Comparação de Modelos de Redes Neurais Recorrentes na Análise de Sentimentos e Emoções

***Resumo*— O grande volume de dados textuais postados diariamente em plataformas digitais, como o Twitter, constitui uma fonte valiosa para a análise de opiniões e a compreensão de estados emocionais coletivos. Este projeto compara a aplicação de quatro modelos de Redes Neurais Recorrentes — *SimpleRNN*, *Long Short-Term Memory* (LSTM), *Gated Recurrent Unit* (GRU) e *Bidirectional* LSTM (BLSTM) — nas tarefas de análise de sentimentos (positivo, negativo e neutro) e identificação de emoções em oito categorias: raiva, expectativa, desgosto, medo, alegria, otimismo, tristeza e surpresa. Os resultados indicam que o modelo BLSTM apresentou o melhor desempenho na tarefa de classificação de sentimentos, com acurácia de 73,24% e F1-scores equilibrados entre as três classes. Na classificação de emoções, as classes: alegria, medo e raiva tiveram os melhores resultados, com precisão superior a 0.40, possivelmente devido a padrões lexicais mais explícitos, enquanto emoções como surpresa, desgosto e tristeza apresentaram precisão inferior a 0.30. Os resultados são promissores para análise de sentimentos e emoções coletivas por meio de postagens em redes sociais, sendo uma metodologia aplicável para o auxílio à tomada de decisões estratégicas para a segurança e defesa nacional.**

***Palavras-Chave*— Redes Neurais Recorrentes, Análise de sentimentos, Twitter, Identificação de Emoções.**

I. Introdução

As redes sociais geram diariamente um enorme volume de dados contendo posicionamentos, discussões e opniões de uma grande parcela da população brasileira. A análise de sentimentos e emoções destas postagens fornecem informações estratégicas para compreensão de opiniões e estados coletivos [1]. Esta abordagem é tradicionalmente aplicada à marketing, mas adquire relevância em contextos de monitoramentos de defesa, segurança e desastres naturais. Estudos recentes, como o de Felbo et al. [2], demonstram que modelos de redes neurais treinados com dados de redes sociais conseguem identificar emoções com precisão comparável à de avaliadores humanos.

O presente estudo aplica e analisa técnicas de inteligência artificial e processamento de linguagem natural (NLP) para a extração de sentimentos a partir de grandes volumes de dados textuais. Utiliza-se, para isso, uma base de tweets rotulada com sentimentos e emoções relacionados à marca Dell, com o objetivo de identificar tanto a polaridade geral das postagens (positiva, negativa ou neutra) quanto a emoção predominante entre oito categorias: raiva, expectativa, desgosto, medo, alegria, otimismo, tristeza e surpresa. Para essa finalidade, foram comparados os desempenhos de quatro arquiteturas de redes neurais recorrentes — *SimpleRNN*, *Long Short-Term Memory* (LSTM), *Gated Recurrent Unit* (GRU) e *Bidirectional LSTM* (BLSTM) — nas tarefas de análise de sentimentos e classificação emocional [3], [4].

II. Metodologia

A. Base de Dados

Foi utilizada uma base de dados pública composta por 24.970 tweets relacionados à marca Dell, contendo anotações sobre três classes de sentimento: positivo, negativo e neutro, e oito emoções: *anger*, *anticipation*, *digust*, *fear*, *joy*, *optimism*, *sadness*, e *surprise* [5].

As principais variáveis consideradas no experimento foram:

- `text` — texto bruto do tweet, utilizado como entrada dos modelos;
- sentiment — rótulo de sentimento associado (positivo, negativo ou neutro);
- `emotion` — rótulo da emoção predominante no texto;
- `sentiment_score` e `emotion_score` — pontuação de confiança para cada classificação.

Outras variáveis disponíveis na base, como `datetime`, `username` e `Tweet_Id`, foram descartadas por não fornecerem informações relevantes à tarefa de classificação textual.

A base não apresenta valores nulos, dispensando tratamentos adicionais de imputação ou eliminação de amostras.

