Avaliação de Modelos de Redes Neurais Recorrentes para Anonimização de Textos em Português

Antônio M. R. Franco[1], **Leonardo B. Oliveira**[1]

[1]Universidade Federal de Minas Gerais

{franco,leob}@dcc.ufmg.br

Abstract. *Currently, there are several approaches to provide anonymity on the Internet. However, one can still identify anonymous users through their writing style. With the advances in neural network and natural language processing research, the success of a classifier when accurately identify the author of a text is growing. On the other hand, new approaches that use recurrent neural networks for automatic generation of obfuscated texts have also arisen to fight anonymity adversaries. In this work, we evaluate two approaches that use neural networks to generate obfuscated texts. In our experiments, we compared the efficiency of both techniques when removing the stylistic attributes of a text and preserving its original semantics. Our results show a trade-off between the obfuscation level and the text semantics.*

1. Introdução

O anonimato na Internet é uma condição que pode ser requerida em diversas situações. Um denunciante relatando um ato fraudulento através de um canal de denúncias pode querer submeter seu relato de forma anônima, assim como um cliente que submete uma avaliação negativa sobre um produto ou serviço pode requerer seu anonimato. Em casos mais graves, o anonimato pode ser requerido para preservar a vida de um denunciante[1].

Um usuário pode aplicar diversas técnicas para ocultar seus atributos de identidade. Um exemplo é a rede Tor[2], que pode ser utilizada para ocultar os endereços de rede dos seus usuários. No entanto, mesmo utilizando estas técnicas uma pessoa ainda poderia ser identificada pelo seu estilo de escrita [Narayanan et al. 2012] utilizando Processamento de Linguagem Natural (PLN).

Para proteger o seu anonimato, um usuário também pode utilizar PLN para ofuscar os atributos estilísticos do seu texto e reduzir as chances de sucesso do adversário, como no exemplo da Figura 1. Duas técnicas que utilizam redes neurais recorrentes para geração automática de textos ofuscados foram propostas recentemente por Emmery et al. [Emmery et al. 2018] e Shetty et al. [Shetty et al. 2018]. O grande desafio destas abordagens – além de mascarar o estilo de escrita do autor – é preservar a semântica do texto gerado.

Objetivo. O nosso objetivo neste trabalho é avaliar a performance de dois métodos de ofuscação de texto que utilizam redes neurais recorrentes para gerar textos automaticamente em um estilo de escrita diferente. O escopo de avaliação deste trabalho é um conjunto de dados com textos da língua portuguesa.

[1]https://www.nytimes.com/2019/03/18/world/africa/south-africa-anc-magaqa-killing-arrests.html

[2]https://www.torproject.org

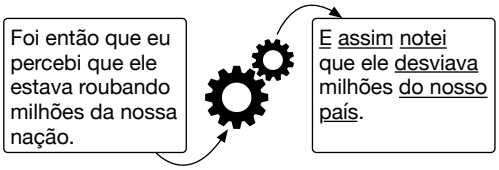

Figura 1. Exemplo de texto gerado por um ofuscador de estilo.

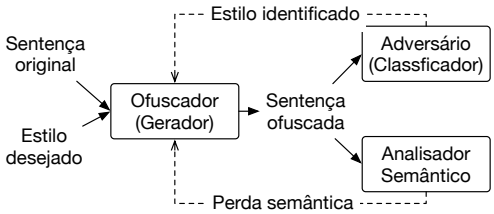

Figura 2. Treinamento do ofuscador de textos.

Contribuições. Nossas contribuições com este artigo são: (i) a avaliação da performance do estado da arte em um corpo de textos escritos em português; e (ii) a disponibilização de modelos pré-treinados para ofuscação de textos escritos em português. Até onde sabemos, este é o primeiro trabalho a focalizar a ofuscação de textos escritos em português.

2. Fundamentos

O problema de ofuscação de textos é similar a um problema de tradução de máquina, onde o objetivo é traduzir um texto de um idioma para o outro [Shetty et al. 2018]. A diferença está no alvo da tradução: em vez de traduzir um texto para outro idioma, o objetivo de um ofuscador é transformá-lo em outro texto escrito no mesmo idioma, porém com um estilo de escrita diferente. Muitos problemas de tradução de máquina atualmente são resolvidos com técnicas de aprendizagem profunda (*deep learning*) aplicadas à PLN, e nesta seção serão apresentados os conceitos fundamentais para compreender as abordagens avaliadas.

