# Detecção de notícias falsas

Matheus Eiji Endo
*Estudante de Ciência da Computação*
*Universidade de Brasília*
Brasília, Brasil
matheus.endo@hotmail.com

Johannes Peter Schulte
*Estudante de Ciência da Computação*
*Universidade de Brasília*
Brasília, Brasil
johpetsc@gmail.com

***Resumo*—Documento com a apresentação do problema e dos resultados para o projeto final da disciplina Fundamentos de Sistemas Inteligentes, onde será feita a análise dos resultados obtidos junto a comparação aos resultados de projetos já feitos anteriormente sobre a detecção de notícias falsas.**

***Index Terms*—Noticias falsas, *fake news*, eleições, fraude, sistemas inteligentes, aprendizado de máquina, *machine learning*.**

## I. Introdução

A utilização de notícias falsas(também frequente o uso do termo em inglês *fake news*) é um problema muito grave e complexo que tem ganhado muito destaque nos últimos anos para distribuir desinformação e boatos através de diferentes meios de comunicação. Esses meios vão desde jornais, até televisão, rádio e principalmente por meio de redes sociais. O intuito dessas notícias fabricadas são de ganhar benefícios financeiros ou políticos pela distribuição de informações falsas, exageradas ou sensacionalistas que alteram a percepção de um público alvo sobre um determinado assunto[1].


Figura 1. Redes Sociais se tornaram uma das principais ferramentas para publicações de notícias.

Com a popularização das redes sociais, as grandes massas ganharam acesso fácil e rápido à informação, o que facilitou a disseminação de notícias e a manipulação de ideias. Se utilizando desse fator, muitas organizações e pessoas com poder começaram a usar os meios de comunicação para distribuir desinformações que possam beneficiá-los de alguma forma[1].

Esses tipos de notícias já foram utilizados para vários contextos diferentes, como falsas alegações de que o aquecimento global não existe, para beneficiar o uso de combustíveis fósseis[2], também foram utilizadas nas eleições norte-americanas de 2016[3](quando começaram a ser mais discutidas na mídia) e nas eleições brasileiras de 2018(situação política atual)[4]. Tendo isso em mente e o contexto histórico no qual estamos atualmente(pós-eleições 2018), o trabalho será mais voltado para para os padrões utilizados nesse ano durante as eleições presidenciais.

As notícias falsas muitas vezes compartilham de semelhanças, como manchetes atraente ou fabricadas, fontes duvidosas ou ausentes, autores desconhecidos, datas contraditórias com os eventos, apelo à preconceitos e alguns padrões de linguagem feitos para atrair a atenção e curiosidade dos leitores. A partir da análise desses fatores, é possível detectar as notícias falsas sem precisar analisar a notícia em si.

O intuito desse projeto é fazer um *script* que consiga analisar esses padrões e detectar a veracidade dos textos analisados. O projeto não será feito a partir do zero, tendo em vista que já existem trabalhos acadêmicos sobre o assunto com o mesmo objetivo. Porém, como esse problema tem significâncias de âmbito muito grande, acreditamos que os resultados devem ser melhorados para que falsos negativos sejam quase que extintos, e assim o programa possa ser aplicado em situações reais sem o medo de falhas.

O trabalho que será usado como base será o "*Contributions to the Study of Fake News in Portuguese: New Corpus and Automatic Detection Results*"[5] realizado na USP em outubro de 2018, que apresenta uma base de dados com a maioria das notícias sobre política atual. Temos como objetivo utilizar diferentes algoritmos de aprendizado de máquina para tentar melhorar os resultados apresentados nesse estudo, e analisar o motivo para tal melhora.

