Aprendizagem de Máquina para Detecção de Notícias Falsas

André Correia Lacerda Mafra

***Abstract*— A detecção de notícias falsas é importante para ajudar a sociedade a identificar notícias que possam prejudicar pessoas, interferir negativamente no mercado ou até mesmo influenciar o resultado de uma eleição presidencial. A predição consiste em analisar previamente, uma grande coleção de notícias falsas e verdadeiras, de tal forma que o resultado final é totalmente dependente da qualidade desta coleção. A falta de conjuntos de notícias conhecidas *a priori* como falsas é um obstáculo na aprendizagem do modelo. No experimento foram utilizados os algoritmos de SVM e Naive Bayes, sendo que ambos trouxeram bons resultados.**

I. INTRODUÇÃO

A palavra do ano de 2016, segundo a Oxford Dictionary foi *Post-Truth*, uma expressão que significa 'relacionado ou denotado por circunstâncias em que fatos objetivos exercem menos influência do que apelos emocionais e crenças pessoais'. A escolha da palavra, com certeza, deve-se ao Brexit e à eleição de Donald Trump como Presidente dos Estados Unidos, pois ambos os casos foram marcados por controvérsias na mídia, principalmente em redes sociais e blogs, e muitos acreditam que os resultados do referendo Britânico e das eleições americanas foram influenciados por tais controvérsias.

O impacto das notícias falsas compartilhadas na internet em 2016, acabou colocando grandes empresas de tecnologia, como o Facebook no epicentro das críticas e atenção da mídia e público, portanto dando-lhes a preocupação de encontrar soluções para validação das notícias publicadas em seu meio, de forma a garantir sua autenticidade.

Rubin, Chen & Conroy (2015) definem três tipos de notícias falsas : Invenções, Farsas e Sátira. Respectivamente, representam testemunhos sensacionalistas ("click-bait"), notícias não validadas ou sem fundamento e por fim de conteúdo humorístico, onde o intuito, muita das vezes, não é negativo porém pode ser utilizado contra leitores desavisados de forma descontextualizada.

Por simplificação, nesta pesquisa foi optado por abrangir todos os três tipos de notícias em somente um tipo genérico denominado "falso". A base de dados utilizada para o aprendizado do algoritmo, também é limitada a somente o tipo genério "falso".

A solução proposta para detecção de Notícias Falsas foi a de desenvolver dois modelos de classificação: Support Vector Machines(SVM) e Naive Bayes, além de dois modelos de seleção de *features* : TF e TF-IDF. O intuito é de comparar o resultado de todos os modelos, combinando cada modelo de classificação com um de seleção de *features*, de forma a obter quatro modelos finais : SVM com TF, SVM com TF-IDF, Naive Bayes com TF, Naive Bayes com TF-IDF, e por fim compará-los em termos de resultado e performance. Com os resultados foi possível aferir que o modelo treinado com SVM e TF-IDF são melhores em classificar as notícias, por outro lado, a diferença de resultados dentre todos os modelos foi baixa, em destaque ao TF-IDF que acrescentou pouca variância de acurácia em relação ao TF.

II. CENÁRIO E ABORDAGEM

A. Coleta de dados e sobre o Dataset

O Dataset utilizado é uma mistura de duas bases de dados distintas em inglês, sendo que uma se refere apenas à notícias falsas e outra à notícias verdadeiras. Respectivamente, a primeira foi encontrada no site Kaggle com o nome de *Getting Real About Fake News*, já o segundo foi disponibilizado por um usuário do Github *GeorgeMcIntire*, que também as mesclou com a base de dados do Kaggle, tudo em um só Dataset, sendo este o utilizado nesta pesquisa, contendo 6.335 exemplos, em que 3.164 são conhecidamente falsos e 3.171 são conhecidamente verdadeiros.

B. Requisitos para um bom corpus

Rubin, Chen, and Conroy(2015) consideraram nove fatores ao avaliarem a qualidade de um Corpus, ou documento para detecção de notícias falsas.