Redes Neurais Recorrentes. No campo de aprendizado de máquina, existem arquiteturas de redes neurais que são mais eficientes para resolver problemas de PLN. As redes neurais profundas tradicionais não memorizam informações relacionadas entre as entradas, e isso é um problema em tarefas de PLN [Goodfellow et al. 2016]. Esse problema pode ser ilustrado com o seguinte exemplo. Suponha que você quer traduzir a frase "*Minha fruta favorita é manga.*" para inglês. Se a palavra manga for analisada fora de contexto, um tradutor não saberá se a tradução correta é *mango* ou *sleeve*. Redes neurais recorrentes resolvem esse problema ao memorizar as informações que são aprendidas anteriormente.

Identificação de Autoria. Um adversário pode utilizar uma rede neural recorrente para analisar um conjunto de textos e extrair características intrínsecas dos estilos de escrita dos seus autores. Estas características, por sua vez, podem ser utilizadas para treinar um modelo de aprendizagem profunda com o objetivo de classificar um texto de entrada como sendo ou não escrito por um autor específico. Dependendo do volume de textos no conjunto de dados utilizados para teste, um modelo de aprendizado de máquina pode obter performance superior à análise manual de um perito experiente em estilometria [Varela et al. 2011].

Ofuscação de Textos. O problema de ofuscação de textos, portanto, consiste em modificar um texto de entrada It de modo que o texto modificado Ot não possua os atributos estilísticos que podem ser utilizados para identificar o seu autor. Este problema pode ser visto como a tradução de um texto de um estilo para o outro, porém dentro da mesma linguagem. O principal desafio em tarefas de ofuscação texto é a preservação da semântica do texto original [Emmery et al. 2018]. Neste trabalho, nós avaliamos duas abordagens de ofuscação de textos baseadas em redes neurais recorrentes: ofuscação por invariância [Emmery et al. 2018] e ofuscação por transferência de estilo [Shetty et al. 2018].

3. Desenvolvimento

Esta seção descreve os conjuntos de dados que utilizamos para treinar e testar modelos, assim como os componentes que implementamos para suportar a nossa avaliação.

3.1. Conjuntos de dados

Seguindo a abordagem proposta em [Emmery et al. 2018], nós utilizamos um conjunto de dados extraído de diferentes traduções em português da bíblia para treinar os modelos. A bíblia foi escolhida porque os versículos de duas traduções diferentes podem ser combinados para formarem pares de sentenças que possuem o mesmo sentido mas que foram escritas em estilos diferentes. Extraímos as traduções do portal `bibliaonline.com.br` utilizando um *script* automático para *scrapping* de conteúdo web. As versões utilizadas foram a Nova Versão Internacional, Almeida Corrigida Fiel e Almeida Revista e Atualizada. No total, o nosso conjunto de dados ficou composto por 31102 versículos de cada versão. Nós permutamos os versículos extraídos para construir pares entre traduções distintas e, no total, obtivemos 186612 pares.

Nós utilizamos um conjunto de dados diferente para testar a eficiência dos modelos de ofuscação. O conjunto de testes foi formado por uma base disponibilizada em [Varela et al. 2011], que é composta por 3000 textos em português escritos por colunistas de portais de notícias diversos. Esta base foi escolhida porque os textos são segmentados por autores, e esse é um requerimento para treinar o classificador de textos, pois são necessárias anotações para distinguir entre os diferentes estilos de escrita que podem ser identificados.

3.2. Ofuscação por invariância (Emmery et al.)

Para implementar o modelo da abordagem de ofuscação por invariância, nós nos baseamos no código disponibilizado pelos autores no GitHub[3]. Nós implementamos a rede neural proposta pelos autores utilizando a linguagem Python 3.7 e o *framework* PyTorch.

A arquitetura da rede é composta por um *encoder*, que é responsável por processar o texto de entrada com células *Long short-term memory* (LSTM) [Hochreiter and Schmidhuber 1997] e gerar um vetor de contexto que armazena as propriedades das sentenças; e por um *decoder*, que recebe como entrada o vetor de contexto e gera a sentença alvo no estilo de escrita desejado. Além disso, esta rede neural possui uma camada de *Gradient Reversal Layer* (GRL) [Ganin and Lempitsky 2014] que é utilizada para aprender os atributos invariantes que não devem ser modificados ao gerar o texto ofuscado.