## II. Modelo

Uma primeira dificuldade em relação à classificação de *fake news* é a forma de sua categorização, pois segundo o artigo de Rubin [8] notícias falsas podem ser divididas em 3 tipos:(i) as humorísticas, utilizados para diversão, usando sarcasmo para fazer sátiras e paródias; (ii) as de conteúdo falso, que tem o propósito claro de enganar e causar confusões, e (iii) os boatos, que não possuem confirmação e geralmente são aceitos publicamente. Por isso, para o intuito desse projeto as notícias humorísticas e que utilizavam fatos verdadeiros para apoiar conclusões enganosas não foram consideradas.

Para a realização do trabalho será utilizada um banco de dados que foi criada nesse ano para um projeto que será utilizado como base para o desenvolvimento do nosso. Para coletar esses dados o grupo da USP demorou alguns meses, o que não seria viável para o prazo que temos para o nosso projeto atual, por isso vamos utilizar os dados já coletados por eles.

A base de dados que será utilizada possui 7200 notícias, divididas igualmente entre falsas e verdadeiras. As notícias são todas apresentadas em texto e possuem tamanhos parecidos. Foram todas retiradas seguindo o intervalo de Janeiro de 2016 até Janeiro de 2018, com algumas exceções. Todas foram manualmente analisadas para que sejam totalmente falsas, e aquelas que continham alguma base de verdade foram retiradas. As notícias verdadeiras foram retiradas de forma semiautomática de fontes confiáveis(G1, Folha de São Paulo e Estadão) onde foram procuradas palavras chaves de notícias falsas para se obter um paralelismo entre as verdadeiras e falsas. Um exemplo disso pode ser visto nos textos a seguir, que tratam sobre o mesmo assunto envolvendo o fim do carnaval:

| Falso |
|---|
| Michel Temer propõe fim do carnaval por 20 anos, "PEC dos gasto". Michel Temer afirmou que não deve haver gastos com aparatos supérfluos sem pensar primeiramente na educação do Brasil. A Medida pretende cancelar o carnaval de 2018. |

| Verdadeiro |
|---|
| Michel Temer não quer o fim do Carnava por 20 anos. Notícias falsas misturam proximidade dos festejos, crise econômicas e medidas impopulares do governo do peemedebista. |

Na tabela a seguir, retira e traduzida do texto disponível em[5], podemos ver as quantidade de notícias para cada tema, e podemos ver a predominância de política, seguida de celebridades de TV e assunto sobre a sociedade. Isso acontece devido ao propósito que as notícias são utilizadas, como fraudar eleições, gerar cliques em páginas, difamar pessoas e tratar de assuntos que atingem a sociedade.

A próxima tabela também retira, traduzida e simplificada(valores arredondados) do texto[5], mostra as principais *features* que o algoritmo irá utilizar para determinar se as notícias são verdadeiras ou falsas. É possível ver grandes discrepâncias em algumas das categorias, como o tamanho dos textos(número de sinais, tanto letras como símbolos, número de letras e número de sentenças), e quantidade de erros ortográficos.

| Categoria | Exemplos | % |
|---|---|---|
| Política | 4,180 | 58 |
| Celebridades de TV | 1,544 | 21 |
| Sociedade | 1,276 | 17 |
| Ciência | 112 | 1 |
| Economia | 44 | >1 |
| Religião | 44 | >1 |

| Features(média) | Falso | Verdadeiro |
|---|---|---|
| Número de sinais | 216 | 1,268 |
| Número de letras | 119 | 494 |
| Tamanho das palavras | 4 | 4 |
| Relação sinais-letras | >1 | >1 |
| Número de sentenças | 12 | 54 |
| Tamanho das sentenças | 15 | 21 |
| Verbos | 14 | 13 |
| Substantivos | 24 | 24 |
| Adjetivos | 5 | 4 |
| Advérbios | 3 | 4 |
| Pronomes | 5 | 5 |
| Palavras vazias | 31 | 32 |
| Erros de ortografia | 36 | 3 |

