Detecção notícias falsas em publicações do Twitter

Stephane de Fretias Schwarz
Anderson Rocha

I. Introdução

Nos últimos anos, as redes sociais se tornaram um palco perfeito para disseminação e ampliação de notícias entre os internautas, contudo, nem todo conteúdo que circula na Web faz referência a um evento ou fato real. Notícias falsas, comumente chamadas fake news, são criadas com diferentes propósitos, difamação de candidatos políticos, manipulação de debates e reportagens, camuflagem de evidências, difusão de pânico e caos entre as pessoas são apenas alguns exemplos do que as fake news podem causar. Contudo, muitas vezes, tais notícias se tornam “virais” e são tomadas como verdades, podendo acarretar consequências severas e impactos negativos imensuráveis a um indivíduo ou sociedade. Para minimizar as perturbações causadas pelas informações falsas que inundam as redes, é essencial o desenvolvimento de soluções capazes de detectar, controlar e evitar a propagação de conteúdos ilegítimos nas mídias sociais, automaticamente.

Nesse sentido o objetivo desse trabalho é identificar uma publicação falsa proveniente do *Twitter.* A analise foi baseada na decomposição de textos em sacolas de palavras com normalização por frequência de palavras TF-IDF, *pad sequences* e *embeddings.* Os dados foram obtidos do desafio lançado por [1], o qual contém cerca de vinte mil posts do T*witer* para treinamento e validação, divididos em três classes, (fake, humor e real) e dez mil para teste. Os algoritmos de classificação utilizados foram *random forest e naïve bayes*. Nas próximas subseções serão apresentados os passos tomados para extração de características e configuração dos parâmetros adotados, resultados e conclusões.

II. Descrição dos dados

O conjunto de dados é composto por publicações extraidas do *Twitter* até o ano de 2015, em diferentes línguas, relacionadas a alguns eventos que marcaram determinados anos, como por exemplo a inundação da estação ferroviária subterrânea de Hoboken, a explosão de uma bomba na maratona de Boston, entre outros.

III. Extração de características

Uma das tarefas primordiais nesse trabalho foi o pré processamento dos dados antes a etapa de extração de características. Como os textos são provenientes do Twitter existem diversos símbolos relatívos ao jargão da mídia, isso é, # hashtag, :-) emoji, @referência entre outros. Para lidar com isso, foram criadas expressões regulares que substitue o ícone em questão por uma tag que indica sua presença, dessa forma não é descartada sua existência, no entanto não é a estrategia mais eficiente, visto que informações preciosas podem estar sendo ignoradas. Muitas vezes uma publicação contém apenas tais símbolos e um link para uma reportagem do evento ocorrido, isso significa se forem substituidos completamente por uma tag restará apenas o link para inferência e se o link não estiver disponível não haverá absolutamente nada pertinente para análise. Ainda nesta fase, todos as publicações foram traduzidas para o inglês, através da implementação do *google translator*, disponível em *python*. Como era de se esperar existem algumas perdas nesse processo, contudo é pouco relevante.

A partir desse ponto, foram aplicadas as técnicas de vetorização no conjunto de dados. O primeiro experimento testado foi *bag-of-word* com normalização por frequência de termos, em outras palavras TF-IDF (*Term frequence, inverse term frequence*). A ideia central consiste em contabilizar a quantidade de vezes que cada termo aparece na coleção, no entanto para evitar a presença de palavras que não expressão a importância para um

documento foi utilizada a métrica IDF, que basicamente computa o logaritimo da quantidade total de documetos dividido pela quantidade de documentos que contém o termo em evidência. Assim, palavras que aparecem em poucos documentos são mais pertinentes. Essa técnica foi usada para extração de um, dois, três e quatro gram, sendo que os melhores resultados foram obtidos com dois-gram, mais detalhes serão apresentados na próxima subseção. Além disso, tais abordagens foram testadas com e sem a presença de stop words e palavras flexionadas ou derivadas.

