



MODELANDO TRAÇOS DE PERSONALIDADE A PARTIR DE ROTEIROS DA SÉRIE FRIENDS UTILIZANDO REDES NEURAIS PROFUNDAS

MONIELE KUNRATH SANTOS¹; LARISSA A. DE FREITAS²

¹Universidade Federal de Pelotas – mksantos@inf.ufpel.edu.br

²Universidade Federal de Pelotas – larissa@inf.ufpel.edu.br

1. INTRODUÇÃO

Com o surgimento de vários modelos de *Machine Learning*, supervisionado e não-supervisionado, consegue-se obter uma análise profunda de grande volume de dados, principalmente de textos provindos da *Web*. Análises estas, que tentam entender o comportamento humano através do Processamento de Língua Natural, como por exemplo nas tarefas de Análise de Sentimento (AGARWAL et al, 2011) e Detecção de Sarcasmo/Ironia (JOSHI et al, 2016). Porém, essas pesquisas vão além do discurso formal e de pessoas reais. Esse tipo de abordagem é muito interessante pois os textos estão bem formados, isto é, sem abreviações e erros de ortografia, o que pode tornar a tarefa de uma rede neural, de entender palavras e o seu contexto, muito mais fácil.

Entretanto, apesar dos inúmeros projetos que abordam arquiteturas e modelos para realizar a tarefa de classificação de traços de personalidade, nota-se que há poucos estudos voltados a utilização de Redes Neurais Profundas (RNP) para classificação de traços de personalidade com personagens fictícios e nenhum que utilize redes pré-treinadas. O objetivo principal desse trabalho é analisar os resultados obtidos nos modelos RNPs, com e sem pré-treino, na tarefa de classificação de traços de personalidade dos principais personagens da famosa série dos anos 90, *Friends*. O estudo será focado na comparação do desempenho de um modelo com pré-treinamento e um modelo sem pré-treinamento.

2. METODOLOGIA

Há um grande interesse científico em práticas e técnicas que consigam entender discurso a partir de textos da *Web*. Entretanto, como mencionado na Seção 1, ainda há lacunas de métodos e estratégias que podem ser aprofundados. Em MAJUMDER et al. (2017), utilizam CNNs para fazer detecção de traços de personalidades *Big Five* (DIGMAN, 1990), cinco fatores de personalidade (Abertura à experiência, Conscienciosidade, Extroversão, Amabilidade e Neuroticismo) em textos. CNN é uma arquitetura bem complexa e torna o custo computacional alto, porém apresenta resultados promissores. A acurácia na melhor classe classificada (OPN) é de 63%. No trabalho PIZZOLLI et al. (2019), os autores identificam os traços de personalidade do *Big Five*, também denominada *Big Five*, de personagens fictícios a partir de peças de teatro do autor William Shakespeare. E seu melhor resultado foi na classe OPN, com F1-Score de 0.66, utilizando *Random Forest* como classificador.

Há também Hernandez et al. (2020), que identifica os traços de personalidade da Tipologia de *Myers-Briggs* em textos, e é o único trabalho que faz uso de Redes Neurais Recorrentes. O autor usa uma arquitetura LSTM e possui acurácia de 67% na classe melhor classificada.

¹<https://convokit.cornell.edu/>

Levando em consideração a motivação e a análise dos trabalhos mencionados nas Seções anteriores, pode-se projetar um modelo que supra as carências da literatura. A Figura 1 apresenta as etapas do desenvolvimento deste trabalho.

