EXERCÍCIOS DE PROCESSAMENTO DE LINGUAGEM NATURAL

AUTOR: FABRÍCIO GALENDE MARQUES DE CARVALHO

UNIDADE 3.

MODELOS PARA PROCESSAMENTO DE LINGUAGEM NATURAL

TERMINOLOGIA E CONCEITOS

TC.3.1. Qual tipo de problema pode surgir na montagem de um modelo do tipo *bag of words* caso etapas tais como a remoção de caracteres especiais (ex. sinais de pontuação) e conversão para minúsculas/maiúsculas não sejam efetuadas? Ilustre isso para as seguintes frases que fazem parte de um mesmo corpus de texto (i.e.: são usadas para a montagem do léxico do modelo):

Frase 1: Eu quero tomar água!

Frase 2: eu, prefiro tomar café.

Na sua resposta mostre como ficaria o léxico do modelo e os *bag of words* correspondentes.

TC.3.2. Qual a relação entre as etapas de pré-processamento de texto e a redução de dimensionalidade quando se lida com extração de características? Ilustre isso considerando 2 exemplos que façam uso de stemização e/ou lematização.

TC.3.3. Descreva como ficaria o léxico do modelo e o vetor de características n-gram para n=1, 2 e 3 para os seguintes documentos pertencentes ao mesmo corpus:

Frase 1: Eu não gostei do produto e o produto parece ruim.

Frase 2: O produto parece bom.

Frase 3: O produto parece ruim.

TC.3.4. Para o exercício TC.3.3, considerando um modelo *bag of words*, com n=1, mostre como se calcula o valor da transformação TFIDF para as palavras produto e ruim, ambas na frase 1. Utilize as expressões fornecidas no material das aulas de PLN.

TC.3.5. Considere as seguintes frases utilizadas em um sistema de processamento de linguagem natural:

Frase 1: Gostaria de pedir ajuda para alguém.

Frase 2: Gostaria de ajuda.

Frase 3: Gostaria de sair do sistema.

Frase 4: Sair do sistema!

Frase 5: Reiniciar o sistema!

Considerando um modelo do tipo bag of n-words, com n= 1 e n =2, monte o léxico do modelo e o vetor de caraterísticas representativo de cada frase considerando:

a) A mera contabilização dos n-gramas na frase.
b) A aplicação da transformação TFIDF para o vetor de contabilização.

Diga se n=1 ou n=2 afeta o tamanho e o valor das componentes do vetor de características obtido para o modelo. Na sua resposta, especifique os valores dos termos IDF para os

unigramas e bigramas. Apresente o detalhamento do cálculo de pelo menos 4 IDFs, sendo 2 para unigramas e 2 para bigramas.

PRÁTICA DE PROGRAMAÇÃO

PP.3.1. Baseando-se nos exemplos fornecidos pelo professor, que fazem uso da biblioteca scikit-learn, ilustre a obtenção de um vetor do tipo *bag of words* com transformação do tipo TFIDF para **três documentos que representem reviews de produtos em um site de e-commerce**. Execute todas as etapas de pré-processamento necessárias para normalizar os dados. Uma vez obtidos os vetores, faça o cálculo de similaridade utilizando a função cosseno e diga quais reviews são mais similares e quais reviews são mais distintas.

PP.3.2. Considerando um corpus de texto contendo revisões de produtos, selecione algumas revisões que possam ser caracterizadas como positivas, negativas ou neutras. Treine cada um dos modelos seguintes, tendo como base o código-fonte fornecido pelo professor, para que sejam capazes de classificar uma determinada revisão informada pelo usuário, diferente daquela que foi utilizada no treinamento do modelo. Salve os dados do seu modelo treinado em um arquivo pickle, recarregue e demonstre a sua utilização para nova classificação.

a) Multilayer perceptron;
b) K-Nearest Neighbors

PP.3.3. Demonstre a técnica de agrupamento hierárquico de documentos similares utilizando alguns dados de reviews de produtos. Ilustre e explique o dendrograma em especial no que se refere aos pontos de corte para as distâncias. Faça uso de dados de reviews de produtos e não se esqueça de normalizar os dados antes de efetuar a montagem dos vetores de características.

PP.3.4. Demonstre a modelagem de tópicos, com LDA, utilizando alguns documentos representativos de revisões de produtos. Efetue todas as etapas de pré-processamento adequadas antes de efetuar a modelagem.

UNIDADE 4.

MODELOS AVANÇADOS PARA PROCESSAMENTO DE LINGUAGEM NATURAL

TERMINOLOGIA E CONCEITOS

TC.4.1. O modelo avançado chamado Word2Vec armazena informação contextual através de qual parâmetro?

PRÁTICA DE PROGRAMAÇÃO

PP.4.2. Construa um modelo do tipo word2vec (W2V) para classificação de revisões de produto que atenda aos seguintes critérios:

a) O modelo deve operar sobre dados de revisão que tenham sido pré-processados;
b) O modelo deve ser comparado, em termos de desempenho de classificação, com um modelo clássico do tipo bag of words (BOW) com transformação TFIDF.
c) Para a classificação, utilizar no mínimo 15 reviews de treinamento, classificador utilizando Multilayer Perceptron e mais 45 reviews de validação, sendo elas igualmente distribuídas entre revisões positivas, negativas e neutras.

Monte a matriz de confusão comparativa para os dois modelos. Essa tabela deve mostrar os dados tais como ilustrados abaixo.