3.3. Ofuscação por transferência de estilo (Shetty et al.)

A implementação do modelo de ofuscação por transferência de estilo foi baseada no código disponibilizado pelos autores[4]. A arquitetura desta abordagem é composta por um gerador de textos, que funciona de maneira análoga a um tradutor; e um classificador de textos, que é utilizado para identificar as classes dos textos gerados pelo gerador. O gerador é uma rede neural recorrente com um *encoder* que recebe a sentença a ser ofuscada e gera um vetor de contexto; e um *decoder* que recebe o vetor de contexto e gera a sentença

[3] `https://github.com/cmry/style-obfuscation`

[4] `https://github.com/rakshithShetty/A4NT-author-masking`

de saída. O classificador é uma rede recorrente convencional treinada para identificar atributos estilísticos dos textos. O gerador e o classificador são conectados um ao outro em uma arquitetura de *Generative Adversarial Network* (GAN) [Goodfellow et al. 2014].

Para treinar o modelo de ofuscação por transferência de estilo, nós executamos os seguintes passos. (i) Treinamos o classificador de estilos proposto pelos autores utilizando o conjunto de dados das versões da bíblia em português. (ii) Treinamos os *autoencoders* de cada estilo do conjunto de dados para gerar os textos iniciais. (iii) Treinamos o gerador de textos final utilizando o *framework* GAN com o classificador e com os *autoencoders* treinados previamente. O processo de treinamento do gerador de textos é ilustrado na Figura 2.

4. Avaliação

Esta seção descreve as métricas utilizadas para comparar os modelos, os experimentos realizados e os resultados obtidos.

4.1. Métricas

Ofuscação. Para mensurar a performance do ofuscador ao esconder os atributos estilísticos do texto, nós utilizamos o *caravel* [Bagnall 2015], que foi a ferramenta que obteve a melhor nota na tarefa de identificação de autoria do PAN[5]. Esta é uma ferramenta adversária, ou seja, ela tenta revelar se um texto foi escrito por um autor ou não. Para isso, ela precisa ser previamente treinada em um conjunto de textos de diferentes autores para que, ao receber um novo texto, ela retorne a probabilidade de cada autor ter escrito o texto em questão. Para computar a nota final do ofuscador de textos, *caravel* utiliza o F1-score computado sobre o conjunto de testes.

Semântica. Para mensurar a consistência semântica entre as sentenças originais e as sentenças geradas pelos ofuscadores, nós utilizamos as métricas METEOR [Banerjee and Lavie 2005] e BLEU [Papineni et al. 2002]. Nós escolhemos estas métricas porque elas são utilizadas para medir a qualidade das sentenças geradas por trabalhos de tradução de máquina. Nós utilizamos o *nlg-eval*[6] para analisar os textos ofuscados e coletar estas métricas.

4.2. Experimentos

Nós treinamos os modelos de Emmery et al. e de Shetty et al. com o nosso conjunto de dados de treinamento e obtivemos taxas de perplexidade de 3.14 e 4.19 respectivamente. Nós utilizamos os mesmos hiper-parâmetros que foram utilizados pelos autores para treinar as configurações que obtiveram melhor performance em seus experimentos.

Após treinar os modelos, nós executamos o nosso *script* de ofuscação para gerar amostras de textos ofuscados para cada um dos 3000 textos do conjunto de dados de testes. Para cada amostra avaliada, nós mantivemos o texto original, o texto ofuscado por Emmery et al. e o texto ofuscado por Shetty et al.. Na sequência, nós treinamos o classificador de textos para reconhecer os atributos estilísticos do conjunto de dados de testes e seus respectivos autores. Após o treinamento, nós executamos o classificador para

[5]PAN é uma série de eventos científicos e tarefas compartilhadas para resolver problemas de estilometria e forense em textos. Mais detalhes podem ser consultados em `https://pan.webis.de`.

[6]`https://github.com/Maluuba/nlg-eval`

Abordagem	[Emmery et al. 2018]		[Shetty et al. 2018]	
	Inglês	Português	Inglês	Português
F1-score adversário	0.39	0.41	0.38	0.42
METEOR	0.38	0.36	0.35	0.32
BLEU	0.50	0.48	0.43	0.41

Tabela 1. Resultados dos experimentos.

analisar cada texto ofuscado e gerar a probabilidade de o mesmo ter sido escrito pelo seu autor original. Por fim, nós executamos o *nlg-eval* para avaliar a diferença semântica entre os textos originais e os textos ofuscados.

