

# Classificação de textos em português utilizando diferentes técnicas de embedding do Word2vec ao GPT

Thomas Edson Cordeiro dos Santos  
Programa de Pós-Graduação em Informática - PPGI  
Universidade Federal do Espírito Santo - UFES  
Vitória, Brasil  
thomas.santos@ufes.edu.br

Wharley Borges Ferreira  
Programa de Pós-Graduação em Informática - PPGI  
Universidade Federal do Espírito Santo - UFES  
Vitória, Brasil  
wharley.ferreira@edu.ufes.br

**Abstract**—Neste trabalho nós comparamos diversos modelos de *embedding* de texto em português brasileiro. A qualidade dos resultados foi avaliada na tarefa de classificação da variável binária conformidade de textos das ordens de manutenção do SAP da indústria de óleo e gás. O modelo que demonstrou melhores resultados foi um Word2vec (Petrovec-O&G de 100 dimensões) treinado com mais de 85 milhões de tokens de documentos públicos desta indústria. Este modelo obteve um recall de 78 %, superando até mesmo modelos baseados na arquitetura transformers, que tem apresentado de maneira consistente resultados no estado da arte em tarefas de processamento de linguagem natural.

**Index Terms**—português, *embedding*, processamento de linguagem natural, classificação de textos

## I. INTRODUÇÃO

O campo de processamento de linguagem natural (do inglês *Natural Language Processing* - NLP) é uma área da inteligência artificial que tem evoluído rapidamente nos últimos anos. O ChatGPT da OpenAI [1] [2] [3] [4] ganhou as manchetes por apresentar um modelo que consegue "conversar" com o usuário com uma facilidade e fluência nunca vistos antes. Grandes modelos de linguagem (do inglês *Large Language Models* - LLM) proprietários apresentam sistematicamente resultados do estado da arte, porém são muito pesados para rodar localmente, sendo disponibilizados por meio de chamadas de API, que possuem custo associado, depende da confiabilidade do serviço e em alguns casos os dados podem ser usados para treinamento de versões posteriores do modelo, levantando questões sobre sigilo e ética. Outra limitação atual dos modelos baseados em transformers é o tamanho fixo da entrada, o que inviabiliza alguns tipos de análise. O presente trabalho apresenta a avaliação da aplicabilidade de tecnologias para classificação de textos em português em uma base de dados de ordem de manutenção no sistema SAP. Atualmente essa análise é realizada manualmente por diversas pessoas, com uma cobertura em parte das ordens que são geradas a cada mês. A proposta é que esta análise seja realizada automaticamente em todas as ordens e, apenas as ordens mais prováveis de oportunidades de melhorias, sejam avaliadas posteriormente por humanos. Nesta abordagem, seria realizada

uma cobertura maior das ordens e o uso do trabalho humano seria otimizado. Essas ordens contêm informações possivelmente sensíveis, dessa forma, os dados devem ser tratados em ambientes seguros. Por esse motivo, os dados só foram tratados através de API privada ou modelos rodando localmente. Um dos objetivos da presente revisão é avaliar se a tecnologia por trás do ChatGPT ou as usadas por seus concorrentes treinados especificamente em português do Brasil [5] [6] podem ser aplicadas para classificar com eficiência textos escritos nesse idioma. Foi realizada uma revisão sistemática sobre a classificação de textos em português para verificar quais as técnicas mais promissoras. Ao final do trabalho são apresentados os desempenhos obtidos com diferentes técnicas e é observado que técnicas simples, abertas e consolidadas geram resultados competitivos com as tecnologias mais novas e proprietárias. Na seção II serão apresentadas as técnicas de *embedding*, na III são apresentados os resultados obtidos com essas técnicas e, por fim, na IV são apresentadas as conclusões.

## II. FERRAMENTAS DE EMBEDDING DE TEXTO UTILIZADAS

Para realizar a classificação de textos com uso de modelos de inteligência artificial, é necessário primeiro transformar esses textos em números. Essa tarefa é chamada de *embedding* e, ao contrário do que pode parecer, não é trivial. A abordagem mais simples para essa conversão é atribuir a cada possível palavra uma dimensão (ou coluna em uma matriz), porém essa abordagem tipicamente gera matrizes esparsas de alta dimensionalidade. Essa abordagem gera inconvenientes chamados de "maldição da dimensionalidade" (do inglês *curse of dimensionality*). Um trabalho de 2003 [7] introduziu a ideia de aprender uma representação distribuída das palavras que consegue, não só representar de forma compacta e eficiente as entidades, mas também as relações entre esses conceitos. Nessa abordagem, as palavras e sentenças devem ser distribuídas em um espaço multidimensional de forma que sentenças similares possuam localizações similares. Por mais simples que possa parecer, essa modelagem se mostrou extremamente poderosa e permite a construção de um "modelo de mundo" de tokens, que é uma das pedras fundamentais da