Nessa última tabela retirada de[5], podemos ver *features* que foram criadas a partir da tabela anterior. Tendo como exemplo a incerteza, calculada pelo número de verbos modais e ocorrência de voz passiva, ou a falta de urgência que usou como medida o número de primeiros e segundos pronomes. É possível perceber que há diferenças, principalmente na terceira e quarta *features* apresentada.

| Features(média) | Falso | Verdadeiro |
|---|---|---|
| Pausalidade | 2.46 | 3.04 |
| Emotividade | 0.20 | 0.21 |
| Incerteza | 4.48 | 23.24 |
| Não-urgência | 0.62 | 4.05 |

## III. Solução e Análise

### A. Tf-idf

O termo tf-idf, abreviação do termo em inglês *term frequency-inverse document frquency* é uma forma de medir estatisticamente a importância de uma palavra presente em algum tipo de texto ou documento, frequentemente utilizada para recuperação de informações e mineração de dados. A comparação é feita de forma a evitar o viés de algumas palavras mais comumente utilizadas, fazendo comparações com documentos diferentes para detectar o padrão médio das palavras.[6]

Essa técnica foi aplicada na base de dados para relacionar as *features* com as palavras mais frequentes e relevantes para que a comparação de notícias falsas e legítimas seja feita. O cálculo é feito tendo um termo *i* em um documento *j*, e aplicamos na equação:

$$w_{i,j} = tf_{i,j} x log \frac{|N|}{|df_i|} \quad (1)$$

Onde *tf* é o número de ocorrências de *i* em *j*, *df* é o número de documentos contendo *i*, e *N* o número total de documentos.

### B. Processamento de Linguagem Natural

Mais conhecido pela sigla em inglês NLP, o processamento de linguagem natural é uma área de estudo que abrange inteligência artificial e linguística, que tem o objetivo de alcançar a compreensão e geração de línguas humanas naturais. O principal desafio desses estudos é encontrar uma forma de fazer os computadores extraírem sentido em textos e linguagens humanas, para isso existem vários modelos de aprendizagem mecânica e estatística que visam resolver esses desafios.[7]

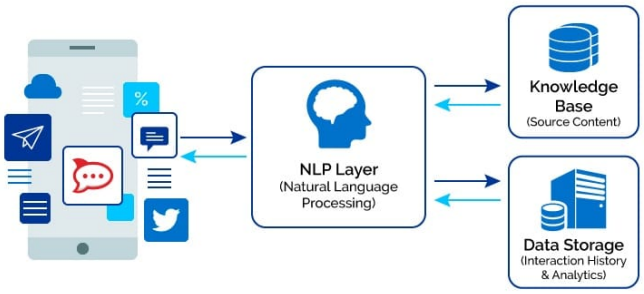


Figura 2. Abstração do funcionamento de um NLP.

Para esse trabalho, algumas técnicas do NLP foram utilizadas para realizar o pré processamento dos dados da nossa base de dados, como por exemplo, retirar os caracteres especias e deixar todas as letras minúsculas, uma forma de padronizar o texto e melhorar a interpretação das notícias para o computador.

### C. Algoritmos

Dados os diversos algoritmos de classificação estudados durante o decorrer da matéria os escolhidos para fazer a classificação das notícias foram Multinomial Naive Bayes e Support Vector Machine utilizando Stochastic Gradient Descent para o treinamento.

Como um breve resumo a esses algoritmos temos que : Multinomial Naive Bayes é um algoritmo que se baseia no teorema de Bayes:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (2)$$

Onde cada par de atributos é considerado independente dos demais e é considerada a distribuição multinomial.

Support Vector Machine(SVM) é um algoritmo que considera os dados como pontos e determina um hiperplano, uma reta, que separe as classes, maximizando a distância entre cada classe e o determinado hiperplano, esse hiperplano é determinado por um subconjunto de pontos da classes, que formam vetores, chamados de vetores de suporte, daí o nome do algoritmo.