B. Balanceamento de Dados

A distribuição das três classes de sentimentos é apresentada na Figura 1, sendo a classe de sentimentos negativos majoritária e as duas outras classes com equivalencia no número de amostras.

O balanceamento dos dados para a tarefa de análise de sentimentos foi realizado reduzindo-se o número de amostras para 7000 para cada classe. As amostras restantes foram utilizadas no conjunto de teste dos modelos.

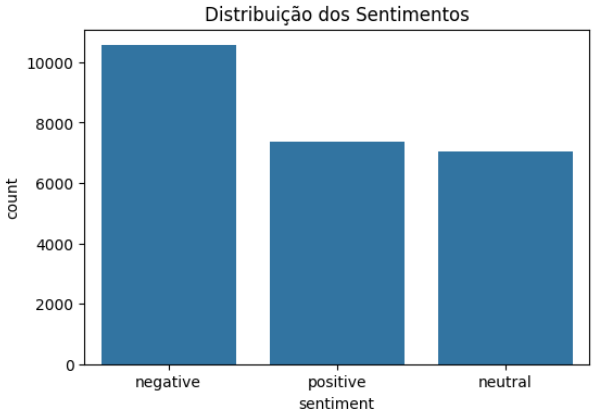

Fig. 1: Distribuição das amostras entre as três classes de sentimentos.

A distribuição das amostras nas classes de emoções, apresenta um grande desequilíbrio, conforme apresentado na Figura 2. As classes *anger*, *joy* e *anticipation* concentram a maior parte das amostras, enquanto emoções como *surprise*, *fear*, *optimism*, *sadness* e *disgust* possuem menor representatividade.

Desta forma, para a tarefa de detecção de emoções, adotou-se uma abordagem híbrida de balanceamento para mitigar

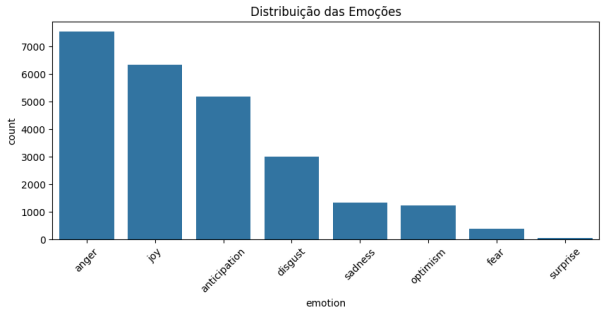

Fig. 2: Distribuição das amostras de emoções nas oito classes.

o viés provocado pela distribuição desigual das classes. As categorias com mais de 3500 amostras foram reduzidas a esse limite por meio de subamostragem aleatória, enquanto as classes minoritárias foram aumentadas até 3500 amostras, por meio da geração de exemplos sintéticos utilizando a técnica *Synthetic Minority Over-sampling Technique* (SMOTE) [6].

C. Pré-processamento dos Dados

O pré-processamento textual é uma etapa essencial para preparar os dados que alimenta os modelos de classificação, impactando diretamente na qualidade das representações e, consequentemente, no desempenho dos modelos [7]. Neste trabalho, empregaram-se as funções da biblioteca `NLTK` e ferramentas da API `Keras` do Python. As seguintes etapas foram efetuadas:

- Conversão de todos os textos para letras minúsculas, visando uniformizar o vocabulário e evitar duplicação de tokens semanticamente iguais com diferentes capitalizações;
- Remoção de menções (@usuários), hashtags, URLs e caracteres especiais, elementos geralmente considerados ruído em tarefas de classificação de texto [8];
- Eliminação de *stopwords* da língua inglesa, palavras que aparecem com alta frequência, mas carregam baixo valor semântico para o contexto do aprendizado supervisionado [9];
- Tokenização dos textos e transformação em sequências numéricas utilizando `Tokenizer(num_words=1000)`, com base nas palavras mais frequentes;
- Padronização do comprimento das sequências com `pad_sequences`, definindo `max_len = 50`, valor estimado a partir da análise da extensão média dos tweets em português;
- Balanceamento das amostras para as tarefas de classificação de sentimentos e emoções.

