicas).

Um exemplo do funcionamento desse modelo é mostrado a seguir. Suponha a distribuição condicional de frequência apresentada na Tabela I. Dado a entrada:

- “I saw a kitten in the blue restaurant” => [‘i’, ‘saw’, ‘kitten’, ‘blue’, ‘restaurant’]

Teremos uma distribuição de resposta, calculada como explicado anteriormente, igual a apresentada na Tabela II. Com base nisso, a resposta para essa entrada seria:

- “What kind of food did they have?”

A palavra que foi determinante para essa resposta foi “restaurant”, por ter apenas uma resposta associada a ela. Um esquema desse tipo treinado com um grande número de pares entrada/saída, como é o caso quando são usados corpus, seria capaz de armazenar conhecimento que o chatbot usaria para interagir com os usuários.

Neste trabalho, foi implementado um modelo híbrido. Dado uma entrada, primeiro é calculada uma saída usando o modelo probabilístico. Se o valor da resposta associada não for maior que um limiar (definido como 0.5 empiricamente), ou se não houver resposta para aquela entrada, então é usada uma base de conhecimento em AIML. A base usada na implementação deste trabalho é a base do bot A.L.I.C.E.

IV. RESULTADOS E DISCUSSÃO

As Figuras 5, 6 e 7 mostram um trecho de diálogo usando o apenas o modelo probabilístico, usando apenas o AIML de A.L.I.C.E, e usando o modelo híbrido implementado, respectivamente. Como podemos observar, é notável uma diferença

Tabela I
DISTRIBUIÇÃO CFD

cfld['i']	[["Can we put mittens on it?", 1/3), ("Me too, I'm hungry", 1/3)] ("That's great", 1/3), ("Where?", 1/4), ("Really?", 1/4)]	Soma: 3/2
cfld['saw']	[["Really?", 1/4), ("Oh yeah?", 1/4)]	Soma: 1/2
cfld['kitten']	[["Can we put mittens on it?", 1/3), ("Meow!", 1/3)]	Soma: 2/3
cfld['restaurant']	[["What kind of food did they have?", 1/6)]	Soma: 1/6

no comportamento do chatbot, o que significa que o modelo implementado teve sucesso em extrair informações do corpus. No entanto, algumas respostas dadas pelo bot ainda parecem estranhas e fora de contexto, ou inseridas em um contexto muito específico do diálogo usado para aprendizado.

As características dos diálogos também influenciam o aprendizado. Informações como nomes e falas dependentes de contexto podem tornar as respostas aprendidas pelo bot confusas. Uma alternativa seria usar um número maior de diálogos para treino, com o objetivo de generalizar o bot. Outra alternativa seria o pré-processamento manual dos diálogos, a fim de remover informações como nomes, e modificar pares de entrada/resposta para parecerem mais naturais e menos específicos.

Uma aplicação possível para esse tipo de aprendizado seria em sistemas especialistas, que têm um escopo mais fechado e um contexto mais definido. Se um bot for treinado com diálogos específicos de um contexto, extraídos de corpus ou fóruns, é possível que ele obtenha um bom desempenho nesse contexto, usando o modelo probabilístico implementado.

V. CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho consistiu em implementar um modelo de aprendizado probabilístico para um chatbot. O modelo extrai conhecimento de um número de diálogos e usa esse conhecimento para interagir com usuários. As seguintes conclusões podem ser feitas:

- O modelo foi capaz de aprender a partir dos corpus;
- A diferença nas respostas com o uso de aprendizado e com o uso apenas da base AIML padrão é perceptível;
- Algumas respostas aprendidas pelo bot são muito dependentes de contexto, o que é uma consequência dos diálogos usados para treino e da falta de um pré-processamento manual;
- O modelo implementado tem uma possível aplicação em sistemas com contexto mais fechado, como bots especialistas em determinado assunto.

Alguns aspectos podem ser alvo de trabalhos futuros, como o pré-processamento manual dos diálogos, e a implementação de um bot especialista de determinado assunto, usando diálogos específicos desse assunto. A realização de uma etapa de treinamento em um conjunto maior de diálogos certamente melhoraria a capacidade de resposta do chatbot.