- **Possibilidade de aprender com as instâncias**: Os métodos de classificação devem ser capazes de encontrar padrões e regularidades dentre os rótulos falso e verdadeiro do Dataset.
- **Estar em formato de texto digital**: Texto é a melhor forma de se utilizar Processamento de Linguagem Natural.
- **Estar em formato de texto digital**: Garantir que o Corpus é confiável, é sempre recomendado utilizar notícias que venham de fontes tradicionais e que sejam reconhecidamente genuínas.
- **Homogeneidade do tamanho**: Notícias com tamanhos diferentes, como uma reportagem de um jornal e um tweet,não constituem uma base de dados homogênea.
- **Homogeneidade da escrita**: Datasets que possuam documentos com escrita familiar, dividida igualmente em tópicos, subtítulos e etc.
- **Tempo pré-determinado**: O corpus tem que ser coletado dentro um tempo específico, para garantir que a notícia não sofra variância, como por exemplo em notícias relâmpago, plantões televisivos etc.
- **Contexto**: Saber o contexto de uma notícia, é de suma importância para minizar falhas, como por exemplo classificar corretamente sátiras como "humor" e não como farsa.

- **Pragmatismo**: Leva em conta custo de obtenção do dado, copyright, disponibilidade do dataset, dentre outros.
- **Cultura e Linguagem**: É importante considerar diferenças linguísticas e culturais quando se avaliar um texto.

C. Literatura Existente

Existe uma quantidade enorme de pesquisa no que se refere à métodos de Machine Learning para detecção de notícias falsas, a grande maioria com o foco em classificação de *reviews* online e postagens em redes sociais.

- **Trabalhos promissores**: Rubin, Chen & Conroy(2015) realizaram várias abordagens em classificação de notícias, com ótimos resultados. Lauren Dyson & Alden Golab propuseram um estudo de detecção de notícias falsas, de forma a abordar inúmeros modelos possíveis de seleção de *features* e de classificação, ao fim entregando uma excelente visão da eficiência de cada modelo em detectar farsas.

D. Linha de Pesquisa

Foi decidido nesta pesquisa focar nos métodos de classificação, e em analisar as situações em que cada modelo se comporta melhor ou pior do que o outro, em termos de Acurácia, Precisão e *Recall*. Dessa forma será possível comparar os resultados obtidos com os de outros autores, como Dyson & Golab.

O objetivo principal deste projeto é identificar, dado um conjunto de dados, se determinada notícia é falsa ou verdadeira.

III. Escolhas do Modelo

A. Representação das Features

No modelo desta pesquisa, cada palavra é uma feature. Os algoritmos de classificação necessitam processar a notícia e ter acesso direto à cada feature, portanto, cada palavra w deve ser separada individualmente e armazenada em uma posição de um vetor X, chamado *bag of words*. Existem várias maneiras de melhorar a construção do vetor X, selecionando as features mais significativas. Neste trabalho, consideramos duas possíveis,sendo elas :

1) TF-Term Frequency: : Cria-se um vetor T do mesmo tamanho de X, em que a i-ésima posição de T refere-se a frequência na qual w_i aparece no documento. Em nosso caso, restringimos as features, a serem palavras que aparecem pelo menos em 20% dos documentos. Tal abordagem evita que palavras pouco utilizadas sejam consideradas na avaliação do modelo.

2) TF-IDF: : Funciona assim como o TF, até o ponto em que multiplicamos TF por IDF (inverso da frequências nos documentos). A frequência do documento DF(i) é o número de vezes que uma palavra w_i aparece em todos os documentos. O IDF é dado por :

$$IDF(w_i) = log(\frac{D}{DF(w_i)})$$

Em que D representa o número de documentos.

Stoplists e Stemming também foram utilizados no projeto para garantir uma melhor seleção das features. Stoplists consistem em listas que contém palavras comumente utilizadas em uma língua (Português, Inglês etc) majoritariamente conjunções e artigos, tais como *a, as, os, para, por*, podendo então serem removidas do dicionário de features, por não representarem um ganho no aprendizado. Stemming, por sua vez, significa transformar palavras com um mesmo prefixo em sinônimos. Ex: Write e Writing, ambos possuem o prefixo "Writ", dessa forma ambos serão substituídos por "Writ" nos documentos. Isto é interessante, pois reduz o número de features e analisa palavras que possuam o mesmo sentido como uma só. Em nosso caso, as Stoplists foram modificadas para além de conterem conjunções, artigos etc, também possuíssem marcas de posse como "'s", pontuação e todas as palavras tiveram suas letras transformadas em letras minúsculas.