revolução que o campo de processamento de linguagem natural tem passado nos últimos anos. Foi realizado um mapeamento sistemático com vistas a localizar ferramentas de *embedding* utilizadas em trabalhos científicos. Um requisito essencial para a seleção das publicações foi buscar as voltadas ao idioma português, preferencialmente o Brasileiro. Considerando a grande revolução que houve nos últimos anos na área de NLP devido ao advento da arquitetura de transformers [1], foram pesquisados artigos publicados a partir de 2018.

#### A. TF-IDF

De forma a evitar um viés pelo uso exclusivo de técnicas novas, foi utilizada como referência a codificação de textos por meio de *Term Frequency-Inverse Document Frequency* (TF-IDF), após o que foi realizada a redução de dimensionalidade para 1000 por meio de SVD no que é conhecido como análise semântica latente (do inglês *latent semantic analysis* - LSA). O pré-processamento de texto consistiu em: conversão para minúsculo, eliminação de linhas com texto em branco, tokenização, remoção de stopwords, stemming após o qual foi criada uma matriz do tipo TF-IDF e realizada uma redução de dimensionalidade com SVD (do inglês *Singular Value Decomposition*).

#### B. Petrolês

Um dos trabalhos identificados através da revisão da literatura [6] corresponde ao resultado de uma colaboração interinstitucional liderada pelo Centro de Pesquisas e Desenvolvimento da Petrobras (CENPES), em parceria com PUC-Rio, UFRGS e PUC-RS, e visa incentivar pesquisas nas áreas de Processamento de Linguagem Natural e Linguística Computacional aplicadas ao domínio de Óleo e Gás (O&G). O Petrolês (<https://petroles.puc-rio.ai/>) é um repositório de artefatos de Processamento de Linguagem Natural especializados no domínio de petróleo em Português, e tem como objetivo servir como uma referência para os grupos de pesquisas em inteligência artificial e empresas atuantes nesse domínio. Dentre os artefatos disponíveis livremente no repositório foram utilizados no presente trabalho três modelos de word embedding do tipo Word2vec sendo dois deles, Petrovec-O&G de 100 e 300 dimensões, treinados unicamente a partir de dados públicos relacionados ao domínio de O&G (Boletins Técnicos da Petrobras; Teses e Dissertações em assuntos relacionados à indústria de Petróleo; Notas e estudos técnicos da ANP). O terceiro modelo Petrovec-híbrido de 100 dimensões treinado nos mesmos documentos que os modelos O&G e adicionalmente em um corpus de contexto geral em Português do NILC.

#### C. BERTimbau

Uma ferramenta citada recorrentemente nas publicações foi o BERT (do inglês *Bidirectional Encoder Representations for Transformers*) da Google, em particular uma versão que passou por um *fine-tuning em português*, o BERTimbau [5]. Esse modelo foi obtido através da biblioteca transformers do Hugging Face distribuída como padrão no pacote ANACONDA.

O nome formal do modelo na biblioteca é o "neuralmind/bert-base-portuguese-cased". Como saída, o BERT gera um tensor contendo um vetor de 768 dimensões para cada token. De forma a obter um *embedding* para toda a sentença, foi realizada a média entre todos os os vetores do tensor, sendo gerada dessa forma apenas um vetor de 768 por sentença.

#### D. SBERT

Também foi utilizada uma versão modificada do BERT específica para o *embedding* de sentenças, o SBERT [8] com uma versão treinada 15 idiomas e outra treinada em mais de 50 idiomas [9]. Como saída esses modelos geram apenas um vetor de 768 dimensões para cada sentença. Os modelos específicos utilizados nas simulações foram o "distiluse-base-multilingual-cased-v1" e o "distiluse-base-multilingual-cased-v2" respectivamente, disponíveis na biblioteca sentence\_transformers.

