Dissertação de Mestrado em colaboração com a Accenture

Machine Learning e Processamento de Linguagens Naturais num Bot

Carlos António Senra Pereira
Orientador: Doutor Luís Filipe Ribeiro Pinto
Co-Orientador: Doutor José João Antunes Guimarães Dias de Almeida
Orientador na Accenture: Ricardo Perleques
Orientador na Accenture: Hugo André Portela

26 de Junho de 2018

Conteúdo

1 Resumo

Resumo do Bot.

2 Introdução

Com a capacidade de hardware cada vez mais evoluída e problemas quer da parte cientifica como problemas do quotidiano, Machine Learning tem sido vista como uma técnica possível para a resolição desses mesmos problemas.

O processamento de Linguagens Naturais é bastante complexo, e ainda mais no ponto de vista de uma máquina determinística. Por isso, utilizando técnicas de Machine Learning é possível construir um modelo capaz de analisar frases morfossintáticamente com bastante precisão.

Através desta análise, podemos tirar grande partido dela no interpretamento das mesmas e assim executar ações correspondentes às frases.

2.1 Objetivos

1. Construir uma aplicação capaz de analisar e reconhecer frases introduzidas por utilizadores;
2. Através de um modelo de Machine Learning, analisar essas frases;
3. Utilizando técnicas de Machine Learning em conjunto com bases de dados de conhecimento, criar um modelo capaz de classificar uma análise e atribuir uma ação.
4. Construir relações entre sujeito e objeto e através dessas mesmas, gerar todas(ou quase todas) as expressões regulares correspondentes.

3 Descrição abstrata do modelo de execução

Após a análise do problema em questão, foi criado um modelo de execução de um Bot.

$$DEF_{\text{Bot}} ::= Regra^*$$

$$\begin{aligned} Regra ::= antecedente &: ERL \\ \text{reação} &: FR^* \end{aligned}$$

Onde FR abrevia Função de Reação e ERL significa Expressão Regular Linguística.

Com este modelo, é possível escrever uma álgebra de Bot's que permite o uso de vários Bot's na mesma solução, por exemplo, em paralelo, em sequência, etc.

3.1 Expressões Regulares Linguísticas

Estas são as expressões definidas pelo utilizador que irão ser compiladas e interpretadas pelo Bot.

$$
\begin{aligned}
ERL &::= EL^* \\
EL &::= IT \quad FR^* \quad | \quad IT2 \quad FR^* \\
IT &::= (pal : ER, \quad tag : ER, \quad catch : ER) \\
IT2 &::= (\text{“relax”}, \quad tag : ER, \quad catch : ER) \\
FR &::= fun : ER
\end{aligned}
$$

Onde IT e IT2 significam Item e ER significa Expressão Regular.
Os campos *pal*, *tag* e *catch* são expressões regulares com o significado de palavra, etiqueta e capturar, respetivamente.

Com estas expressões regulares, podemos contruir expressões simples, onde o Bot irá compilar em expressões regulares ordinais com todas as propriedades definidas.

3.2 Função verifica

No momento em que o utilizador introduz uma frase, o Bot terá de verificar se a frase introduzida tem correspondência. É nesse momento que a função verifica atua.

$$verifica(frase, EL) \mapsto (id \hookrightarrow val) \quad | \quad \bot$$

Esta função aceita uma frase e uma expressão regular, e devolve um resultado de dois possíveis: um dicionário com o id e valor das expressões que se captaram; ou dá vazio quando a expressão não corresponde à frase.

3.3 Função executa

Esta função é a função mostra o funcionamente do Bot.

$$
\begin{aligned}
&executa(DEF_{\text{Bot}}, str)\\
&\quad for(antecedente, \text{reação} \in DEF_{\text{Bot}})\\
&\qquad v = verifica(str, antecedente)\\
&\qquad r1 = sortear(\text{reação})\\
&\qquad r1(v)
\end{aligned}
$$

4 Modelo V2

Após uma análise ao modelo anterior, foram feitas algumas alterações para simplificar a representação e implementação do modelo. Segue, então, abaixo o novo modelo linguístico:

$$
\begin{aligned}
ERL &::= EL^*\\
EL &::= EL \quad \text{“/”} \quad IT \quad ?^*\\
ER &::= IT \quad | \quad (\backslash w = IT)\\
IT &::= \$\backslash w \quad | \quad \backslash w
\end{aligned}
$$

Onde,

$$
\begin{aligned}
ER &\leftarrow \text{Expressão Regular}\\
IT &\leftarrow \text{Item}\\
\backslash w &\leftarrow [a-zA-Z0-9]^*\\
ERL &\leftarrow \text{Expressão Regular Linguística}\\
.^* &\leftarrow \backslash w^*
\end{aligned}
$$

Explicação:

- $.^*/.^*$: significa que podemos ter qualquer coisa de um tipo qualquer;
- $.^*/\$verbo$: significa que podemos ter qualquer coisa de um tipo chamado *verbo*;
- $(lugar = .^*)/\$cidades$: significa que podemos ter qualquer coisa, que vai ser guardada numa variável chamada de *lugar*, captura de variáveis, do tipo cidades;
- $.^*/.^*?$: significa que podemos ter ou nao qualquer coisa de um tipo qualquer.

Estes tipos, estão definidos num mapa de subtituição que é utilizado pelo compilador do Modelo Linguístico.

Um exemplo de uma expressão deste modelo é:

$$.^*/pronome\_geral \quad .^*/verbo\_geral \quad (name = .^*)/nome\_proprio\_geral \quad ?/Fit? \quad (1)$$

Nesta expressão fazemos a correspondência de um pronome qualquer seguindo um verbo qualquer, onde de seguida capturámos um nome próprio para a variável *name* e podemos ter ou não o sinal de pontuação ?

5 Arquitetura

5.1 FreeLing

FreeLing é uma biblioteca escrita em C++ que analisa frases de linguagens naturais morfossintáticamente. Com esta biblioteca, é possível analisar qualquer frase e obter uma análise com grande probabilidade de estar correta, pois é utilizado um modelo de Machine Learning que foi treinado com o *Bosque* da *Linguateca*.

Este *Bosque* contém milhares de frases analisadas por Linguístas, que o torna um excelente conjunto de treino para qualquer modelo de análise morfossintática.

5.1.1 Servidor

Devido ao facto do modelo de análise do *FreeLing* demorar bastante tempo a ser construído, implementou-se um pequeno servidor que tem como objetivo gerar todas as variáveis necessárias para o funcionamento do *FreeLing* e fornecer também ao utilizador uma consola para gestão e teste rápido das funcionalidades do servidor.

5.1.2 Cliente

Cada instância que precise de utilizar recursos do *FreeLing*, utilizam-se *sockets* para se ligar ao servidor e utilizar o protocolo *JSON* para enviar e receber informação.

5.1.3 Wrapper

5.2 Main

5.3 Ações

5.4 Estado

POR FAZER