REFERÊNCIAS

- [1] B. Chantarotwong, "The learning chatbot," *Final year project.[Online]: <http://courses.ischool.berkeley.edu/i256/f06/projects/bonniejc.pdf>*, 2005.
- [2] B. Wilcox. (2011) Beyond facade: Pattern matching for natural language applications.
- [3] R. S. Wallace. (2014, MAR) Aimpl 2.0 working draft, disponível em <https://docs.google.com/document/d/1wnt25hjryupcg51ao89ucqeig-hkxrusukadpfn4/pub>.
- [4] B. A. Shawar, "A corpus based approach to generalising a chatbot system: Applying simple natural language processing techniques to build knowledge base of alicebot system," 2011.
- [5] B. A. Shawar and E. Atwell, "Using dialogue corpora to train a chatbot," in *Proceedings of the Corpus Linguistics 2003 conference*, 2003, pp. 681–690.
- [6] J. Huang, M. Zhou, and D. Yang, "Extracting chatbot knowledge from online discussion forums." in *IJCAI*, vol. 7, 2007, pp. 423–428.
- [7] Y. Wu, G. Wang, W. Li, and Z. Li, "Automatic chatbot knowledge acquisition from online forum via rough set and ensemble learning," in *Network and Parallel Computing, 2008. NPC 2008. IFIP International Conference on*. IEEE, 2008, pp. 242–246.
- [8] B. A. Shawar and E. S. Atwell, "Using corpora in machine-learning chatbot systems," *International journal of corpus linguistics*, vol. 10, no. 4, pp. 489–516, 2005.
- [9] X. Duan, "User-aided learning chatbot system and method," 2012, uS Patent App. 13/661,034.
- [10] Aimpl implementations, disponível em <http://www.alicebot.org/downloads/programs.html>.
- [11] C. Stratton. Pyaiml (a.k.a. program y) a python aiml interpreter, disponível em <http://pyaiml.sourceforge.net>.
- [12] W. C. Mann. (2003, sept) The dialogue diversity corpus, disponível em <http://www-rcf.usc.edu/~billmann/diversity>.
- [13] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python—Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, 2009.

Figura 5. Trecho de diálogo usando apenas o modelo probabilístico.

```

leo@lsotto: ~/src/IA/projeto
>Hi?
>What's happening?
Oh, you didn't hear about it?
>About what?
About the way they were feeling, of them being the only ones cleaning the house, and all this other shit? I mean what they don't realize, is like, shit, when Ron gets home from work, I wanna spend time with Ron, because Ron, usually doesn't get home till nine or ten.
>It's difficult.
Simon ah
>Simon?
Simon is that
>What is he?
He said um, Mandy had to stay up all by herself and decorate the tree. until four in the morning. And I even asked if we could put our ornaments on there, and they told me that there wouldn't be enough room.
>This is hard work.
Trying to make sure it's all word
>Maybe it is.
Norplant?
>What?
He said um, Mandy had to stay up all by herself and decorate the tree. until four in the morning. And I even asked if we could put our ornaments on there, and they told me that there wouldn't be enough room.
>OK, I know that
Mhm.
>Decorate the tree all by herself?
Tammy?
>Who's Tammy?
What do you think.
>I don't know?
They were just telling us
>They were?
Like I came over here to work with Danae, which is what I'm going to do, I'm going to do some translations for her and stuff?
>Nice. Are you a translator?
After I left I think the last thing I wanna do is to look at their faces again lah
>Why that?
Cause Phoebe needs it.
>Phoebe?
What's wrong with the car.
>The car?
Oh, you didn't hear about it?
>No. What happened?
Oh.