Outra forma de representar textos em valores numéricos é mediante o uso de *one hot encoding*. Nessa interpretação cada palavra é expressada como um número inteiro dentro de um intervalo, isso implica que o tamanho do vetor de caracteristicas é limitado por tal valor. Esses dígitos são atribuidos sem levar em consideração qualquer relação semântica ou frequência de palavras, podendo acarretar em diversos infortúnios. Para representar uma coleção de documentos essa implementação pode ser combinada com *pad sequences* que transforma uma lista de "palavras" em um *array* de duas dimensões, onde as linhas representam cada amostra e as colunas são os valores atribuidos as palavras. Para documentos com dimensões menores as do vetor final basta completar com zeros. Não é difícil perceber que essa abordagem apresenta diversas desvantagens quando comparada a anterior, já que não há qualquer relação de similaridade entre palavras, apenas entre documentos visto que são equidistantes uma das outras no espaço característico, entrementes quando combinado com pesos de *embeddings* pré treinadas com conjuntos grandes de dados, como por exemplo *GoggleNews*, *os* resultados obtidos são melhores.

As *embeddings* geradas pela rede *Word2Vec* treinadas com o conjunto de dados do *Google News* gera para cada palavra um vetor com 300 elementos. Uma das abordagens para representar um documentos é obtendo a média dos valores dos termos de uma sentença. Assim, combinando as representações das palavras que compoe a sentença é possível descrever uma frase sem descartar informações semânticas.

Nas imagens 1 e 2 é possível visualizar a distribuição dos documentos no espaço característico após a transformação dos textos em vetores segundo as estratégias supramencionadas. Na figura 1.B a separabilidade dos artigos de acordo com suas classificações é mais clara que as demais, no entanto ainda há uma correlação grande entre os grupos.

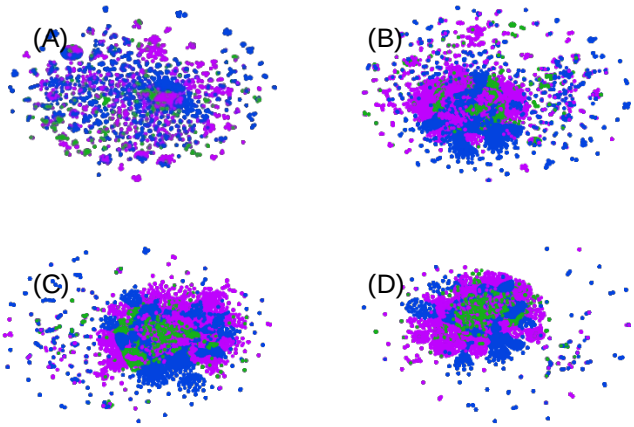

Imagem 1: Resultados do T-SNE para 1, 2, 3 e 4 gram, que são respectivamente (A), (B), (C) e (D). Pontos vermelhos indicam notícias falsas, azuis - reais e verdes notícias com senso de humor. Dados pré-processados e traduzidos.

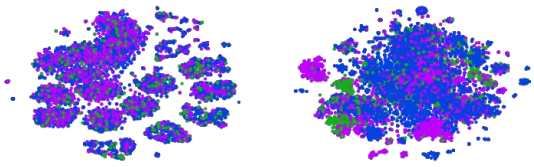

Imagem 2: Resultados do T-SNE para *pad sequences* (direita) e *embeddings* (esquerda). Pontos vermelhos indicam notícias falsas, azuis - reais e verdes notícias com senso de humor. Dados pré-processados e traduzidos.

Embora as *words embeddings* conservem a relação semântica do texto, como é possível ver na imagem 2 da esquerda, não há uma divisão clara entre as classes dos documentos. A fim de

conquistar resultados melhores foram extraidos dos textos a polaridade do sentimento que o texto transmite, isso é positivo (bom) ou negativo (mau), entretanto os resultados permaneceram semelhantes aos obtidos anteriormente.

É importante salientar que cada documeto é representado por um vetor com 500 características, número esse escolhido após alguns experimentos que evidenciou que aumentado a quantidade de dados os resultados permaneceram equiparáveis.

IV. Arcabouço proposto

Nesse trabalho foram explorados três algoritmos de classificação que são, *random forest e naive bayes*, visto que trabalhos anteriores [2], [3] e [4] conseguiram resultados promissores.