Em conjunto com SVM optamos por utilizar Stochastic Gradient Descent(SGD) para o treinamento do algoritmo, SGD é um algoritmo de otimização usando na fase de treinamento, que encontra os pesos dos parâmetros com o objetivo de minimizar a perda. Para explicar SGD primeiro uma explicação sobre Gradient Descent(GD). Um gradiente é um vetor, ou seja ele possui direção e magnitude, o algoritmo de Gradient Descent multiplica o gradiente por um número chamado de *Learning rate* ou *Step size* a fim de determinar o próximo ponto, a escolha desses valores são muito importantes para o algoritmo uma vez que se for muito baixo o algoritmo demora demais para chegar em uma perda mínima e se for muito alto pode-se acabar pulando-a. É dessa forma que o algoritmo funciona, dado o gradiente calcula-se a mudança dos parâmetros dado o valor do *Learning rate*, faz-se o cálculo do novo gradiente com os novos valores dos parâmetros e se repetem esses passos até que a função de custo não se altere de forma significativa, para calcular o gradiente da função de custo é preciso rodar esse laço para cada instância do treino, de forma que possa ficar pesado o algoritmo. No algoritmo SGD a mudança dos parâmetros ocorre para cada instância, então ao invés de rodar no laço uma vez para cada instância ele só roda uma vez, ele tende a ser mais rápido que o GD, mas o seu caminho para o mínimo da função pode ser mais randômico.

A escolha desses algoritmos foi feita com base em artigos com propósitos similares ao nosso. O artigo base para esse projeto obteve melhores resultados utilizando svm e em [9] os melhores resultados foram obtidos utilizando tf-idf e Stochastic Gradient Descent para o treinamento dos algoritmos, além disso Multinomial Naive Bayes também é um algortimo de classificação conhecido para lidar com linguagem textual, por isso decidimos testar o Multinomial Naive Bayes e SVM com SGD no seu treinamento, ambos utilizando tf-idf como atributo.

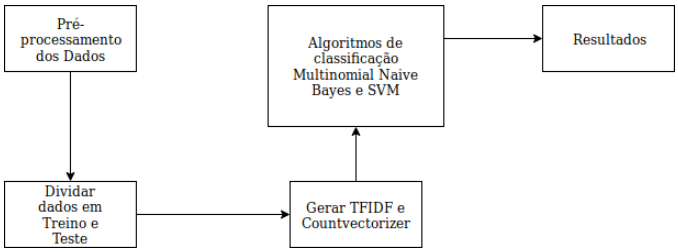


Figura 3. Diagrama do Projeto

### D. Implementação

O código foi implementado em Python versão 2.7 e as principais bibliotecas utilizadas foram:

1) *Pandas*
2) *SkLearn*
3) *Numpy*
4) *re*

Os dados disponíveis utilizados podem ser encontrados aqui. Eles são divididos entre *full text* e *size normalized text*, utilizamos somente os textos com o tamanho normalizado, pois o nosso intuito utilizar os valores de tf-idf como atributos.

O primeiro passo necessário para manipular os dados foi a criação de um dataframe da biblioteca *Pandas* e colocar os dados dos diversos arquivo em txt para esse dataframe. Para isso foi definida uma função que cria dois dataframes, 1 para a pasta de notícias falsa e outra para a de notícias verdadeiras, obtendo os textos dos arquivos e os transformando em unicode(pois devido à língua portuguesa utilizar acentos obtivemos problemas na hora do pre-processamento desses textos) e removendo todos os acentos, são é substituído por sao, por exemplo.

Após isso é feito o pré-processamento dos dados, onde todas as letras são transformadas em minúsculas e os dígitos e caracteres especiais são removidos. Os dataframes são então adicionados de uma coluna label, 1 para notícia verdadeira e 0 para notícia falas, e os dois dataframes são transformados em 1.

Com os dados já arrumados, utilizamos as funções da biblioteca *SkLearn CountVectorizer* e *TfidfTransformer* para obtermos os valores de tf-idf, a função *GridSearchCV* é utilizada para obter os melhores parâmetros utilizados nas funções de classificação.