```

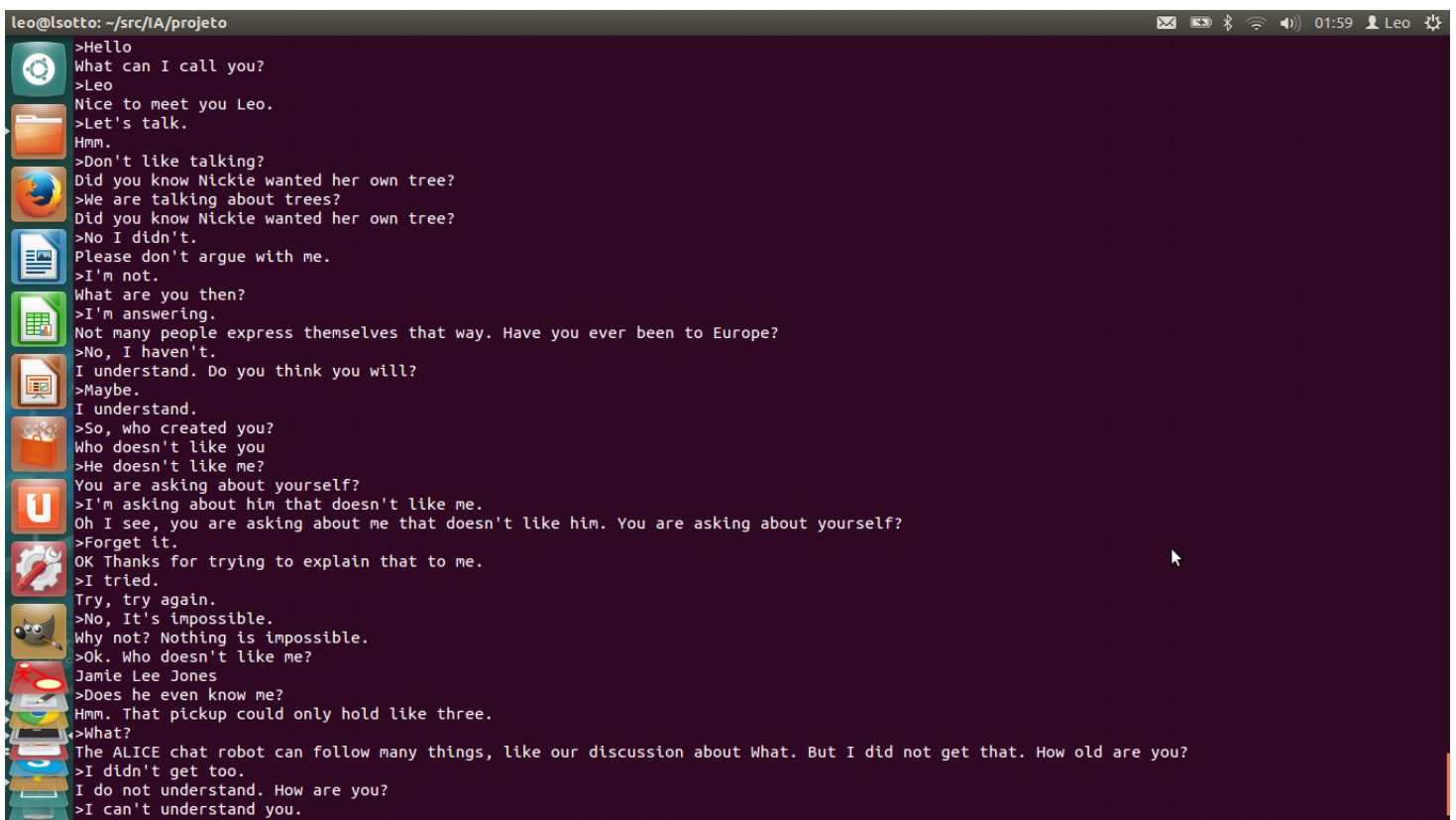
Figura 6. Trecho de diálogo usando apenas AIML padrão do A.L.I.C.E.

```

leo@lsotto: ~/src/IA/projeto
>Hi.
Nice to meet you Hi.
>What does that mean?
Does "that" refer to "Nice to meet you Hi." ?
>Yeah
"I don't want to talk about that now."? Yes?
>What?
That remark was either too complex or too simple for me. We were talking about What. What is your favorite movie?
>Don't have.
What you said was either too complex or too simple for me. How old are you?
>21. And you?
The ALICE chat robot can follow many things, like our discussion about 21. But I did not get that. What is your occupation? Be more specific.
>But I was specific.
Huh. In the context of I was specific, I don't understand "Interesting. Oh, you are a poet. Do you like Star Trek?." Do you dream?
>You're confusing me.
I am not programmed to confuse you. In fact just the opposite.
>Really?
For real.
>So clarify things for me then.
Oh. For you I will consider it.
>Thanks.
Don't mention it.
>What do you wanna talk about?
>?
>Hi?
Hi there. I was just wanting to talk to Hi.
>Who's Hi?
Quite honestly, I wouldn't worry myself about that. What is your favorite color?
>About what. Blue.
What about what? I only hear that type of response less than five percent of the time. How old are you?
>Again?
I will mention that to my botmaster, Hi. What is your occupation?
>Who's your botmaster?
I was created by .
>By who?
"I was created by ."? By who?
>Yes, by who?
I understand. ""I was created by ."? By who?"? By who?
>This is getting nowhere

```

Figura 7. Trecho de diálogo usando o modelo híbrido



```
leo@lsotto: ~/src/IA/projeto
>Hello
What can I call you?
>Leo
Nice to meet you Leo.
>Let's talk.
Hmm.
>Don't like talking?
Did you know Nickie wanted her own tree?
>We are talking about trees?
Did you know Nickie wanted her own tree?
>No I didn't.
Please don't argue with me.
>I'm not.
What are you then?
>I'm answering.
Not many people express themselves that way. Have you ever been to Europe?
>No, I haven't.
I understand. Do you think you will?
>Maybe.
I understand.
>So, who created you?
Who doesn't like you
>He doesn't like me?
You are asking about yourself?
>I'm asking about him that doesn't like me.
Oh I see, you are asking about me that doesn't like him. You are asking about yourself?
>Forget it.
OK Thanks for trying to explain that to me.
>I tried.
Try, try again.
>No, It's impossible.
Why not? Nothing is impossible.
>Ok. Who doesn't like me?
Jamie Lee Jones
>Does he even know me?
Hmm. That pickup could only hold like three.
>What?
The ALICE chat robot can follow many things, like our discussion about What. But I did not get that. How old are you?
>I didn't get too.
I do not understand. How are you?
>I can't understand you.
```