#### E. GPT

Uma ferramenta de processamento de linguagem natural que ganhou muita atenção recentemente foi o ChatGPT da OpenAI. Em conjunto com os chatbots, a OpenAI disponibiliza as *engines* por trás desses serviços através de APIs. A Petrobras disponibilizou, em parceria com a Microsoft um serviço com algumas das mais poderosas e recentes ferramentas da OpenAI. No caso do trabalho em questão, não foi cogitado usar a API de chat devido ao limite do tamanho de entrada. O conjunto de dados possui milhares de textos com dezenas de milhares de tokens. Foi utilizada, no entanto uma API de segunda geração, que de acordo com a própria OpenAI supera as APIs da primeira geração na maioria das tarefas. No presente trabalho foi utilizada especificamente uma versão interna da engine "text-embedding-ada-002". O serviço recebe uma sentença de até 8191 tokens e devolve um vetor com 1536 dimensões. Existe um trabalho bastante detalhado descrevendo o funcionamento da primeira geração de APIs [10], porém a documentação relativa à segunda geração de embeddings é mais escassa, o que reflete uma tendência recente dos produtos oferecidos pela OpenAI desde os grandes investimentos recebidos da Microsoft.

### III. RESULTADOS

Foram realizadas simulações com uma amostra de cerca de 2000 ordens. Considerando que a base de dados é desbalanceada (muito mais classe 0 que classe 1), também foram realizadas simulações com data augmentation por meio de SMOTE. A partir dos dados processados para cada técnica de embedding, combinada ou não com SMOTE foram realizadas classificações com três classificadores diferentes, um do tipo Random Forest (RF), Redes Neurais (NN) e XGBoost. As principais métricas para cada uma dessas simulações se encontram listadas na tabela I

Uma vez que objetivo do trabalho é automatizar uma auditoria de textos, é desejável identificar como da classe 1 uma boa parcela dos textos que pertencem à classe 1. Uma boa métrica para essa função é o recall, definido como:

TABLE I  
PRINCIPAIS RESULTADOS DAS SIMULAÇÕES COM DIFERENTES MÉTODOS DE EMBEDDING E DIFERENTES CLASSIFICADORES

Embedding	SMOTE	Modelo	TN	FP	FN	TP	precision	recall	F1	F1 accuracy	AUC_ROC
TF-IDF	N	RF	349	10	23	17	0,63	0,42	0,51	0,92	0,809
	S	RF	331	28	18	22	0,44	0,55	0,49	0,88	0,799
GPT	N	RF	353	6	31	9	0,60	0,23	0,33	0,91	0,789
		NN	355	4	32	8	0,67	0,20	0,31	0,91	0,772
		XGBoost	355	9	28	12	0,57	0,30	0,39	0,91	0,765
	S	RF	348	11	23	17	0,61	0,42	0,50	0,91	0,794
		NN	340	19	21	19	0,50	0,47	0,49	0,90	0,769
		XGBoost	340	19	21	19	0,50	0,47	0,49	0,90	0,789
Bertimbau	N	RF	354	5	33	7	0,58	0,17	0,27	0,90	0,735
		NN	355	4	33	7	0,64	0,17	0,27	0,91	0,733
		XGBoost	354	5	30	10	0,67	0,25	0,36	0,91	0,743
	S	RF	348	11	28	12	0,52	0,30	0,38	0,90	0,741
		NN	272	87	17	23	0,21	0,57	0,31	0,74	0,717
		XGBoost	341	18	27	13	0,42	0,33	0,37	0,89	0,716
Petrolês H 100	N	RF	345	14	23	17	0,55	0,42	0,48	0,91	0,816
		NN	352	7	24	16	0,70	0,40	0,51	0,92	0,823
		XGBoost	343	16	23	17	0,52	0,42	0,47	0,90	0,816
	S	RF	322	37	14	26	0,41	0,65	0,50	0,87	0,790
		NN	300	59	11	29	0,33	0,72	0,45	0,82	0,813
		XGBoost	325	34	16	24	0,41	0,60	0,49	0,87	0,779
Petrolês O&G 100	N	RF	346	13	25	15	0,54	0,38	0,44	0,90	0,811
		NN	351	8	26	14	0,64	0,35	0,45	0,91	0,793
		XGBoost	343	16	23	17	0,52	0,42	0,47	0,90	0,806
	S	RF	332	27	17	23	0,46	0,57	0,51	0,89	0,817
		NN	265	94	9	31	0,25	<b>0,78</b>	0,38	0,74	0,796
		XGBoost	327	32	17	23	0,42	0,57	0,48	0,88	0,806
Petrolês O&G 300	N	RF	344	15	25	15	0,50	0,38	0,43	0,90	<b>0,839</b>
		NN	352	6	26	14	0,70	0,35	0,47	0,92	0,815
		XGBoost	342	17	23	17	0,50	0,42	0,46	0,90	0,805
	S	RF	331	28	15	25	0,47	0,62	0,54	0,89	0,806
		NN	267	92	12	28	0,23	0,70	0,35	0,74	0,777
		XGBoost	323	36	19	21	0,37	0,53	0,43	0,86	0,800
SBERT v1	N	RF	352	7	31	9	0,56	0,23	0,32	0,90	0,794
		NN	354	5	30	10	0,67	0,25	0,36	0,91	0,798
		XGBoost	347	12	27	13	0,52	0,33	0,40	0,90	0,782
	S	RF	343	16	24	16	0,50	0,40	0,44	0,90	0,788
		NN	331	28	20	20	0,42	0,50	0,45	0,88	0,817
		XGBoost	340	19	23	17	0,47	0,42	0,45	0,89	0,805
SBERT v2	N	RF	353	6	30	10	0,62	0,25	0,36	0,91	0,770
		NN	354	5	29	11	0,69	0,28	0,39	0,91	0,822
		XGBoost	351	8	27	13	0,62	0,33	0,43	0,91	0,789
	S	RF	345	14	25	15	0,52	0,38	0,43	0,90	0,813
		NN	312	47	16	24	0,34	0,60	0,43	0,84	0,802
		XGBoost	342	17	22	18	0,51	0,45	0,48	0,90	0,785