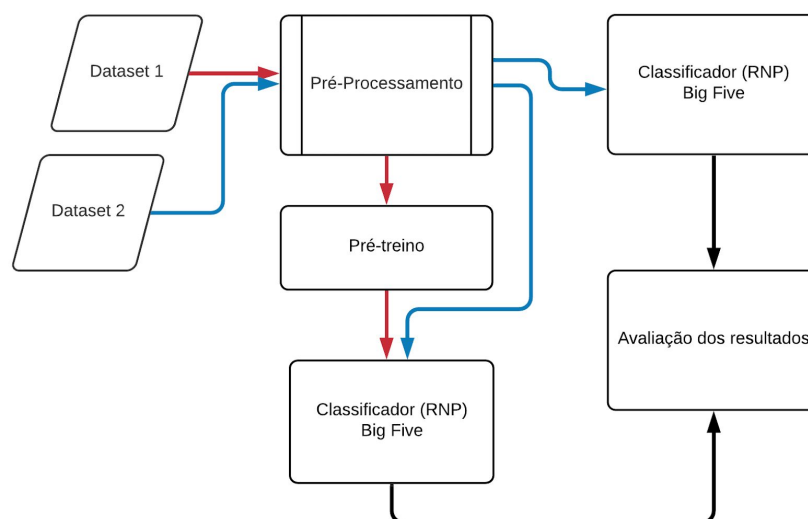


Figura 1 - Fluxograma do modelo proposto

2.1 DATASET

O *Dataset 1* contém todas as falas de todos os episódios da série, encontra-se disponível em ConvoKit¹. Ele possui 67373 linhas de falas e 3107 conversas. Já, o *Dataset 2*, contém conversas multiatores, com respectivos personagens e traços de personalidade (*Big Five*) do personagem em foco na conversa. Ele possui 711 conversas curtas das 4 primeiras temporadas da série (contendo 2 ou mais personagens), que foram extraídas e anotadas.

2.2 PRÉ-TREINO

Nesta etapa, utilizamos o *Dataset 1* pré-processado como entrada do classificador, para trabalhar como algum tipo de "contextualizador". Isto é, terá como objetivo principal, ensinar a rede neural sobre o vocabulário utilizado nos roteiros da série, bem como quem os fala.

2.3 CLASSIFICADOR DE TRAÇOS DE PERSONALIDADE

Depois de extraídas as *features* na etapa anterior, testamos o desempenho dos classificadores com pré-treinamento e sem pré-treinamento. Utilizamos neste trabalho os classificadores *Long Short-Term Memory* (LSTM) (HOCHREITER, 1997) e *Bidirectional Encoder Representations from Transformers* (BERT) (SUN, 2019).

Em um primeiro momento, usamos/testamos uma LSTM *Vanilla* a nível de caractere. A complexidade desse algoritmo é simples, entretanto não é possível salvar o contexto das palavras e as próprias palavras, já que o grau de aprendizado é baseado em letras. A configuração testada utilizou: 4 camadas, 2 camadas escondidas de tamanho 512 e otimizador Adam com *learning rate* de 0,0005.

¹<https://convokit.cornell.edu/>

Entretanto utilizamos um modelo de RNP que recentemente tem ganho destaque em trabalhos de classificação de texto é chamado de BERT e foi um desenvolvido pela *Google*. Testamos seu desempenho com suas configurações padrões *base-line*: 12 camadas, 768 camadas escondidas, 12 cabeças, 110M parâmetros, tamanho máximo do texto exemplo como 200 caracteres e otimizador Adam com *learning rate* de 1e-05.

3. RESULTADOS E DISCUSSÃO

O melhor desempenho foi do classificador BERT com acurácia geral de 55%. A partir desse resultado, mostramos a comparação da métrica *F1-Score* de cada classe do *Big Five*, ou seja, de cada traço de personalidade, com o classificador BERT (Tabela 1).

	NEU	EXT	AGR	CON	OPN
Precisão	46%	53%	48%	40%	67%
<i>Recall</i>	37%	27%	96%	29%	99%
F1-Score	41%	36%	65%	34%	80%

Tabela 1 - F1-Score por classe no Modelo Proposto com classificador BERT.

Há um aumento considerável quando pré-treinamos LSTM Vanilla, Como apresenta a tabela 2. A hipótese do porquê deve-se ao motivo da carência de vocabulário. Pois criamos essa rede neural a nível de caractere e sem a presença de nenhum tipo de biblioteca ou *wrapper*. Então pode-se concluir que o pré-treino faz muita diferença nesse tipo de modelo. Ao contrário de modelos a nível de sentença como BERT, uma vez que são instanciados de bibliotecas, já possuem modelos pré-treinados em alguns idiomas, entendendo inclusive vocabulários.

