Nome: Fabio Grassiotto
RA: 890441
Disciplina: IA369Y, 2º S 2017

T2 – Análise de Sentimentos em Textos

Objetivo

O objetivo desta atividade foi a determinação de valência de 500 manchetes de diversos jornais brasileiros no 1º semestre de 2017 (problema 1).

Abordagem do problema

Para a análise dos textos foi utilizada a linguagem de programação Python juntamente com a biblioteca de processamento de linguagem natural NLTK, devido à disponibilidade de classificadores de texto com boa performance para a determinação de valência. Para análise dos resultados e apresentação foi utilizada a biblioteca matplotlib.

Todo o código e arquivos necessários para execução da classificação podem ser encontrados e clonados do github em https://github.com/fabiograssiotto/IA369Y.

Etapas para solução do problema

1. Preparação dos dados de entrada

Como primeiro passo, foi verificado que os dados de entrada foram providos em arquivo em formato csv. Para parsear o documento, utilizei a biblioteca padrão do Python (CSV) para obter listas de strings de texto contendo as datas em que as manchetes foram publicadas, a fonte de publicação e as manchetes.

Verifiquei que ao parsear o texto, a codificação padrão utilizada para os caracteres em Python não seria capaz de compreender os caracteres da acentuação brasileira. Para solucionar este problema, utilizei codificação de unicode utf-8 para parsear o texto corretamente, obtendo assim as strings com os acentos.

2. Remoção de stopwords e stemização

Após obter as strings das manchetes, as notícias foram quebradas em palavras usando o recurso de tokenização da NTLK. Após obter a lista de palavras de cada notícia, converti todas as letras em minúsculas e utilizei a lista de stopwords (ou palavras mais comumente utilizadas) da biblioteca NLTK para remover as palavras que não iriam contribuir para a análise de sentimento do texto.

Da lista de palavras, utilizei o recurso de stemização da NLTK (stemmer RSLP) para remover as inflexões das palavras em português para chegar à lista final de palavras que será utilizada para a análise de sentimento.

Por exemplo, a partir da notícia

```
'BC cria novo instrumento de política monetária.'
```

Obtive a lista de palavras

```
['bc', 'cri', 'nov', 'instrument', 'polít', 'monetár']
```

3. Definição do algoritmo a ser utilizado

De acordo com a literatura (Bo Pang et al., 2002), (Domingos et al., 1998), apesar da simplicidade, o algoritmo de classificação de texto Naïve Bayes tende a ter boa performance, sendo considerado ótimo para classes de problemas com features altamente dependentes. Historicamente, esse tipo de algoritmo começou a ser usado no final dos anos 90 para a classificação de spam em email.

O algoritmo pode ser considerado ainda uma baseline inicial para a solução deste tipo de problema. Por esses motivos, resolvi utilizar o classificador que implementa este algoritmo na biblioteca NLTK.

Esse é um algoritmo de classificação supervisionada, descrito como Naïve Bayes Multinomial. Como entrada para a fase de treinamento, recebe um conjunto de palavras rotuladas *a priori* como positivas, negativas ou neutras a partir de um corpus léxico. O algoritmo utiliza o princípio de *"bag of words",* ou seja, considera que a probabilidade que cada palavra possa ocorrer em um documento é independente do contexto e posição da palavra. Através das palavras de treinamento, são estimados parâmetros para uma distribuição multinomial estatística.

Após a fase de treinamento, o algoritmo utiliza o princípio de independência de features para calcular a probabilidade que o documento pertença a uma das classes determinadas pelos rótulos de treinamento e assim classifica um trecho de texto em um dos rótulos utilizados.

Para gerar a intensidade do sentimento positivo ou negativo de uma sentença, assumi que as probabilidades de cada um dos rótulos mediria o quão positiva ou negativa uma sequência de palavras seria. Para o caso de um trecho classificado como neutro, a probabilidade não faz sentido para determinar o quão neutro um texto poderia ser classificado. Portanto, nesse caso assumi que todos os textos neutros tem a mesma intensidade de sentimento, 50%.

4. Definição do Corpus Léxico para classificação

Não existem muitos corpus léxicos em português com análise de sentimento. Encontrei dois, o SentiLex-PT e o OpLexicon.

O SentiLex (Mário J. Silva et al., 2010) é um léxico do português de portugal, constituído por 6.321 lemas adjectivais (por convenção, na forma masculina singular) e 25.406 formas flexionadas.

O OpLexicon, ou Opinion Lexicon, (Souza M. et al., 2012) foi composto de 346 análises de filmes extraídos dos sites CinePlayers3 e Cinema com Rapadura4 e 970 textos de jornais sobre temas diversos extraídos do corpus PLN-Br CATEG, resultando em um corpus com 1317 documentos e cerca de um milhão de palavras.