B. Algortimos de Classificação

1) Support Vector Machines: : Se duas classes são linearmente separáveis, então podemos encontrar um vetor de peso ótimo w. Os vetores de suporte definem dois hiperplanos, um para cada classe. A distância entre os vetores definem uma margem, a qual é maximizada quando a norma do vetor de peso w é mínima.

$$||w^*||^2 + C\sum_{i=1}^{N}\xi \tag{1}$$

A vantagem de um SVM Linear é que sua execução é rápida e a Regularização depende somente da constante C. Outra vantagem do SVM é que ele é intolerante ao tamanho das classes. Na maioria dos algoritmos de aprendizado de máquina, se existem muito mais exemplos de uma classe do que outra, ele tende à classificar melhor a classe com maior número de exemplos, portanto reduzindo o erro. Porém como o SVM não está diretamente preocupado em reduzir o erro e sim em separar os hiperplanos, o resultado final do SVM é independente da diferença do número de classes.

A principal desvantagem do SVM é o longo tempo de treino se existem muitas entradas, porém isto não aconteceu neste trabalho.

2) Multinomial Naive Bayes: : MNB é altamente utilizado para classificação de textos. O modelo captura a frequência das palavras nos documentos.

O pressuposto do Naive Bayes é de que as features são independentes.A vantagem de tal consideração é que a estimativa de p_θ é mais rápida e mais precisa com menos dados.

A probabilidade de uma palavra w é:

$$p(w|y) = \frac{count(w,c)+1}{count(c)+|v|} \tag{2}$$

3) Número de Features: : A escolha é entre usar algumas features ou todas as features. Em Reconhecimento de Textos, features são palavras. Uma vantagem de não utilizar muitas

features é que o algoritmo generaliza melhor, isto é, mantém uma boa performance para novos casos de teste.

Existem mecanismos desenvolvidos para encontrar o número ótimo de features, além disso pesquisadores já trabalharam em estudos que comparam vários modelos de classificação em relação ao número de features. No caso deste trabalho, foi utilizado a mesma quantidade de features para o Naive Bayes e SVM, porém não todas as features, por volta de 80% do total.

O problema de se encontrar o número de features ótimo, muito das vezes é a questão de tempo, pois exige maior tempo de treino. Em nosso caso, uma maior quantidade de features causava *overfitting* e uma menor quantidade diminuía gradualmente a acurácia. Em todos os testes o SVM se manteve como o modelo de maior acurácia.

4) Avaliação da Performance:

- **Precisão e Recall**: Em Recuperação da Informação, muitas vezes algo pode ter múltiplas classificações, como por exemplo Futebol ser categorizado como esporte e também negócios. No contexto deste artigo, temos somente duas classes, de tal forma que Precisão e Recall não são estritamente necessários, porém serão utilizados para calcular o F1 Score.

$$Recall = \frac{verdadeiros\_positivos}{total\_correto}$$

$$Precision = \frac{verdadeiros\_positivos}{total\_encontrado}$$

  Pense em um exemplo, em que pede-se para reconhecer alguns animais em uma image, estando nela contidos 11 cachorros e 3 gatos. O algoritmo de classificação acerta detecta 8 cachorros, sendo que dos 8, apenas 6 são de fato cães. Qual a precisão? 6/11, pois o algoritmo acertou 6 dos cachorros dentre os 11 existentes na imagem. Qual o recall? 6/8, pois representa os 6 cachorros reconhecidos corretamente dentre os 8 inicialmente apresentados.
- **F1 Score**: Em análise estatística e classificação binária, o F1 Score é a medida para testar acurácia. É necessário saber previamente a precisão e o recall.

$$F_1 = 2 * \frac{precisao * recall}{precisao + recall} \tag{3}$$

- **Curva ROC**: As Curvas ROC foram desenvolvidas no campo das comunicações como forma de demonstrar as relações entre sinal-ruído. Interpretando o sinal como os verdadeiros positivos (sensibilidade) e o ruído, os falsos positivos (1 - especificidade).
  A Curva ROC é um gráfico de verdadeiros positivos versus taxa de falsos positivos.
  Como pode ser visto nas figuras 1 e 2 da próxima seção, quando mais próximo do canto superior esquerdo, melhor é a sua acurácia, pois terá os valores de verdadeiros positivos maximizados e os de falsos positivos zerados.

IV. Resultados

Por fim, serão apresentados os resultados da pesquisa,com base nas metodologias de avaliação e modelos de classificação e seleção de features citados e discutidos na seção anterior (III).

