MAC5725 – Artigo para Avaliação
Treinamento de Gramáticas Livres de Contexto Probabilísticas

Marcelo Finger

Data de entrega: 04/10/2014

Exercício 1

Fazer um analisador sintático que recebe como entrada:

- Uma gramática livre de contexto (GLC); e
- uma sentençaa do português

e retorna a(s) árvore(s) sintática(s) caso a sentença seja reconhecida. O seu programa deve implementar o algoritmo de Earley. Seu programa poderá ser implementado em uma das seguintes linguagens de programação: C, C++, Java, Perl, Python ou Prolog[1].

A gramática deve ser extraída de 80% do córpus de treinamento e verificada em 20% do córpus. Rodar 5 experimentos em que as partes de treino e teste são extraídas aleatoriamente e de forma distinta para cada experimento. O córpus é dado em:

- Córpus filtrado para ser usado em treinameto: `aires-treino.parsed`. É esse arquivo que deve ser usado tanto para teste quanto para treino.
- Córpus sem filtro: `a_aires_penn.mjc`, apenas para se ter uma idéia de como é um texto de verdade.

Do córpus de treinamento, deverão ser extraídas as regras gramaticais. O córpus de teste deverá ser processado para extrair apenas a sentença.

Em cada caso, medir a proporção de sentenças do córpus de treinamento que foram reconhecidas pela gramática (esta medida é chamada de *cobertura*). Dentre as sentenças reconhecidas, medir a porcentagem de sentenças que admite a árvore do córpus de teste (esta medida é chamada de *precisão*).

Exercício 2

[1] Para outras linguagens, falar diretamente com o professor.

Alterar o seu programa para que a entrada seja agora uma GLCP (gramática livre de contexto probabilistica). Caso a sentença seja reconhecida, seu programa deverá computar, para cada nó da árvore sintática, a probabilidade interna, a fim de gerar ao final da análise apenas a árvore de probabilidade máxima.

Vamos treinar uma gramática livre de contexto probalística através do mesmo córpus de treinamento do exercício anterior.

É importante salientar que o programa do exercício anterior será *intensamente utilizados* neste processo.

O córpus de treinamento será dividido, aleatoriamente, em duas partes: 90% para treino e 10% para teste. Você deverá rodar 10 experimentos de treinamento/teste e para cada um deles, computar a cobertura, precisão e medida-f. No final, tirar a média aritméticas da cobertura e da precisão, e computar a medida-f correspondente. Isto se chama *validação cruzada* de multiplicidade 10.

O processo de treinamento deverá usar o córpus de treino para aprender as regras e as probabilidades associadas a ela. Algum processo de suavização poderá ser aplicado.

Uma coisa importante a ser notada é que o córpus nao possui sentenças marcadas com S. É parte integrante do treinamento determinar quais os não-terminais que podem dominar uma sentença. A menos de falha na filtragem (por favor, avisem), toda sentença termina com ponto final, vírgula ou ponto-vírgula; e todo ponto final é fim de senteça.

Por fim, com a gramática treinada, aplicar esta gramática ao córpus de teste e calcular os valores de precisão, cobertura e medida-f. No caso da precisão, há mais de uma medida possível:

1. A *precisão da parentetização* mede a porcentagem de constituintes corretos, ignorando-se o sintagma atribuído ao constituinte.

2. A *precisão total* mede a porcentagem de sintagmas corretos. Claramente a precisão total é menor ou igual à precisão de parentetização.

Exercício 3

Fazer um artigo de 10 a 20 páginas que descreve sua implementação. O artigo deve abordar os seguintes tópicos:

1. A finalidade do programa e suas hipóteses básicas de construção.

2. O formato da entrada e saída na fase de treinamento e na fase de execução.

3. Uma descrição breve dos algoritmos.

4. Uma descrição das principais estruturas de dados utilizadas, da arquitetura do seu sistema e das estratégias de imlementação.

5. Os valores de precisão, cobertura e medida-f do córpus de teste para cada um dos experimentos, bem como o tempo de treinamento e o tempo de execução em cada um dos experimentos.

6. Uma discussão da qualidade do seu programa, e de formas de melhorar sua performance e eficiência. Aponte também as principais dificuldades encontradas no desenvolvimento do algoritmo.

Note que a clareza e a correção da exposição das idéias são elementos fundamentais da avaliação. Erros gramaticais e o estilo também serão levados em consideração. Não enrole!!

Devem ser entregues os seguintes itens, em um único arquivo zipado (max 10MB):

- O artigo a ser avaliado.
- Uma breve explicação de como usar o programa na fase de treinamento e de execução.
- Os fontes do programa.