Dentre os dois, selecionei o OpLexicon por três motivos:

- Dentre os dois corpora disponíveis, é o único na língua portuguesa do Brasil;
- Não é composto unicamente de adjetivos como o SentiLex;
- Entre outras fontes, foram utilizados textos de jornais para sua composição. Portanto está em um domínio similar às manchetes de jornais que seriam classificadas pelo algoritmo.

5. Parseamento do Corpus Léxico

O corpus OpLexicon é distribuído como um arquivo texto em formato csv com uma lista de palavras e classificações sintáticas e de sentimento. Para extração de dados novamente foi utilizada a biblioteca CSV para extrair os dados relevantes (palavra e sentimento).

Analisei a distribuição de polaridade do OpLexicon e pude constatar, conforme na Figura 1, que as palavras tendem a ser majoritariamente de sentimento negativo, provavelmente por causa das fontes de dados utilizados.

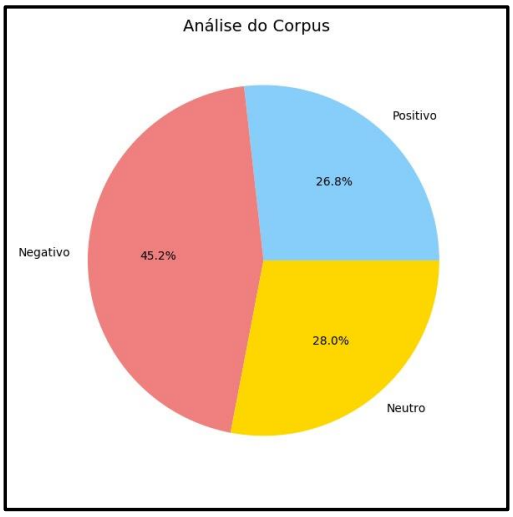

Figura 1 - Análise do OpLexicon

6. Preparação dos dados para treinamento do classificador

Utilizando o stemmer RSLP, foram removidas as terminações das palavras e os dados foram formatados em forma de dicionário para criar o conjunto de dados de treinamento (“*training set*”) do classificador. Assim, uma linha no arquivo do corpus tal como

```
abalar-se,vb,0,A
```

Foi convertida em formato de dicionário para

```
{'abal': True}
```

Note que o segundo ítem do dicionário indica para o classificador a presença da palavra no documento.

7. Alimentação das features no classificador

As palavras assim formatadas foram definidas como as features do *training set* e foram alimentadas ao classificador. Em total, cerca de 31000 features foram utilizadas.

Após este processo, percebi que algumas das palavras mais comuns nos textos a serem analisados não estavam presentes no OpLexicon. Adicionei assim as seguintes palavras, definindo-as com valência neutra.

```
trump,n,0,A
velloso,n,0,A
cabral,n,0,A
temer,n,0,A
doria,n,0,A
presidente,n,0,A
janot,n,0,A
brasil,n,0,A
lula,n,0,A
morgan,n,0,A
freeman,n,0,A
r,n,0,A
```

```
lava-jato,n,0,A
governo,n,0,A
govwerno,n,0,A
govertno,n,0,A
```

Notei que as duas últimas palavras, “govwerno” e “govertno”, foram introduzidas de forma proposital para reduzir a performance da classificação. Adicionei as variantes para evitar este problema.

8. Classificação das Manchetes

As manchetes de entrada foram então entradas no classificador Naives Bayes da NLTK, obtendo como saída um rótulo positivo, negativo ou neutro que, juntamente com as probabilidades de cada rótulo retornadas pelo classificador, foram utilizadas para compor um score em percentagem da valência para cada manchete. Esses dados foram agregados e escritos no arquivo de saída resultados.txt no formato abaixo:

```
                     Manchete                        Valência (%)
-------------------------------------------------------------------
BNDES encolhe e volta ao nível de 20 anos atrás              16.26
BC cria novo instrumento de política monetária.              14.69
Câmbio gera bate-boca entre UA e UE.                         24.69
Indenização a transmissoras de energia já chega à tarifa.     4.59
Políticos esperam que relator separe "joio do trigo".        50.00
```

A valência apresentada na tabela representa em uma escala de 0 a 100% o quanto positiva uma manchete foi classificada. Para finalizar, foi utilizada a biblioteca matplotlib para analisar os dados obtidos agregando as classificações das manchetes por mês e por publicação.