Naive bayes é uma classificador probabilístico baseado no teorema de Bayes, que assume a independência das características para a predição. Embora a maioria das caracteristicas realmente são independentes nesse caso, o classificador não apresentou bons resultados em todas as amostras de dados extraídas do conjunto, como pode ser observado na tabela 1. Além disso, tem um custo computacional consideravelmente alto quando comparado a outros classificadores binários.

Sabendo que grande parte dos dados são categóricos, e que as arvores de decisão funcionan de maneira satisfatória com esse tipo de amostra, o método *Random forest* foi usado para generalização do problema. Esse algorítmo é baseado em *ensemble learning*, o qual gera diversas árvores classificadoras e combina o resultado de cada uma, por essa motivo tal modelo se torna um poderoso algoritmo para classificação. Para determinar o número de árvores, a classificação dos documentos foi feita variando os valores entre 5 a 100. Como pode ser visto na imagem 3, o número que proporcionou a melhor acurácia foi 38.

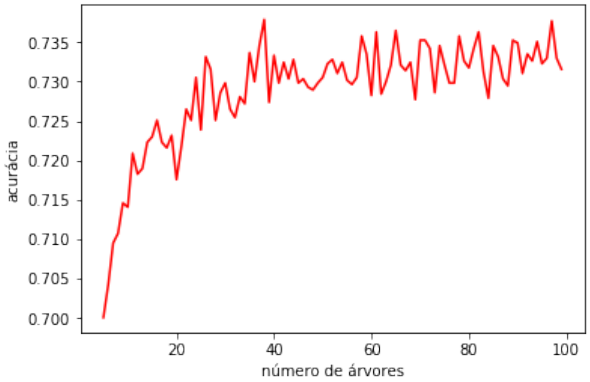

Imagem 3: Classificação com *random forest* variando o número de árvores do modelo. Eixo y representa a acurária da classificação e o eixo x o número de árvores. Valores obtidos do conjunto de validação.

Na tabela 1 é possível visualizar os resultados da média harmónica F1 e da acurácia dos algorítimos quando treinados e testados com os diferentes conjuntos de dados extraídos das publicações após o pré processamento.

	Random Forest		Naive bayes	
	F1	ACC	F1	ACC
1-gram	0.7234	0.7308	0.5582	0.5333
2-gram	0.6300	0.6462	0.4752	0.4563
3-gram	0.5420	0.5895	0.4035	0.3948
4-gram	0.4754	0.5545	0.3453	0.3519
Hot-enc.	0.4734	0.5529	0.3453	0.3519
Embed.	0.6577	0.6809	0.4765	0.4843

Tabela 1: Resultados obtidos da classificação com diferentes características extraídas dos documentos de teste.

O algortimo que melhor classificou os documentos foi *Random Forest* sendo que as características que geraram os melhores resultados foram, 1-gram, *word embeddings* e 2-gram.

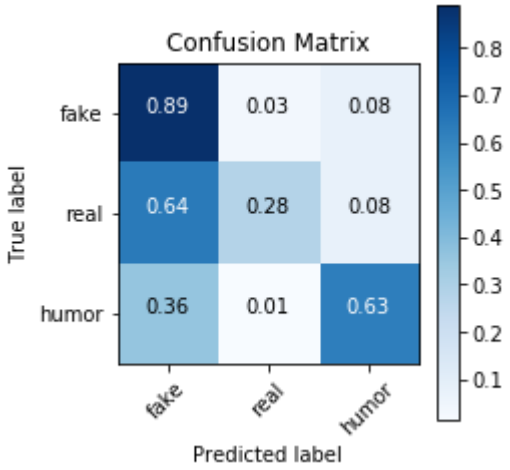

Imagem 4: Matriz de confusão para classificação com embeddings.