Classificador	Com pré-treinamento	Sem pré-treinamento
LSTM Vanilla	42%	16%
BERT	55%	50%

Tabela 2 - Acurácia dos classificadores.

Outro possível motivo, pode ser devido a quantidade de dados inseridos no pré-treinamento do LSTM Vanilla em comparação com o BERT. Como não tivemos acesso para utilizar TPUs dos *Transformers*, o modelo BERT só conseguiu pré-treinar (em tempo hábil) 3.000 amostras do *Dataset* 1 (ao todo aproximadamente 60.000 linhas), que fora completamente usado na etapa de pré-treino da LSTM.

Ainda, os melhores valores de *recall* foram das classes Agradabilidade (AGR) e Abertura para a experiência (OPN) e o pior valor foi da classe Extroversão (EXT), apresentados na Tabela 1. Isso deve-se ao desbalanceamento de classes do *Dataset* 2, que possui mais exemplos de AGR e OPN. Entretanto, apesar de terem um bom *recall*, sua precisão não corresponde na mesma intensidade. Portanto, é importante, nesse caso, analisar a métrica F1-Score, além da acurácia, uma vez que o *Dataset* 2 não é totalmente balanceado.

4. CONCLUSÕES

Este trabalho discutiu a proposta de um modelo que usa RNPs com etapa de pré-treino para classificar traços de personalidade a partir de roteiros do seriado *Friends*. E essa estratégia se mostra muito promissora, visto que os resultados demonstram uma significativa melhora, quando pré-treinados. Em comparação com os trabalhos vistos na literatura, nossa abordagem demonstra resultados levemente superiores, alcançando 55% de acurácia geral e 80% *F1-Score*, na classe OPN, com o classificador BERT. Por outro lado, devemos citar que o tamanho do dataset é certamente menor do que o recomendado em problemas que utilizam RNPs como classificadores. Tendo isso em vista, a partir da eficácia do nosso modelo, podemos replicá-lo para outros trabalhos de classificação de traços de personalidade. No futuro esperamos conseguir testar outros modelos de *Transformers*.

5. REFERÊNCIAS BIBLIOGRÁFICAS

PIZZOLLI, D.; STRAPPARAVA, C. Personality traits recognition in literary texts. **Proceedings of the Second Workshop on Storytelling**, p.107–111, 2019.

AGARWAL, A.; XIE, B.; VOVSHA, I. Sentiment analysis of twitter data. **Proceedings of the workshop on language in social media (LSM 2011)**, p. 30–38, 2011.

JOSHI, A.; TRIPATHI, V. ; BHATTACHARYYA, P. Harnessing sequence labeling for sarcasm detection in dialogue from tv series ‘friends’. **Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning**, p. 146–155, 2016.

MAJUMDER, N.; PORIA, S. ; GELBUKH, A. Deep learning-based document modeling for personality detection from text. **IEEE Intelligent Systems**, v. 32, n. 2, p. 74–79, 2017.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, v. 9, n. 8, p. 1735–1780, 1997.

SUN, C.; QIU, X; XU Y. How to fine-tune bert for text classification?. **China National Conference on Chinese Computational Linguistics**. Springer, p. 194–206, 2019.

JIANG, H.; ZHANG, X. ; CHOI, J. D. Automatic text-based personality recognition on monologues and multiparty dialogues using attentive networks and contextual embeddings. **Proceedings of the AAAI Conference on Artificial Intelligence**, v. 34, n. 10, p. 13821–13822, 2020.

DIGMAN, J. M. Personality structure: Emergence of the five-factor model. **Annual review of psychology**, v. 41, n. 1, p. 417–440, 1990.

Hernandez, R.; Knight, I. S. Predicting myers-briggs type indicator with text. **Multimodal Technologies and Interaction**, v. 4, 2020.

¹<https://convokit.cornell.edu/>