O parâmetro `num_words`, utilizado na função `Tokenizer` do Keras, define o número máximo de palavras consideradas no vocabulário, com base na sua frequência de ocorrência no corpus. A seleção desse valor foi fundamentada em uma análise da distribuição de frequência das palavras presentes nos tweets. Conforme ilustrado na Figura 3, as 500 palavras mais frequentes cobrem aproximadamente 55% do conteúdo textual, enquanto as 1000 mais comuns abrangem entre 60% e 70%. A partir de 2000 palavras, os ganhos em cobertura se tornam marginais, ao passo que o risco de *overfitting* aumenta devido à inclusão de termos pouco relevantes ou específicos demais [10].

Considerando que os tweets são textos curtos e limitados a 280 caracteres, o uso de um vocabulário com até 1000 palavras representa um bom compromisso entre representatividade e generalização. Além disso, definiu-se o valor de `max_len` com base na estrutura típica dos tweets em português. Em média, um tweet completo pode conter em média 50 palavras, sendo esse o valor escolhido para padronizar o comprimento das sequências textuais utilizadas como entrada nos modelos.

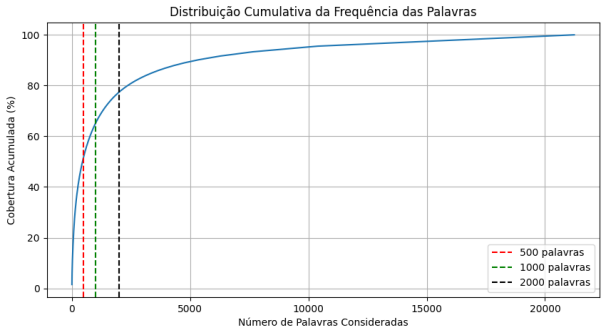

Fig. 3: Distribuição cumulativa da frequência de palavras.

D. Arquitetura e Treinamento dos Modelos

Os modelos recorrentes SimpleRnn, LSTM, BiLSTM e GRU são projetadas para capturar dependências temporais em sequências, sendo particularmente adequadas para dados textuais, onde a ordem das palavras é relevante [11], [12], [13].

O modelo SimpleRNN representa a forma mais básica de rede recorrente, com estrutura simples e baixo custo computacional. No entanto, é limitado pela dificuldade em aprender dependências de longo prazo devido ao problema do desaparecimento do gradiente.

A LSTM supera essa limitação ao introduzir portas (de entrada, esquecimento e saída) que controlam o fluxo de informações, permitindo a modelagem eficiente de dependências de longo alcance. Sua principal desvantagem é o maior custo computacional, decorrente da complexidade estrutural.

A GRU é uma alternativa mais recente à LSTM que simplifica sua arquitetura ao combinar algumas de suas portas. Isso resulta em um modelo mais leve, com desempenho comparável ao da LSTM em muitas tarefas, especialmente quando os dados de treinamento são mais escassos. Por outro lado, pode ser menos expressivo para sequências extremamente longas ou complexas.

Já a BiLSTM estende a LSTM ao processar a sequência tanto no sentido direto quanto reverso, permitindo que o modelo tenha acesso ao contexto passado e futuro simultaneamente. Essa bidirecionalidade aumenta significativamente o desempenho em tarefas como análise de sentimentos e tradução automática. Contudo, essa abordagem requer mais memória e tempo de processamento, o que pode ser uma limitação em sistemas com restrições de recursos.

Todos os modelos seguiram uma arquitetura unificada para ambas as tarefas (análise de sentimentos e emoções), estruturada em quatro camadas principais:

- Camada de *Embedding*: Vetores densos de dimensão 16 para representação lexical.

- Camada Recorrente: SimpleRNN, LSTM, GRU ou LSTM Bidirecional com 16 unidades, utilizando *dropout* e `recurrent_dropout` ambos de 0.4 para mitigar *overfitting* e capturar dependências temporais no texto.
- Regularização: Camada adicional de *Dropout* com taxa de 0.4, após a saída recorrente.
- Classificação: Camada `Dense` com função de ativação `softmax`, dimensionada conforme o número de classes da tarefa (3 para análise de sentimentos e 8 para identificação de emoções).