Nós também executamos os mesmos experimentos com os conjuntos de dados em inglês que foram disponibilizados pelos autores para ter uma base de comparação.

4.3. Resultados

A Tabela 1 apresenta os valores obtidos nos nossos experimentos. O modelo proposto por Emmery et al. [Emmery et al. 2018] obteve performance similar no F1-score adversário que o modelo proposto por Shetty et al. [Shetty et al. 2018], o que sugere que os dois possuem a mesma eficiência ao esconder os atributos que podem identificar o autor do texto. Os valores de METEOR e BLEU de Emmery et al., no entanto, foram superiores, o que evidencia que este modelo é mais eficiente em preservar a semântica do texto original. Ambos os modelos obtiveram performance inferior nos textos em português quando comparados aos textos em inglês, o que deixa evidente que as diferenças do português influenciam diretamente na performance dos ofuscadores.

5. Trabalhos Relacionados

O PAN publica relatórios anualmente com os resultados das avaliações das ferramentas que são submetidas para a força tarefa de ofuscação de textos [Mihaylova et al. 2016, Stamatatos et al. 2018]. As ferramentas submetidas para o PAN até então foram baseadas em regras pré-definidas que são aplicadas para transformar os textos, e não houve nenhuma submissão de ferramentas baseadas em redes neurais. Além disso, o idioma Português não faz parte do escopo de avaliação do PAN.

[Potthast et al. 2016] propuseram um experimento para comparar a performance de diversos ofuscadores de texto. Assim como nos nossos experimentos, eles também utilizaram métricas para comparar a eficiência em esconder os atributos estilísticos e preservar a semântica dos textos gerados. Dentre os ofuscadores que foram avaliados, no entanto, não havia nenhum que fosse baseado em aprendizado de máquina e que fosse treinado para ofuscar textos em português.

6. Conclusão e Trabalhos Futuros

Nós implementamos e avaliamos dois métodos de ofuscação de textos que obtiveram as melhores performances relatadas por seus autores. Dos dois métodos, o que obteve melhor performance ao ofuscar os textos do nosso conjunto de dados foi o de [Emmery et al. 2018].

Como trabalhos futuros, como é de praxe em trabalhos de tradução de máquina, nós realizaremos experimentos com avaliação humana para análise qualitativa das

sentenças geradas pelos ofuscadores. Após analisar os resultados da avaliação humana, nós utilizaremos o modelo que obtiver a melhor performance para compor uma ferramenta de ofuscação de textos para uso do público geral. Por fim, nós investigaremos se criptossistemas como SMC (*Secure Multi-party Computation*) e criptografia homomórfica podem ser empregados para executar os algoritmos de ofuscação de textos em um ambiente de computação em nuvem sem enviar o texto não-ofuscado para os servidores.

Referências

Bagnall, D. (2015). Author identification using multi-headed recurrent neural networks. *arXiv preprint arXiv:1506.04891*.

Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments.

Emmery, C., Manjavacas, E., and Chrupała, G. (2018). Style Obfuscation by Invariance. In *COLING 2018*, pages 984–996.

Ganin, Y. and Lempitsky, V. (2014). Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Mihaylova, T., Karadjov, G., Kiprov, Y., Georgiev, G., Koychev, I., and Nakov, P. (2016). SU@ PAN'2016: Author Obfuscation. In *CLEF (Working Notes)*, pages 956–969.

Narayanan, A., Paskov, H., Gong, N. Z., Bethencourt, J., Stefanov, E., Shin, E. C. R., and Song, D. (2012). On the feasibility of internet-scale author identification. In *2012 IEEE Symposium on Security and Privacy*, pages 300–314. IEEE.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318.

Potthast, M., Hagen, M., and Stein, B. (2016). Author obfuscation: Attacking the state of the art in authorship verification. In *CLEF (Working Notes)*, pages 716–749.

Shetty, R., Schiele, B., and Fritz, M. (2018). A4NT: Author Attribute Anonymity by Adversarial Training of Neural Machine Translation. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1633–1650, Baltimore, MD. USENIX Association.

Stamatatos, E., Rangel-Pardo, F. M., Tschuggnall, M., Stein, B., Kestemont, M., Rosso, P., and Potthast, M. (2018). Overview of PAN 2018. Author identification, author profiling, and author obfuscation. *Lecture Notes in Computer Science*, 11018:267–285.

Varela, P., Justino, E., and Oliveira, L. S. (2011). Selecting syntactic attributes for authorship attribution. In *IJCNN*, pages 167–172. IEEE.