

Geração de Linguagem Natural com Earley

INF05005 Linguagens Formais e Autômatos - Trabalho Prático

Profa. Aline Villavicencio Parsing e Geração de Linguagem Natural

Andy Ruiz
João Vitor de Camargo
Leonardo Vianna
Marlize Ramos
Matheus Rhoden

O trabalho final de geração de linguagem natural, adaptando o algoritmo de Earley, foi separado em alguns arquivos e/ou funcionalidades:

[Input.py](#) :

Possui os métodos responsáveis por ler o arquivo de entrada .txt conforme o padrão do [sample.txt](#) e o [gramatica.txt](#) .

[TerminalClass.py](#) e [variableClass.py](#) :

As variáveis e terminais foram consideradas como classes no nosso trabalho para lidar melhor com elas e funcionar de forma mais organizada e clara.

[generator.py](#) :

O principal arquivo e que poderia ser considerado o “main”. Nele é processada a gramática, vinda do [input.py](#) e aplicado o Earley adaptado para gerar frases, ao invés de fazer o reconhecimento.

Como adicional no nosso trabalho, fizemos ele devolver a frase em forma de Webservice **, podendo ser acessada externamente. Juntamente, desenvolvemos um aplicativo Android que manda um request para esse Webservice e mostra a frase gerada na tela do App e lê ela com a voz do Google.

[offline_generator.py](#) :

O generator.py é executado apenas no nosso webservice, que fica hospedado externamente conforme explicado na seguinte página ***. Com isso, caso se deseje apenas rodar o Gerador de linguagem localmente desde um computador, sem usar um celular Android, disponibilizamos essa versão para ser usada “offline” ou sem o App.

[funsample.txt](#) :

É nossa gramática principal, focada em frases motivacionais e cômicas ao mesmo tempo, que serão geradas e lidas ao usuário no nosso App e permitirá ser compartilhada no WhatsApp.

[sample_old.txt](#) :

Foi a primeira gramática que submetemos no Moodle. Na época ainda não havíamos tido a ideia das frases engraçadas, então havíamos feito um gerador de lero lero ou enrolação : frases sem nexos apenas para falar para uma pessoa quando não tivéssemos o que dizer mas a intenção fosse puxar assunto.

[sample.txt](#) :

Gramática de teste postada no Moodle com recursão na palavra inicial.

[parser.py](#) :

O reconhecedor de linguagem natural a partir de uma gramática passada em .txt , usando também o [input.py](#) e retornando uma mensagem na tela se RECONHECEU ou NÃO RECONHECEU

A proposta do nosso trabalho foi você baixar o emotivApp e receber lindas frases que farão você rir e melhorar seu ânimo. E pode enviar/compartilhar elas no Zap. Não deixe de testar nosso App!

Embora possa testar localmente apenas rodando o [offline_generator.py](#) :(

**

Caso quisesse testar o WebService e toda a funcionalidade junto com o Android, precisa instalar o web.py através do PIP ou <http://webpy.org/install>.

No entanto, nós estamos rodando esse serviço em um servidor externo exatamente com os arquivos descritos anteriormente e submetidos no Moodle, portanto poderia apenas rodar o App disponibilizado que ele contaria com nosso servidor externo

Esse serviço está hospedado em: <http://jvdecamargo.pythonanywhere.com/>.

A plataforma Python Anywhere permite com que aplicações pequenas em Python sejam hospedadas

temporariamente de forma gratuita (nosso serviço ficará por mais 3 meses disponível).

O arquivo generator.py é o "principal". Seu método principal faz com que a web.py inicie

um serviço que espera requisições. A cada nova requisição, uma nova frase é gerada.

Atualize a página e verá que a frase muda.

Dessa forma, nosso cliente Android faz uma requisição pro endereço acima e recebe como resposta uma frase, que é mostrada (e narrada).

para o usuário. O mesmo usuário então pode pedir outra frase, compartilhar a frase atual no WhatsApp, etc. Para rodar a aplicação, é só instalar o .apk entregue em um celular Android.