$$recall = \frac{TP}{(TP + FN)}$$

É possível observar que são obtidos os melhores resultados de recall com o uso dos *embeddings* gerados pelo Petrolês O&G H 100. Outra métrica que permite comparar o desempenho de diferentes modelos é a curva ROC. Os gráficos para esse método utilizando ou não SMOTE estão contidos na Figura 1 e na Figura 2.

Os resultados obtidos mostram que é possível obter desempenho competitivo com técnicas consolidadas, simples e que rodam localmente, eliminando os custos e a dependência de um serviço fornecido através de uma API [10].

É possível observar dos resultados que ambos os modelos de *embedding* aplicados conseguem se distanciar bastante de um classificador aleatório (linha tracejada preta) porém ainda estão distantes de um classificador perfeito. Ordens de manutenção são objetos que possuem, além do texto longo associado, diversas outras características relacionais (data de criação, data

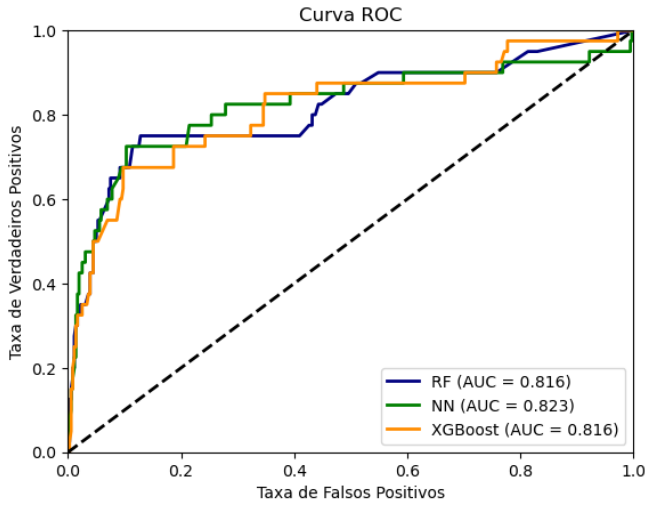


Fig. 1. Curva ROC de classificadores treinados com dados obtidos dos textos longos através de do petrolês O&G 100

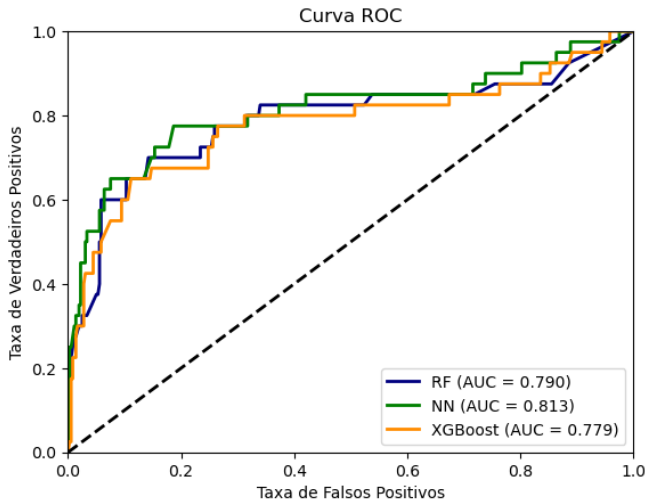


Fig. 2. Curva ROC de classificadores treinados com dados obtidos dos textos longos através de do petrolês O&G 100 e com data augmentation usando SMOTE