  A configuração hiperparamétrica (16 neurônios e taxa de *dropout* de 0.4) foi configurada empiricamente para equilibrar capacidade de generalização e viabilidade computacional, considerando restrições de hardware. Essa configuração busca manter a capacidade de generalização do modelo sem comprometer o tempo de treinamento ou a viabilidade computacional da abordagem.

A configuração para o treinamento dos modelos foi:

- Otimizador: `Adam`;
- Função de perda: `categorical_crossentropy`;
- Número de épocas: 15;
- `batch size`: 128;
- Divisão dos dados: 80% para treinamento e 20% para teste;
- Métrica de avaliação: `categorical_accuracy`.

III. Resultados

A. Classificação de Sentimentos

A Tabela I apresenta a acurácia dos quatro modelos recorrentes avaliados no conjunto de validação, considerando a média global de classificações corretas. Embora o conjunto de treinamento e validação tenham sido balanceados entre as três classes de sentimento (positivo, negativo e neutro), o conjunto de teste apresenta-se desbalanceado, com mais amostras para a classe 'negativo'.

Nesse cenário, o modelo BLSTM obteve o melhor desempenho, com acurácia média de 73,24%, seguido pelos modelos LSTM (72,02%) e GRU (72,00%), que apresentaram desempenhos bastante próximos. O SimpleRNN, por outro lado, apresentou o desempenho mais baixo, com 67,18%.

TABELA I: Acurácia média dos modelos RNN no conjunto de validação - classificação de três sentimentos: positivo, negativo e neutro.

Modelo	Acurácia (%)
SimpleRNN	67,18
LSTM	72,02
GRU	72,00
BLSTM	73,24

Para complementar a análise e avaliar o equilíbrio do desempenho entre as classes, a Tabela II apresenta as métricas de desempenho — precisão, *recall* e F1-score.

Observa-se que a classe negativa obteve os melhores resultados em todos os modelos, com destaque para a alta precisão nos modelos LSTM, GRU e BLSTM, superando 90%. O F1-score mais alto foi registrado pelo BLSTM (0,84), seguido por GRU (0,83).

A classe positiva também foi bem identificada pelos modelos LSTM, GRU e BLSTM, com F1-scores de cerca de 0,67, enquanto o SimpleRNN teve desempenho inferior (F1-score = 0,55).

TABELA II: Comparação das métricas por classe de sentimentos para os quatro modelos de RNN.

Classe	Métrica	SimpleRNN	LSTM	GRU	BLSTM
Positivo	Precisão	0.46	0.67	0.65	0.67
	recall	0.68	0.69	0.70	0.69
	F1-Score	0.55	0.68	0.67	0.68
Negativo	Precisão	0.77	0.91	0.91	0.90
	recall	0.85	0.75	0.76	0.78
	F1-Score	0.81	0.82	0.83	0.84
Neutro	Precisão	0.52	0.41	0.42	0.43
	recall	0.04	0.65	0.62	0.61
	F1-Score	0.07	0.50	0.50	0.51

A classe neutra apresentou o maior nível de dificuldade para a correta classificação pelo modelo, evidenciando limitações na sua distinção em relação às demais categorias. O SimpleRNN teve *recall* de apenas 0,04 e F1-score de 0,07. Os demais modelos apresentaram F1-score em torno de 0,50, indicando maior capacidade de generalização. Essa dificuldade pode ser atribuída à natureza ambígua da classe neutra, geralmente menos marcada lexicalmente.

De forma geral, os modelos LSTM, GRU e BLSTM se mostraram superiores ao SimpleRNN, com destaque para o BLSTM, que apresentou melhor F1-score nas três classes. Indicando que arquiteturas capazes de explorar relações temporais bidirecionais e dependências de longo prazo são solucionam melhor as tarefas complexas de análise de sentimentos.

B. Classificação de Emoções

Para a tarefa de classificação das oito emoções da base de dados também foram empregados os quatro modelos de redes neurais recorrente.