## IV. Resultados

Todos resultados foram obtidos utilizando a função *GridSearchCV* do *SkLearn* com cv=3, ou seja, com validação cruzada. Além disso foi utilizada a função *Pipeline* para aplicar os transformadores.

Na figura 4 é possível ver a matriz de confusão do algoritmo Multinomial Naive Bayes, e na figura 5 seu classification report, sendo que foram utilizados os melhores parâmetros decorridos do gridsearchcv. Sendo eles: 'vect__ngram_range': (1, 2), 'tfidf__use_idf': False e 'clf-svm__alpha': 0.001.

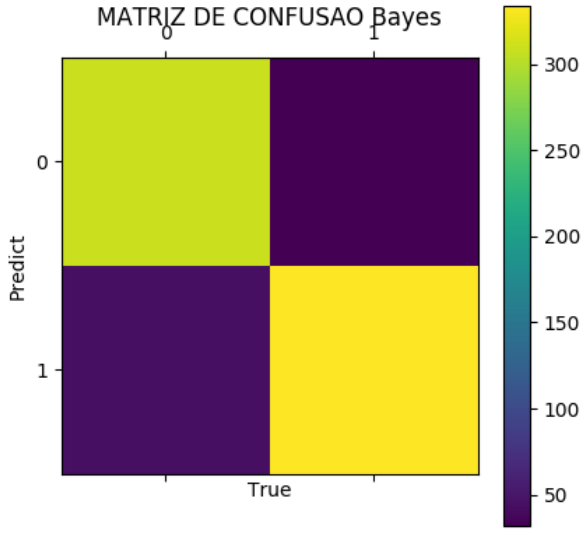


Figura 4. Matriz de Confusão do Multinomial Naive Bayes

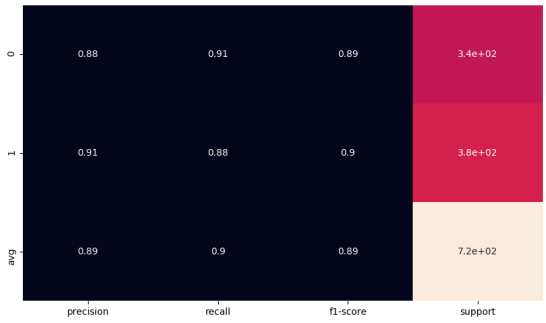


Figura 5. Classification Report Multinomial Naive Bayes

O algoritmo Multinomial Naive Bayes teve uma acurácia média de 88% testando o algoritmo 5 vezes. É interessante notar que para este algoritmo o melhor parâmetro foi não usar tf-idf, de forma que este atributo não melhorava a acurácia para este algoritmo.

A figura 6 representa a matriz de confusao do classificador SVM com SGD já a figura 7 apresenta *Precision, recall, f1-score* e *support* do mesmo classificador. Esses resultados utilizando o *SGDClassifier* com os parâmetros: loss = 'hinge', penalty = 'l2', alpha = 1e-3, random_state = 42, max_iter = 500 e tol = 1e-3. Os parâmetros que geraram os melhores resultados foram: 'vect__ngram_range': (1, 2), 'tfidf__use_idf': True e 'clf-svm__alpha': 0.001. Esses melhores parâmetros foram então utilizados em dados novos, separados antes da fase de treino, para obtenção dos resultados.

