Seleção Vlibras 2020

Desenvolvedor deep learning

Este desafio possui caráter classificatório. No entanto, espera-se um desempenho razoável do candidato. Além da resolução dos problemas, também serão avaliados critérios secundários como a organização e a eficiência do código.

O desafio é composto por três problemas que deverão ser resolvidos apenas com **Python 3**. Cada um deles utiliza um arquivo de entrada diferente e todos eles podem ser encontrados dentro do diretório "**/home/selecao/corpora**" na seguinte máquina:

- **IP:** 150.165.204.181
- **Usuário:** selecao
- **Senha:** lavidselecao

Utilize **ssh** para acessar essa máquina e copie os arquivos para sua máquina usando **scp**.

Os três corpus (base de dados textual) estão no formato **csv**. A primeira linha de cada arquivo contém os nomes das colunas ("gr" e "gi") e **deve ser desconsiderada nas questões 2 e 3**. Os campos de cada linha estão separados por "," (vírgula).
Utilize apenas os módulos **csv** ou **pandas** para ler esses arquivos

O código-resposta utilizado **deverá** ser disponibilizado em um repositório no GitHub, cujo link deverá ser enviado, **junto com os corpus-resposta(!)**, para o endereço de e-mail "**selecao-vlibras@lavid.ufpb.br**". Só serão consideradas submissões que contenham tanto o código-resposta quanto o corpus-resposta. Os candidatos terão até às **23:59 do dia 14/02/2020** para enviar suas soluções (será avaliada apenas a **última versão com código e corpus gerados** enviada).

Não serão considerados commits/e-mails posteriores a esse horário!

Q1 - No corpus "corpus-q1.csv" temos duas notações de glosa, uma representação textual da linguagem brasileira de sinais (LIBRAS). Nessas glosas, alterações dos sinais de LIBRAS são representadas por "**qualificadores**" (veja o **ANEXO I** caso queira saber mais sobre qualificadores; não é necessário para a resolução dos problemas).
Mantendo a primeira linha ("gr,gi"), resolva os seguintes problemas (preferencialmente, na mesma ordem) utilizando o módulo de regex do python, "**re**":

- Há várias ocorrências da seguinte notação: S ou P seguidos de 1, 2 ou 3. Por exemplo: 1S_DAR_3P. Faça um regex que corrija casos invertidos como P2_PERGUNTAR_1S para 2P_PERGUNTAR_1S.

- Simplifique múltiplos sinais '+' entre parênteses para apenas um "+" (i.e. "(+++)" deve virar "(+)").

- Simplifique múltiplos espaços em apenas um. Por exemplo:
  "ABELHA AMARELA" deve virar "ABELHA AMARELA".

- Remova um ou mais caracteres '-' (hífen) entre dígitos. Por exemplo: "222-3333" deve virar "2223333"

- Remova espaços incorretamente inseridos antes de qualificadores de local (_CIDADE, _ESTADO, _PAÍS). Por exemplo: "IR RECIFE _ESTADO" deve virar "IR RECIFE_ESTADO"

- Remova espaços incorretamente inseridos antes de qualificadores direcionais na direita. Por exemplo: "1S_DAR _3P" deve virar "1S_DAR_3P"

- Adicione espaços após os qualificadores direcionais pela esquerda. Por exemplo: "1S_DAR_3P" deve virar "1S_ DAR_3P"

- Remova os espaços à esquerda dos qualificadores de intensidade ("+" ou "-" entre parênteses). Por exemplo: "AMAR (+)" deve virar "AMAR(+)".

- Em toda palavra que vier precedida de "NÃO", substitua o espaço por um _. Por exemplo: "NÃO GOSTAR" deve virar "NÃO_GOSTAR"

- Substitua o “_” (sublinhado ou underline) por um “&” (e comercial ou ampersand) nos qualificadores de pessoa famosa (“_FAMOSO”, “_FAMOSA”). Por exemplo: “ALBERT_EINSTEIN_FAMOSO” deve virar “ALBERT_EINSTEIN&FAMOSO”.

- Remova todas os pontos que não pertençam a um número decimal. Por exemplo: "IR. ONTEM. BANCO" deve virar "IR ONTEM BANCO".

- Acrescente o 0 implícito em números decimais. Por exemplo: "PAGAR .72" deve virar "PAGAR 0.72".

Q2 - Escreva um script que gere uma lista de frequências para as palavras presentes no arquivo "corpus-q2.csv", gerando um JSON onde os campos são as palavras e o valor de cada campo é o número de ocorrências daquela palavra.

Por exemplo, se o arquivo contiver apenas a frase: "ENTREVISTA JÔ SOARES, ENTREVISTA JÔ_SOARES_FAMOSO", o arquivo JSON retornado deverá ser:

```
{
    "entrevista": 2,
    "jô": 1,
    "soares": 1,
    "jô_soares_famoso": 1
}
```

Q3 - Escreva um script para fazer um *data augmentation* (geração de novos dados a partir de dados pré-existentes) com as frases presentes no arquivo "corpus-q3.csv". O *augmentation* deverá funcionar da seguinte forma:

Caso a frase tenha um ou mais verbos com qualificadores direcionais (1S, 2S, 3S, 1P, 2P, 3P), deverão ser geradas novas frases com todas as flexões possíveis. Caso a Por exemplo, dada a frase "ONTEM 1S_DAR_3S LIVRO", deverão ser geradas frases como:

ONTEM 1S_DAR_1S LIVRO
ONTEM 1S_DAR_2S LIVRO
ONTEM 1S_DAR_3S LIVRO
...
ONTEM 3P_DAR_1P LIVRO
ONTEM 3P_DAR_2P LIVRO
ONTEM 3P_DAR_3P LIVRO

Todas as frases possíveis devem então ser salvas em um arquivo de texto, com uma frase por linha. A ordem das frases não importa.

Dúvidas: **selecao-vlibras@lavid.ufpb.br**

ANEXO I

Os qualificadores podem ser de:

- Intensidade: indicam a intensidade do sinal; são representados pelo nome do sinal seguido, sem espaço, por parênteses com um sinal "+" ou "-" dentro deles. Por exemplo, "DORMIR(+)".
- Local: indicam sinais específicos para nomes de locais; são representados pelo nome do sinal seguido, sem espaço, por "_" (sublinhado ou underline) e um desses valores "CIDADE", "ESTADO" ou "PAÍS". Por exemplo, "SÃO_PAULO_CIDADE" e "SÃO_PAULO_ESTADO".
- Famosos: indicam sinais específicos para nomes de famosos; são representados pelo nome do famoso seguido, sem espaço, por "_" (sublinhado ou underline) e "FAMOSO" e "FAMOSA". Por exemplo, o sinal para o cantor Djavan seria representado por "DJAVAN_FAMOSO".
- Direcionalidade: indicam o agente e o receptor de sinais específicos; são representados pela flexão de pessoa e flexão de número do **agente** ("1S" - primeira pessoa do singular, "2P" segunda pessoa do singular) seguido de **"_", verbo (no infinitivo), "_"** e flexão de pessoa e flexão de número do **receptor**. Por exemplo, "NÓS GOSTAMOS DELA" (português) é representado por "1P_GOSTAR_3S" (glosa).