		Condição prevista		
		Positivos	**Negativos**	**Neutros**
Condição obtida pelo modelo	**Positivo**	**Verdadeiro Positivo (TP)**	**Falso Negativo Positivo (FNP)**	**Falso Neutro Positivo (FNeP)**
	Negativo	**Falso Positivo Negativo (FPN)**	**Verdadeiro Negativo (TN)**	**Falso Neutro Negativo (FNeN)**
	Neutro	**Falso Positivo Neutro (FPNe)**	**Falso Negativo Neutro (FNNe)**	**Verdadeiro Neutro (TNe)**

Notar que, para o total de amostras T, tem-se.

T = P + N + Ne, sendo P = Positivas, N= Negativas, Ne = Neutras

P = TP + FPN + FPNe

N = TN + FNP + FNNe

Ne = TNe + FNeNe + FNeP

Calcule (demonstre executando a classificação com o seu modelo):

TPR (True Positive Rate) = $\frac{TP}{P}$, (TP representa True positive – Verdadeiros Positivos)

FPN (False Positive Negative Rate) = $\frac{FPN}{P}$

FPNe (False Positive Neutral Rate) = $\frac{FPNe}{P}$

FPR (False Positive Rate) = $\frac{FNP + FNeP}{N + Ne}$

TNR (True Negative Rate) = $\frac{TN}{N}$ (TN representa True Negative – Verdadeiro Negativo)

FNP (False Negative Positive Rate)= $\frac{FNP}{N}$

FNNe (False Negative Neutral Rate): $\frac{FNNe}{N}$

FNR (False Negative Rate) = $\frac{FPN + FNeN}{P + Ne}$

TNeR (True Neutral Rate) = $\frac{TNe}{Ne}$ (TNe representa True Neutral – Verdadeiro Neutro)

FNeP (False Neutral Positive Rate) = $\frac{FNeP}{Ne}$

FNeN (False Neutral Negative Fate) = $\frac{FNeN}{Ne}$

FNeR (False Neutral Rate) = $\frac{FPNe + FNNe}{P + N}$

Acurácia do modelo = $\frac{TP + TN + TNe}{T}$

Razão de verossimilhança para amostras positivas: $\frac{TPR}{FPR}$

Razão de verossimilhança para amostras negativas: $\frac{TNR}{FNR}$

Razão de verossimilhança para amostras neutras: $\frac{TNeR}{FNeR}$

Analisando os dados obtidos pelo seu modelo e os valores para as várias taxas, acurácia geral e razões de verossimilhança, diga o que poderá acontecer caso esse modelo seja utilizado em uma base de dados contendo novas reviews (qual a previsão de comportamento da classificação)?

O que acontece se o número de amostras de uma das classes for reduzido enquanto o número de amostra das demais for mantido? (ex. Utilize 2 amostras negativas, 8 positivas e 8 neutras para treinamento e repita o cálculo das taxas de acerto e erro para positivas, negativas e neutras, assim como os demais indicadores de desempenho – acurácia e razões de verossimilhança).

Para dar uma melhor ideia dos dados calculados, monte uma matriz de confusão utilizando cores de fundo que se aproximam de verde caso o modelo acerte a classificação (ex. elevados valores para a diagonal) e se aproxime de vermelho, caso erre a classificação (ex. elevados valores para os elementos não diagonais).

PP.4.3. Crie uma aplicação baseada em word2vec que, dada uma frase informada pelo usuário, reescreva essa frase substituindo algumas de suas palavras por sinônimos. Esses sinônimos deverão ser escolhidos utilizando um critério de distância ou ângulo mínimo.

Na sua resposta, mostre a base de dados utilizada para treinamento e substituição das palavras e diga se isso afeta ou não o desempenho do sistema.

PP.4.4. Ilustre uma aplicação simples de chatbot que utilize o modelo pré-treinado word2vec. Seu modelo deve fazer uso dos componetes fornecidos no repositório do professor da disciplina e deve possuir todas as componentes básicas da pipeline de pré-processamento. O Chatbot deve ser utilizado para responder perguntas relacionadas ao uso de determinada aplicação (ex. Como gero um relatório? Como faço o upload de dados, etc.).

PRÁTICA PARA COMPLEMENTAÇÃO DE APRENDIZADO:

Nesses exercícios, a apresentação deverá conter, no mínimo:

1. Os fundamentos e conceitos da técnica.
2. Suas principais aplicações
3. Um exemplo de código-fonte, em python, que seja funcional e que possa ser executado pelos colegas de classe e pelo professor durante sua apresentação. Seu exemplo deve ser executado sobre corpus da língua portuguesa.

PP.C.1. Prepare uma pequena apresentação que explique o conceito de sumarização extrativa. Ilustre o algoritmo baseado em similaridade denominado TextRank. Faça um exemplo simples, em Python, que efetue a sumarização de um determinado corpus de texto em língua portuguesa.

PP.C.2. Prepare uma pequena apresentação que explique o conceito de sumarização abstrativa. Ilustre um algoritmo, em python e fazendo uso de quaisquer bibliotecas de sua escolha, que opere sobre um texto da língua portuguesa.

PP.C.3. Prepare uma pequena apresentação discutindo sobre a utilização de transformers. Dê um exemplo simples relacionado à análise de sentimentos. Para esse caso, deixe o modelo pré-treinado apto a receber frases digitadas pelo usuário (via prompt de comando) e que retorne a classificação (ex. Frase positiva, negativa ou neutra). Na sua apresentação/exemplo, discuta se seu sistema considera ou não algum contexto de informação.

PP.C.4. Prepare uma pequena apresentação discutindo sobre análise de componentes principais. Ilustre a aplicação dessa técnica a um sistema de análise de sentimentos utilizando

um modelo BOW. No seu exemplo forneça um conjunto de dados contendo uma quantidade não muito grande de palavras no léxico (em língua portuguesa), que permita a visualização mais clara do efeito de redução de dimensionalidade.