A. Ambiente de Teste

O ambiente de desenvolvimento foi em uma máquina Dell Inspiron 15R, 8 GB de RAM, 1 TB de HD, Processador Intel Core i7-4500U CPU @ 1.80GHz x 4 e Sistema Operacional Ubuntu 16.04 LTS.

B. Recursos Técnicos

Foram utilizados para a criação desse projeto a linguagem de programação Python 3.5 com bibliotecas Numpy, scikilearn, pandas, strings, nltk e matplotlib.Todas foram utilizadas para desenvolvimento do código, exceto a última que foi usada para o *plot* dos gráficos.

C. Medidas de Acurácia

Foram realizados quatro testes, em que cada modelo de classificação se combina com outro modelo de seleção de features. Foram avaliados várias medidas de teste, de acordo com a tabela abaixo.

Modelos	AUC	Precision	Recall	Acurácia	F1 Score
SVM com Tf-idf	-	83,4%	83,4%	83,4%	83,4%
SVM com Tf	-	82,4%	82,2%	82,3%	82,3%
MNB com Tf-idf	90%	82.5%	82.5%	82.5%	82.5%
MNB com Tf	88%	88%	81,6%	81,6%	81,6%

TABLE I
PERFORMANCE DE CADA MODELO

De acordo com a Tabela 1, o SVM com Tf-Idf foi o modelo que propôs o melhor resultado, com Acurácia e F1 Score de 83,4% aproximadamente. É interessante ressaltar porém que todos os outros modelos obtiveram ótimos resultados e bastante próximos ao SVM com Tf-Idf. Isto é verdade pelo fato de que com o uso de Stoplists personalizadas,Stemming e outras técnicas de Processamento de Linguagem Natural no dataset, muito do trabalho feito pelo Tf-Idf também foi aplicado ao Tf. Porém, ainda sim, o Tf-Idf se mostrou mais proveitoso.

Lauren Dyson & Alden Golab em seus trabalhos obtiveram êxito com acurácia de 84,3% utilizando SVM e Tf-Idf. Em comparação, nossos resultados se mostram extremamente satisfatórios, inclusive pelo fato de que o Dataset deles possuía 11.000 samples, contra 6.355 nossos, ou seja, com mais testes nosso modelo poderia atingir um estado em que erro de treino e erro de teste iriam convergir e aumentar a capacidade do modelo.

D. Curvas ROC

As Curvas ROC são uma importante aliada para nos ajudar a validar os testes de verdadeiros positivos e falsos positivos em modelos probabilísticos, assim analisando com mais recursos o desempenho e acurácia de nosso modelo em termos de acertos e erros no espaço amostral.

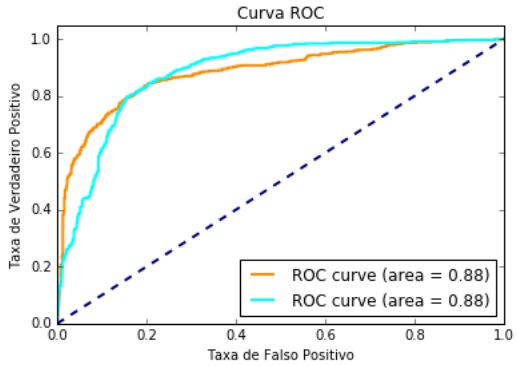

Fig. 1. Curva ROC Naive Bayes com TF

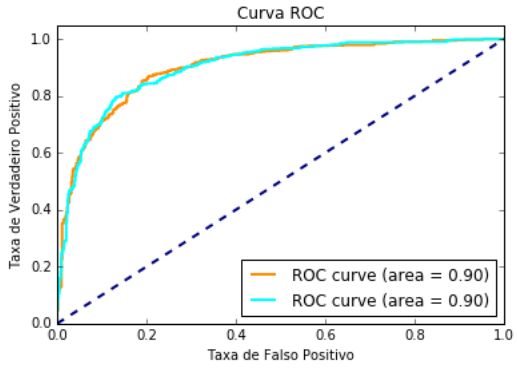

Fig. 2. Curva ROC Naive Bayes com TF

É possível observar dado as figuras 1 e 2 dos gráficos (a linha laranja representa as notícias falsas e a azul notícias verdadeiras), que com a alteração da feature-selection para Tf-Idf, houve um aumento da AUC (Area Under Curve), ou seja a área embaixo da curva. Isto significa que o modelo ficou mais sensivel e menos específico, e que aumentou o número de verdadeiros positivos e diminuiu o número de falsos positivos. Tal resultado já era esperado uma vez que a ROC é um reflexo da F1 Score, que por sua vez aumentou de Tf para Tf-Idf com o modelo de Multinomial Naive Bayes.

