LÍNGUA NATURAL 2017/2018 —— Mini-Projecto Nº 2 (MP2)

A realizar:	☐	individualmente	☒ **em grupo**
Local de entrega:	☐	aula teórica	☒ **submissão electrónica**
Data limite entrega:	**até às 15:30 do dia 6/Nov**		

OBJECTIVOS

Construção de modelos de língua estatísticos. Sua utilização prática.

ENUNCIADO

Pretende-se realizar a desambiguação de formas superficiais verbais associadas a dois verbos (eg., "foi" é uma forma verbal do verbo "ir" e do verbo "ser").

1. Para cada uma das palavras/corpus a desambiguar:

    a. Anote o lema em todas as ocorrências das palavras a desambiguar usando a aplicação java: *CorpusAnnotator.class*.
        - Por exemplo, para a palavra "*foi*", o corpus a anotar é o ficheiro "foiIrSer.txt", e o ficheiro de parametrização a usar na aplicação é "foiParametrizacao.txt";
        - Para iniciar o processo de anotação deve executar o comando: "java CorpusAnnotator". Abre-se uma janela onde deve selecionar os ficheiros com o corpus e a parametrização;
        - O ficheiro anotado deve ter o mesmo nome do ficheiro a anotar, mas com a extensão ".out".

    b. Converta o o ficheiro com a extensão ".out" para outro com a extensão ".final", em que:
        - são eliminadas as linhas em que a palavra a anotar não é um verbo;
        - são eliminadas as linhas em que a palavra a anotar ocorre duas (ou mais) vezes na mesma frase;
        - são eliminadas as linhas em que a frase tem erros (que dificultem a geração dos modelos) ;
        - são eliminadas as linhas em que o anotador teve dúvidas;
        - a palavra a desambiguar é substituída pelo lema que aparece no início de cada linha;
        - é removido o lema que aparece no início de cada linha;

    c. Calcule os unigramas e bigramas, sem e com alisamento presentes no ficheiro ".final". (qualquer estratégia de alisamento é aceite);

        Nota: pode usar qualquer ferramenta para calcular os ficheiros de unigramas e bigramas;

        Nota: qualquer estratégia de alisamento é aceite;

        Nota: para facilitar a tarefa de avaliação por parte do docente, os ficheiros calculados devem apresentar uma de duas sintaxes:
        - contagem por linha (ver os ficheiros "unigramasDEMO.txt" e "bigramasDEMO.txt" que contêm o formato desejado);
        - ARPA format (ver secção 4.8 do [Jurafsky & Martin, 2009], ver o ficheiro "gramasDEMO.arpa" que contém o formato desejado).

2. Escreva um programa que deve indicar o lema mais provável para cada frase a processar, de acordo com os modelos de língua carregados. O programa tem como entradas:
    - um ficheiro com unigramas (por exemplo, unigramasDEMO.txt ou gramasDEMO.arpa);
    - um ficheiro com bigramas (por exemplo, bigramasDEMO.txt ou gramasDEMO.arpa);
    - um ficheiro contendo a forma superficial ambígua e os respectivos lemas (por exemplo, foiParametrizacao.txt);
    - um ficheiro com frases de teste a processar (por exemplo, frasesDEMO.txt).

Nota: o programa deve listar o valor calculado para cada uma das opções avaliadas;

3. Teste o programa desenvolvido com 5 novas frases para cada uma das palavras que lhe estão atribuídas. Comente os resultados obtidos.

4. Comente a viabilidade de desenvolver sistemas que seleccionem o lema correcto.

Distribuição do trabalho pelos grupos

Cada grupo receberá por email os ficheiros a anotar.

Submissão

Submeta no Fenix, agrupamento Mini-Projecto, um ficheiro zip (O nome do ficheiro deve ter a forma numGrupo-X, em que X é “MP1-TAGUS” ou “MP1-ALAMEDA”) que deve conter:

- um ficheiro de texto (com o nome “opcoes.txt”) com a descrição das opções tomadas, não podendo exceder 1 página A4;
- os 4 ficheiros anotados usados para calcular os bigramas (“palavra1Anotado.out”, “palavra2Anotado.out”, “palavra1Anotado.final”, “palavra2Anotado.final”) [tarefa 1.a e 1.b];
- os ficheiros com os unigramas e bigramas alisados (“palavra1Unigramas.txt”, “palavra2Unigramas.txt”, “palavra1Bigramas.txt”, “palavra2Bigramas.txt” ou “palavra1.arpa”. “palavra2.arpa”) [tarefa 1.c];
- os ficheiros com as frases usadas para teste (“palavra1Frases.txt” e “palavra2Frases.txt”) [tarefa 3];
- o ficheiro com os resultados obtidos (“palavra1Resultado.txt” e “palavra2Resultado.txt”) [tarefa 3];
- todo o código necessário à obtenção dos resultados apresentados [tarefa 2];
- o ficheiro de texto (“viabilidade.txt”) com a análise à viabilidade, não podendo exceder 1 página A4 [tarefa 4];
- um ficheiro de texto ("run.sh", ou "run.bat") com os comandos usados para obter todos os resultados reportados;

Sempre que possível, todos os ficheiros devem conter a identificação do grupo e dos alunos participantes na elaboração deste trabalho.

Critérios de avaliação

Na avaliação serão tidos em conta os seguintes critérios:

1. Correcção na construção dos corpora anotados (1,0 valor);
2. Correcção no cálculo dos n-gramas sem e com alisamento (0,5 valores);
3. Frases de teste (0,1 valores);
4. Programa apresenta os valores calculados e usados para escolher o lema mais provável (0,4 valores);
5. Existência do script run.sh (0,2 valores);
6. Correção dos valores calculados para os exemplos apresentados (1,4 valores);
7. Descrição das opções tomadas e viabilidade (0,2 valores);
8. Cumprimento dos prazos e correção ortográfica e sintáctica (0,2 valores);
9. Cumprimento de todas as regras de submissão. O não cumprimento de qualquer regra implica um desconto mínimo de 1 valor. Se os programas não respeitar o formato de entrada indicado, ocorrerá uma penalização extra de 2 valores (em 4);

Código de honra na Universidade de Stanford (http://www.stanford.edu/dept/vpsa/judicialaffairs/guiding/honorcode.htm)

The Honor Code is the University's statement on academic integrity written by students in 1921. It articulates University expectations of students and faculty in establishing and maintaining the highest standards in academic work:

1. The Honor Code is an undertaking of the students, individually and collectively:
    1. that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;
    2. that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.
2. The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.
3. While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

Examples of conduct that have been regarded as being in violation of the Honor Code include:

- Copying from another's examination paper or allowing another to copy from one's own paper
- Unpermitted collaboration
- Plagiarism
- Revising and resubmitting a quiz or exam for regrading, without the instructor's knowledge and consent
- Giving or receiving unpermitted aid on a take-home examination
- Representing as one's own work the work of another
- Giving or receiving aid on an academic assignment under circumstances in which a reasonable person should have known that such aid was not permitted

In recent years, most student disciplinary cases have involved Honor Code violations; of these, the most frequent arise when a student submits another's work as his or her own, or gives or receives unpermitted aid. The standard penalty for a first offense includes a one-quarter suspension from the University and 40 hours of community service. In addition, most faculty members issue a "No Pass" or "No Credit" for the course in which the violation occurred. The standard penalty for multiple violations (e.g. cheating more than once in the same course) is a three-quarter suspension and 40 or more hours of community service.