de encerramento, status, tipo, centro, entre outras). É provável que o uso da classificação por texto longo em conjunto com a classificação por essas características relacionais gere resultados ainda melhores.

#### IV. CONCLUSÃO

Foram encontradas diversas referências voltadas para a classificação automática de textos em português e os grandes modelos de linguagem possuem em suas bases de treinamento muitos conteúdos em português e, graças a isso, "falam" o idioma brasileiro com bastante fluência. Foi utilizada como referência de desempenho uma técnica tradicional de processamento de linguagem natural com a criação de uma matriz TF-IDF e posterior redução de dimensionalidade com SVD e treino de classificadores. Posteriormente foi realizado *em-*

TABLE II  
TABELA COMPARATIVA DE DIFERENTES TÉCNICAS DE CODIFICAÇÃO

Técnica	Dimensões	Camadas	Parâmetros
TF-IDF	1000	-	-
GPT	1536	-	-
BERTimbau	768	12	110 M
Petrolês 100	100	-	-
Petrolês 300	300	-	-
SBERT	512	-	-

*bedding* de sentenças com uma API da OpenAI e treino de um classificador similar ao do método anterior. Também foi realizado o *embedding* com o SBERT e com o Petrolês. Finalmente foi realizado o *embedding* com o modelo BERTimbau [5]. Os *embeddings* gerados pelo Petrolês apresentaram resultados competitivos com as vantagens de não depender do uso de uma API (cujo custo pode mudar, pode ser descontinuada, ou pode simplesmente estar indisponível devido à elevada demanda) e de não levar várias horas para rodar localmente (como foi observado com o BERTimbau). Considerando os fatores citados anteriormente, para a aplicação específica de classificação de textos longos de ordens de manutenção, é indicado o uso de um dos modelos treinados disponíveis no repositório do Petrolês. Trabalhos futuros podem avaliar o desempenho dessas técnicas aplicadas a um subconjunto diferente de ordens, assim como explorar melhor os hiperparâmetros dos classificadores.

#### V. REPOSITÓRIO

Códigos utilizados para a obtenção dos resultados se encontram disponíveis no seguinte repositório público do GitHub: <https://github.com/ThomasAndarilho/AutoSAO/>

#### REFERENCES

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.
- [2] A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training," in *Computer Science, Linguistics*, 2018. [Online]. Available: <https://api.semanticscholar.org/CorpusID:49313245>
- [3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019. [Online]. Available: <https://www.semanticscholar.org/paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-Wu/9405cc0d616998371b2755e573cc28650d14dfe>
- [4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.
- [5] F. Souza, R. Nogueira, and R. Lotufo, "BERTimbau: pretrained BERT models for Brazilian Portuguese," in *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*, 2020.
- [6] D. d. S. M. Gomes, F. C. Cordeiro, B. S. Consoli, N. L. Santos, V. P. Moreira, R. Vieira, S. Moraes, and A. G. Evsukoff, "Portuguese word embeddings for the oil and gas industry: Development and evaluation," *Computers in Industry*, vol. 124, 2021, cited by: 12.

- [7] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," in *Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich, and V. Tresp, Eds., vol. 13. MIT Press, 2000.
- [8] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. [Online]. Available: <https://arxiv.org/abs/1908.10084>
- [9] —, "Making monolingual sentence embeddings multilingual using knowledge distillation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2020. [Online]. Available: <https://arxiv.org/abs/2004.09813>
- [10] A. Neelakantan, T. Xu, R. Puri, A. Radford, J. M. Han, J. Tworek, Q. Yuan, N. Tezak, J. W. Kim, C. Hallacy, J. Heidecke, P. Shyam, B. Power, T. E. Niekoul, G. Sastry, G. Krueger, D. Schnurr, F. P. Such, K. Hsu, M. Thompson, T. Khan, T. Sherbakov, J. Jang, P. Welinder, and L. Weng, "Text and code embeddings by contrastive pre-training," 2022.