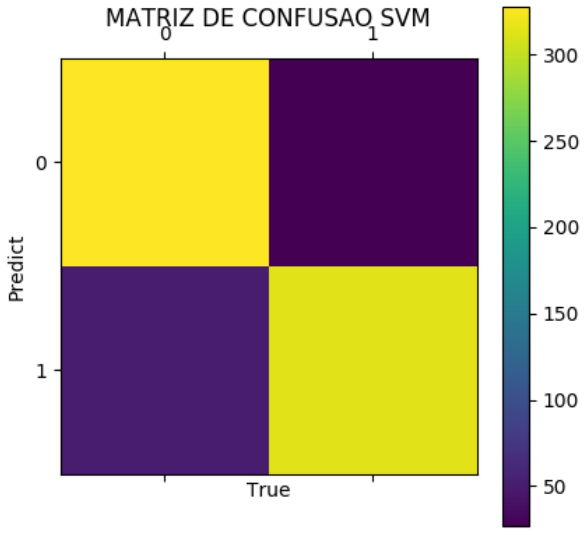


Figura 6. Matriz de Confusão do SVM com SGD

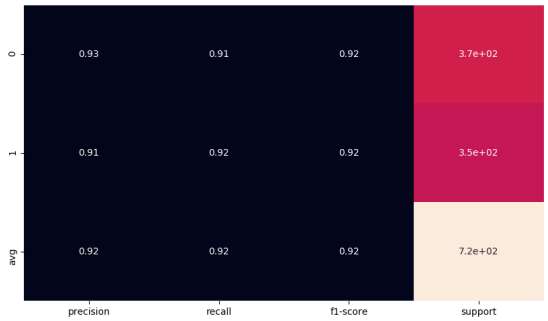


Figura 7. Classification Report do SVM com SGD

Percebe-se que o algoritmo obteve resultados bons, com uma precisão em torno de 91%, superando a precisão do classificador Multinomial Naive Bayes. Os parâmetros considerados melhores são bem similares aos do algoritmo anterior, porém, para SVM o uso de tf-idf melhorou sua acurácia.

## V. Conclusões

A classificação de notícias em falsas ou verdadeiras é de extrema importância para os dias atuais, e utilizar aprendizagem de máquina pode ser uma importante ferramenta nesse quesito. Utilizando como base o artigo [5] e o banco de dados disponíveis na web, conseguimos classificar notícias como falsas ou verdadeiras com uma acurácia de 91% superando os números obtidos no artigo base.

O uso de NLP para o tratamento dos dados e de tf-idf para conseguir uma métrica de importância das palavras nos possibilitou classificar notícias com alta acurácia.

## Referências

[1] Hunt, Elle. «What is fake news? How to spot it and what you can do to stop it». The Guardian. Consultado em 15 de janeiro de 2017.

[2] Moraes, Eduardo Cruz Carneiro, Erica Mariosa Moreira. "A evolução do jornalismo na divulgação científica". ComCiência — Revista eletrônica de jornalismo científico, 10/04/2018.

[3] Wendling, Mike. "Como o termo 'fake news' virou arma nos dois lados da batalha política mundial". BBC Brasil, 27/01/2018

[4] "OEA destaca uso 'sem paralelos' das fake news nas eleições brasileiras". Rede Brasil Atual, 26/10/2018

[5] Rafael A. Monteiro, Roney L. S. Santos1, Thiago A. S. Pardo1, Tiago A. de Almeida, Evandro E. S. Ruiz, and Oto A. Vale. "Contributions to the Study of Fake News in Portuguese: New Corpus and Automatic Detection Results"http://conteudo.icmc.usp.br/pessoas/taspardo/PROPOR2018-MonteiroEtAl.pdf. Oct/2018

[6] Rajaraman, Anand; Ullman, Jeffrey David. Data Mining. [S.l.: s.n.] p. 1-17

[7] Hutchins, J. (2005). "The history of machine translation in a nutshell"

[8] Rubin, V.L., Chen, Y., Conroy, N.J.: Deception detection for news: Three types of fakes. Proceedings of the Association for Information Science and Technology 52 (1), 1–4 (2015)

[9] S. Gilda, "Evaluating machine learning algorithms for fake news detection,"2017 IEEE 15th Student Conference on Research and Development (SCOReD), Putrajaya, 2017, pp. 110-115.