A Tabela III apresenta a acurácia dos modelos. Os modelos LSTM, GRU e BLSTM apresentam desempenho de cerca de 36%, enquanto que o modelo SimpleRNN, apresentou acurácia de somente 24,52%, refletindo sua limitação estrutural em capturar dependências temporais complexas em sequências textuais.

Dado que o número de classes é oito e o conjunto é balanceado, o desempenho aleatório esperado seria de aproximadamente 12,5%. Assim, mesmo os modelos mais simples foram capazes de aprenderam alguns padrões discriminativos. No entanto, os valores absolutos de acurácia ainda são relativamente baixos, o que aponta para a dificuldade da tarefa de capturar nuances emocionais que, muitas vezes, são sutis e subjetivas.

TABELA III: Acurácia média dos modelos RNN no conjunto de validação - classificação de oito emoções.

Modelo	Acurácia (%)
SimpleRNN	24,52
LSTM	36,80
GRU	36,55
BLSTM	36,57

A Tabela IV apresenta as métricas de precisão, *recall* e F1-score por classe emocional.

As emoções *joy* e *anger* apresentaram os melhores F1-score, atingindo 0,52 (BLSTM) e 0,50 (LSTM/GRU), respectivamente. Essas emoções têm expressões linguísticas mais marcadas e menos ambíguas. Além disto, ambas as classes possuiam mais de 3500 amostras, não tendo sido geradas amostras artificiais com o SMOTE.

As emoções *fear*, *optimism* e *surprise*, correspondiam as três classes minoritárias, tendo sido geradas amostras artificiais no processo de *oversampling*. Todavia, elas também mostraram desempenho interessante, com F1-score superior a 0.35, indicando que os modelos conseguiram captar parcialmente os padrões dessas emoções. Um desempenho equivalente também foi apresentado pela *anticipation*, cujas amostras eram todas reais.

As emoções *sadness* e *disgust* foram as mais difíceis de classificar, com F1-score inferior a 0,25. Isso pode se dever à sobreposição semântica com outras emoções (como medo ou raiva) ou à falta de expressões características em textos curtos.

O SimpleRNN apresentou os piores resultados em quase todas as classes, com F1-scores abaixo de 0,34 em todos os casos — e abaixo de 0,20 em várias emoções (ex.: tristeza, desgosto, expectativa). Isso confirma que sua simplicidade estrutural é inadequada para lidar com tarefas linguísticas multiclasse complexas.

LSTM, GRU e BLSTM tiveram desempenho semelhante, com diferenças pequenas mas consistentes em favor do LSTM. Isso confirma a expectativa da literatura, dado que estas arquiteturas são projetadas para lidar melhor com dependências temporais e gradientes longos, comuns em tarefas de classificação sequencial.

TABELA IV: Comparação das métricas de classificação por classe de emoções para os quatro modelos de redes neurais recorrentes.

Classe	**Métrica**	**SimpleRNN**	**LSTM**	**GRU**	**BLSTM**
raiva (*anger*)	Precisão	0.30	0.40	0.41	0.41
	Recall	0.37	0.65	0.64	0.64
	F1-Score	0.33	0.49	0.50	0.50
expectativa *(anticipation)*	Precisão	0.17	0.40	0.36	0.37
	Revocação	0.02	0.34	0.38	0.34
	F1-score	0.04	0.37	0.37	0.35
desgosto (*disgust*)	Precisão	0.21	0.23	0.26	0.23
	Recall	0.08	0.14	0.19	0.14
	F1-Score	0.11	0.18	0.22	0.18
medo (*fear*)	Precisão	0.31	0.43	0.45	0.47
	Recall	0.36	0.35	0.32	0.32
	F1-Score	0.33	0.39	0.38	0.38
alegria (*joy*)	Precisão	0.21	0.45	0.44	0.45
	Recall	0.83	0.58	0.59	0.62
	F1-Score	0.34	0.51	0.50	0.52
otimismo (*optimism*)	Precisão	0.29	0.40	0.34	0.40
	Recall	0.12	0.34	0.34	0.33
	F1-Score	0.17	0.37	0.34	0.36
tristeza (*sadness*)	Precisão	0.12	0.23	0.25	0.24
	Recall	0.00	0.08	0.13	0.13
	F1-Score	0.01	0.11	0.18	0.17
surpresa (*surprise*)	Precisão	0.26	0.29	0.30	0.27
	Recall	0.17	0.46	0.32	0.39
	F1-Score	0.20	0.35	0.31	0.32