Discussão dos Resultados

Precisão do classificador

Para discussão do algoritmo de classificação, uma amostra aleatória de 10 manchetes foi extraída dos resultados no arquivo amostras.txt:

```
Velloso pode comandar a Justiça.                                50.00
Janor pede que Aécio seja ouvido sobre Furnas no STF.           50.00
Rombo no caixa do Rio só cresce.                                13.16
"Falha elétrica total, sem combustível", avisou piloto.          9.47
Temer: 'Nunca caí de pinguela'.                                 15.79
Doria inclui mais pobres e tira jovens do Leve Leite.            4.30
Do vinho de garrafão a prêmios no exterior.                     20.00
Varejo surpreende e interrompe dois anos de retração.            5.92
Estados vão privatizar empresas de gás natural.                 21.20
Tempestade cobre NY de neve e finda primavera precoce.          25.71
```

Das 10 manchetes, concordo com a classificação do algoritmo em 60 a 70% dos casos acima. Resultados similares são encontrados na literatura para o algoritmo de Naïve Bayes Multinomial (Ismail, Heba et al., 2016) e (Mccallum, Andrew & Nigam, Kamal, 2001).

Distribuição temporal da classificação

Foi realizada uma análise temporal da classificação das manchetes conforma na Figura 2 abaixo.

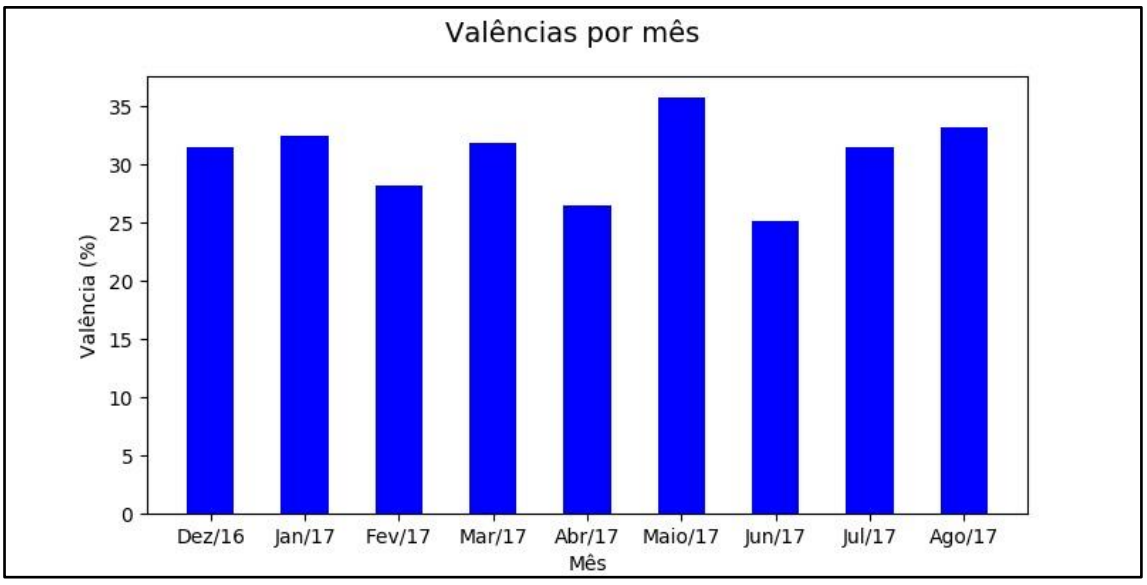

Figura 2 - Valências por mês

Notamos que a média das valências não mudou muito mês a mês, apenas com o mês de maio de 2017 com uma média ligeiramente superior. De uma forma geral, as valências das manchetes tendem a ser negativas, como parece ser característica dos jornais e revistas brasileiros.

Distribuição por publicação

Da mesma forma foi realizada uma análise das valências por fonte de publicação da notícia, conforme a figura 3 abaixo.