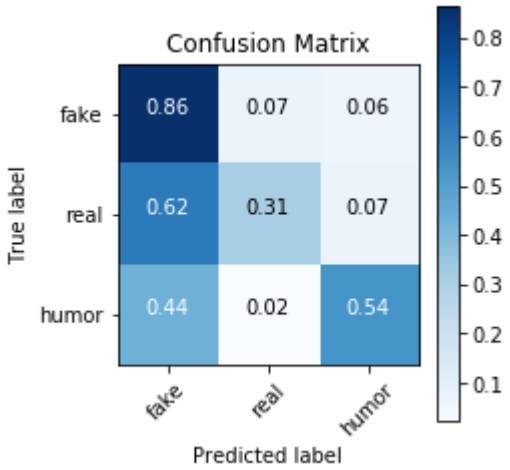

Imagem 5: Matriz de confusão para classificação com TF-IDF 2-gram.

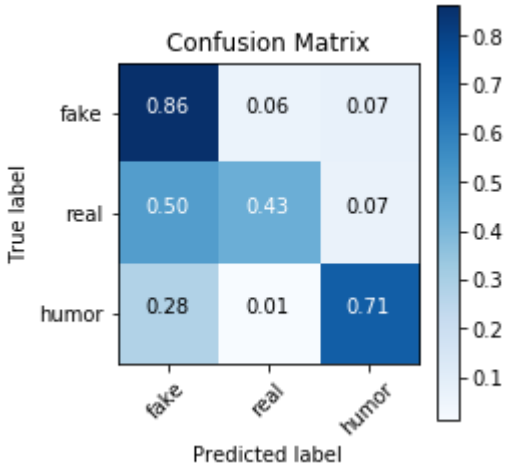

Imagem 6: Matriz de confusão para classificação com TF-IDF 1-gram.

Como é possível perceber nas figuras 4, 5 e 6 o classificador tem grande dificuldade para predizer documentos que apresentam conteúdos verdadeiros. Em todos os casos mais de 50% do conjunto classificado como real foi predito na classe falsa. A hipótese para que isso ocorra é que a maioria das notícias falsas contém muitos ícones como *hashtags* e *emojis*, desses 50% classificados como falsos, mas na verdade são reais também contém muitos simbolos tais como esses. Isso india que esse tipo de tratamento nesse caso não é o mais adequado visto que a inferência é feita a partir de dados ruidosos. Nos casos de checagem de fatos, essa abordagem não deve ser adotada, mas sim usar esses símbolos para participar ativamente do processo de classificação.

Já a classe humor, em todos os casos, foi classificado corretamente pelo menos 50% dos dados. Isso acontece pois textos com senso de humor contém palavras chave que indicam a presença de sarcasmo, ironia etc, contribuindo assim para uma categorização mais acertada.

V. Conclusão

Nesse trabalho foram explorados alguns caminhos para identificação de notícias falsas mediante o uso de representações vetoriais e texto e análise de sentimento. Foi possível perceber que o algortimo *random forest* proporcionou os melhores resultados, aproximadamente 73% de acurácia, quando treinados com sacola de palavras com normalização TF-IDF para 1 e 2 gram, bem como com as *word embeddings* obtidas com a rede pré treinada no conjunto *Google News*.

Além disso, camuflar elementos do texto com tags, nesse caso não é a melhor alternativa, visto que informações importante podem estar sendo desconsideradas, mas ao invés disso combinar com checagem de fatos e *event repurposing* possivelmente os resultados obtidos seriam consideravelmente melhores.

Referências

[1] C. Boididou et al., “Verifying Multimedia Use at MediaEval 2016”, Sum SITE Central Europe Workshop, vol. 1739, pp. 4–6, 2016.

[2] C. Boididou et al., “The VMU participation @ verifying multimedia use 2016”, Sum SITE Central Europe Workshop, vol. 1739, pp. 5–7, 2016.

[3] C. Maigrot, V. Claveau, E. Kijak, and R. Sicre, “MediaEval 2016: A multimodal system for the Verifying Multimedia Use task”, Sum SITE Central Europe Workshop, vol. 1739, 2016.

[4] C. Boididou et al., “Verifying Multimedia Use at MediaEval 2016”, Sum SITE Central Europe Workshop, vol. 1739, pp. 4–6, 2016.