IV. Conclusão

Este estudo investigou o desempenho de quatro arquiteturas de redes neurais recorrentes (SimpleRNN, LSTM, GRU e Bidirectional LSTM) aplicadas à classificação de sentimentos e emoções em tweets, com o objetivo de avaliar o impacto da escolha arquitetural sobre a performance em tarefas de processamento de linguagem natural com dados curtos e informais.

Na tarefa de análise de sentimentos, o modelo Bidirectional LSTM apresentou o melhor desempenho, alcançando uma acurácia de 73,24% e F1-scores mais equilibrados entre as três classes (positivo, negativo e neutro). Essa superioridade pode ser atribuída à sua capacidade de capturar dependências contextuais em ambas as direções da sequência textual. Por outro lado, o SimpleRNN obteve desempenho inferior, evidenciando suas limitações na modelagem de sequências mais complexas e na retenção de informação ao longo do tempo.

A tarefa de classificação de emoções revelou-se mais complexa, tanto pelo maior número de classes (oito categorias emocionais), quanto pela necessidade de identificar nuances linguísticas sutis em mensagens breves, informais e potencialmente ambíguas. Apesar do balanceamento das classes no conjunto de dados, os modelos LSTM, GRU e BLSTM alcançaram acurácias em torno de 36%, enquanto o SimpleRNN ficou restrito a 24,5%. Emoções como *joy* e *anger* foram reconhecidas com precisão superior a 40%, enquanto que as emoções *disgust*, *surprise* e *sadness* tiveram precisão abaixo de 30%. Esses resultados sugerem que a identificação de algumas emoções depende de mecanismos mais sofisticados de representação semântica, capazes de lidar com fenômenos como ironia, sarcasmo, contexto pragmático e ambiguidade lexical.

Os resultados indicam que a arquitetura LSTM foi a mais eficiente para a tarefa de detecção de emoções, conciliando desempenho competitivo e estabilidade nas métricas entre classes.

O desempenho geral obtido pelos modelos em ambos os problemas, aponta para limitações das redes recorrentes tradicionais em tarefas multiclasse com expressividade emocional sutil, sinalizando a necessidade de investigar abordagens mais avançadas, como modelos pré-treinados com atenção, como BERT (*Bidirectional Encoder Representations from Transformers*) ou híbridos com mecanismos de atenção e *embeddings* contextuais, como próximos passos de pesquisa.

Referências

[1] B. Liu, *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012.

[2] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm," *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1615–1625, 2017.

[3] A. Lakshmanarao, A. Srisaila, and T. S. R. Kiran, "Twitter sentiment classification with deep learning lstm for airline tweets," in *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, 2022, pp. 520–524.

[4] C. Pezoa, "Sentiment and emotion on twitter: The case of the global consumer electronics industry," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 18, pp. 765–776, 03 2023.

[5] A. Kumar, "Sentiment and emotions of tweets," Kaggle Dataset, 2023, accessed: [Insert Last Accessed Date]. [Online]. Available: https://www.kaggle.com/datasets/ankitkumar2635/sentiment-and-emotions-of-tweets/data

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[7] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, "A brief survey of text mining: Classification, clustering and extraction techniques," 2017. [Online]. Available: https://arxiv.org/abs/1707.02919

[8] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc., 2009.

[9] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[10] J. Zhang and R. Jin, "Understanding bag-of-words model: a statistical framework," *International Journal of Machine Learning and Cybernetics*, 2010.

[11] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[12] T. Tahseen and M. Kabir, *A Comparative Study of Deep Learning Neural Networks in Sentiment Classification from Texts*, 02 2022, pp. 289–305.

[13] T. Filimonova, O. Pursky, V. Babenko, A. Nechepourenko, V. Shvets, and V. Gamaliy, "Text sentiment analysis using different types of recurrent neural networks," 07 2024, pp. 383–387.