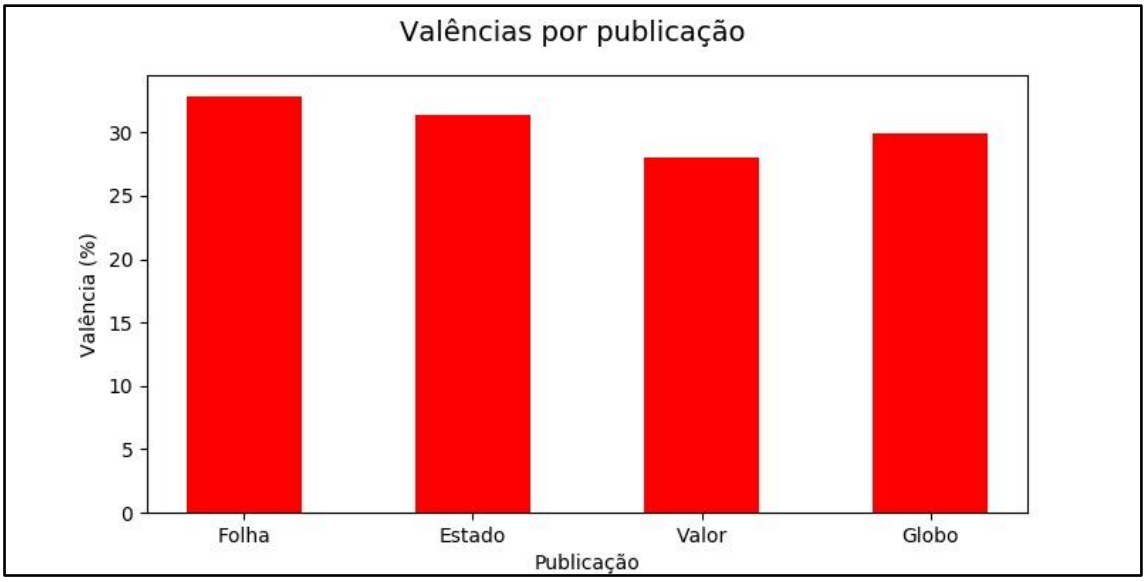

Figura 3 - Valências por publicação

Notamos que, em média, as valências das notícias na revista Valor são mais negativas que as outras publicações. Isso pode ser justificado pelo foco em notícias de política e economia da revista.

Lições Aprendidas

Ao longo do desenvolvimento deste trabalho, pude chegar às conclusões abaixo quanto ao processo prático de classificação emocional de textos:

- A linguagem de programação Python é de aprendizagem rápida. Minha experiência anterior era no uso de linguagens compiladas, como C/C++ e não tive dificuldade em utilizar Pyton para este trabalho.
- Existem várias bibliotecas disponíveis para classificação de textos, assim como interfaces de programação remotas (API), como a Watson da IBM, Google, etc. Em Python, além da NLTK podemos ressaltar a Scikit-Learn e TextBlob (uma interface simplificada da NLTK).
- A documentação da NLTK para classificação é muito esparsa e não é suficiente para a implementação de um sistema prático. Foi necessário acessar vários fóruns na internet como StackOverflow, Quora, entre outros para entender como utilizar a API.
- Os stopwords providos pela NLTK é muito simples para a língua portuguesa. Eu adicionei várias palavras, a maioria nomes próprios, ao corpus para evitar classificações incorretas. Imagino que outra abordagem prática poderia ser a criação de uma lista de stopwords a partir da análise dos textos a serem classificados.
- A stemização das palavras utilizadas na fase de treinamento do classificador é essencial para sua performance. Quando usamos este processo, o classificador recebe mais amotras de um mesmo stem e aumenta a confiabilidade de atribuição de um rótulo.
- O classificador utilizado, Naïve Bayes, tem uma performance muito razoável quando comparado às APIs disponíveis na internet. Acredito que isso se deve ao fato que as bases de dados de treinamento estejam disponíveis de forma majoritária no idioma inglês.
- O classificador utilizado não entrega intensidades de sentimento na saída. Ao tomar a decisão de utilizar as probabilidades de rótulos para determinar a intensidade de valência, verifiquei que a classificação neutra não permite atribuir intensidade e precisei atribuir, de forma arbitrária, uma valência de 50% para as manchetes assim classificadas.
- A biblioteca matplotlib é essencial para a geração de gráficos para análise.

Referências

Domingos, Pedro & Pazzani, Michael. (1998). On the Optimality of the Simple Bayesian Classifier Under Zero-One Loss. Machine Learning. 29.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10* (EMNLP '02), Vol. 10. Association for Computational Linguistics, Stroudsburg, PA, USA, 79-86.

Mário J. Silva, Paula Carvalho, Carlos Costa, Luís Sarmento, Automatic Expansion of a Social Judgment Lexicon for Sentiment Analysis Technical Report. TR 10-08. University of Lisbon, Faculty of Sciences, LASIGE, December 2010.

Souza, M.; Vieira, R.; Busetti, D.; Chishman, R. e Alves, I. M. Construction of a Portuguese Opinion Lexicon from multiple resources. 8th Brazilian Symposium in Information and Human Language Technology, 2012.

Ismail, Heba & Harous, S & Belkhouche, Boumediene. (2016). A Comparative Analysis of Machine Learning Classifiers for Twitter Sentiment Analysis.

Mccallum, Andrew & Nigam, Kamal. (2001). A Comparison of Event Models for Naive Bayes Text Classification. Work Learn Text Categ. 752.