V. Conclusão

Baseado nas seções anteriores, nos estudos sobre seleção de features, modelos de classificação dentre outroas aspectos citados, chegamos às seguintes conclusões :

- **O SVM com Tf-Idf** foi o melhor modelo utilizado, alcançando uma acurácia de 83,4%. Por outro lado o Multinomial Naive Bayes demonstrou também uma alta acurácia, bastante próxima do SVM e com AUC de 90%.
- **Tf-Idf obteve melhores resultados** do que TF, mesmo com a adição de stoplist e stemming. Portanto é uma melhor escolha utilizar Tf-Idf do que TF para o trabalho deste artigo.
- **O aumento no número de samples provavelmente aumentaria a acurácia** do modelo, pois a capacidade do mesmo aumentaria. Além do mais, Lauren Dyson & Alden Golab obtiveram uma acurácia um pouco maior com um dataset de quase o dobro de tamanho.
- **A qualidade do Dataset influencia muito no resultado final**, assim a homogeneidade do Dataset, confiabilidade das fontes e distribuição das classes (aproximadamente 50% falsos e 50% verdadeiros) fez com que o modelo possuísse um valor alto de acerto e generalizasse bem pra novos casos de teste, como mostra o gráfico da Curva ROC.

É importante ressaltar que estas conclusões refletem os resultados do trabalho e não podem ser generalizadas para qualquer outra situação de Notícias Falsas sem maiores estudos.

Os resultados foram satisfatórios e mostraram o poder dos algoritmos de classificação de Aprendizagem de Máquina conjuntamente com os algoritmos de frequência de termos de Processamento de Linguagem Natural. Apesar do sucesso nos resultados, ainda não é possível provar a total generalização e correta predição do algoritmo para todas as notícias, pois existem outras validações a serem feitas, como por exemplo identificar os diversos tipos de notícias falsas e tratá-las de forma diferenciada.

Como pesquisa futura seria importante testar novos modelos de classificação para comparar os resultados, implementar novas técnicas de Processamento de Linguagem Natural, como por exemplo o LDA afim de identificar tópicos e relacioná-los com a eventualidade de serem falsos ou não. O dataset também é limitado e não possibilita analisar a fonte da notícia, uma possível abordagem seria agrupar notícias falsas dado a fonte e estudar cada palavra, sentença e parágrafo no texto de forma a classificá-los independentemente como influenciadores de uma classe, exemplo : ”Palavras mais falsas”, ”Sentenças mais falsas” etc.

A dificuldade em classificar notícias falsas, é que na grande maioria das vezes, o intuito da notícia é enganar, ou seja, ela foi criada propositalmente, o que dificulta bastante encontrar padrões dentro do documento.

VI. Referências

[1] Deception Detection for News: Three Types of Fakes Victoria L. Rubin, Yimin Chen and Niall J. Conroy Language and Information Technology Research Lab (LIT.RL) Faculty of Information and Media Studies University of Western Ontario, London, Ontario, CANADA

[2] Support Vector Machines for Spam Categorization Harris Drucker, Senior Member, IEEE , Donghui Wu, Student Member, IEEE , and Vladimir N. Vapnik

[3] A Comparison of Event Models for Naive Bayes Text Classification Andrew McCallum, Kamal Nigam

[4] Faking Sandy: Characterizing and Identifying Fake Images on Twitter during Hurricane Sandy Aditi Gupta, Hemank Lamba, Ponnurangam Kumaraguru, Anupam Joshi Indraprastha Institute of Information Technology, Delhi, India IBM Research Labs, Delhi, India University of Maryland Baltimore County, Maryland, USA.

[5] Fake News Detection Exploring the Application of NLP Methods to Machine Identification of Misleading News

Sources Lauren Dyson & Alden Golab CAPP 30255: Advanced Machine Learning for Public Policy University of Chicago Winter 2017

[6] Um Modelo Adaptativo para a Filtragem de Spam Ígor Assis Braga, Marcelo Ladeira Departamento de Ciência da Computação Universidade de Brasília (UnB) – Brasília, DF – Brasil

[7] On Building a “Fake News” Classification Model,George McIntire