EACH | campus capital USP LESTE
Escola de Artes, Ciências e Humanidades
Universidade de São Paulo

Programa de Pós-graduação
em Sistemas de Informação

Enhancing LSTM-Based Sarcasm Detection on Social Media with LLM-Generated Sentences

Gabriel Francisco dos Santos Silva
gabfssilva@usp.br

2024, 17 of June

Contents

The problem

Sarcasm and its challenges

Sarcasm and its challenges

Briefly speaking, sarcasm is a type of irony that aims to mock or make fun of someone.

Sarcasm and its challenges

Briefly speaking, sarcasm is a type of irony that aims to mock or make fun of someone.

Can you tell the different between these two?

Sarcasm and its challenges

Briefly speaking, sarcasm is a type of irony that aims to mock or make fun of someone.

Can you tell the different between these two?

#1:

Such a wonderful day, I love hurricanes!

Sarcasm and its challenges

Briefly speaking, sarcasm is a type of irony that aims to mock or make fun of someone.

Can you tell the different between these two?

#1:

Such a wonderful day, I love hurricanes!

#2:

Seriously, Sherlock? You're such a smart guy!

Sarcasm and its challenges

Sarcasm and its challenges

Irony is a subset of a much broader term, which itself falls under the category of rhetorical questions. (Kreuz 2020)

Sarcasm and its challenges

Irony is a subset of a much broader term, which itself falls under the category of rhetorical questions. (Kreuz 2020)

You can differentiate sarcasm from other types of rhetorical questions by identifying if the sentence has a target.

Sarcasm and its challenges

Irony is a subset of a much broader term, which itself falls under the category of rhetorical questions. (Kreuz 2020)

You can differentiate sarcasm from other types of rhetorical questions by identifying if the sentence has a target.

Nonetheless, it's safe to say that even humans may struggle to identify sarcasm in text.

How do machines learn sarcasm?

How do machines learn sarcasm?

How do machines learn sarcasm?

Mostly **contrast** and **context**.

How do machines learn sarcasm?

Mostly **contrast** and **context**.

The contrast can be detected while finding two very different tones within a single sentence:

How do machines learn sarcasm?

Escola de Artes, Ciências e Humanidades
Universidade de São Paulo

Mostly **contrast** and **context**.

The contrast can be detected while finding two very different tones within a single sentence:

Such a wonderful day, I love hurricanes!

How do machines learn sarcasm?

Mostly **contrast** and **context**.

The contrast can be detected while finding two very different tones within a single sentence:

Such a wonderful day, I love hurricanes!

The context, on the other hard, is all the ~~hidden~~ information available besides the sentence itself:

How do machines learn sarcasm?

Mostly **contrast** and **context**.

The contrast can be detected while finding two very different tones within a single sentence:

Such a wonderful day, I love hurricanes!

The context, on the other hard, is all the ~~hidden~~ information available besides the sentence itself:

Message: Oh, of course she believes the earth is round! She's so smart!

Answering to: Johnny anti-science

Community: Flat-earth society

The state-of-the-art

What's the state-of-the-art on sarcasm recognition?

What's the state-of-the-art on sarcasm recognition?

It's possible to perform sarcasm detection using classic machine learning techniques (Sarsam et al. 2020), but the academia overall agrees Recurrent Neural Networks are a best fit for the task.

What's the state-of-the-art on sarcasm recognition?

It's possible to perform sarcasm detection using classic machine learning techniques (Sarsam et al. 2020), but the academia overall agrees Recurrent Neural Networks are a best fit for the task.

Maynard and Greenwood (2014) propose a technique that utilizes hashtags inside the Tweets to improve model accuracy.

What's the state-of-the-art on sarcasm recognition?

It's possible to perform sarcasm detection using classic machine learning techniques (Sarsam et al. 2020), but the academia overall agrees Recurrent Neural Networks are a best fit for the task.

Maynard and Greenwood (2014) propose a technique that utilizes hashtags inside the Tweets to improve model accuracy.

Using Reddit data, Hazarika et al. (2018) present a unique, still obvious approach: gather context from the user and use it to train the model. This technique is quite clever since the model can know infer based on the user's profile instead of only the sentence alone.

The hypothesis

LLMs as synthetic sentence generators

LLMs as synthetic sentence generators

LLMs are quite good at sarcasm detection (Shrivastava and Kumar 2021), but they are not very resource efficient in terms of computing power. (Bai et al. 2024)

LLMs as synthetic sentence generators

LLMs are quite good at sarcasm detection (Shrivastava and Kumar 2021), but they are not very resource efficient in terms of computing power. (Bai et al. 2024)

If LLMs are effective in sarcasm detection tasks, can they be used as synthetic sentence generators?

LLMs as synthetic sentence generators

LLMs are quite good at sarcasm detection (Shrivastava and Kumar 2021), but they are not very resource efficient in terms of computing power. (Bai et al. 2024)

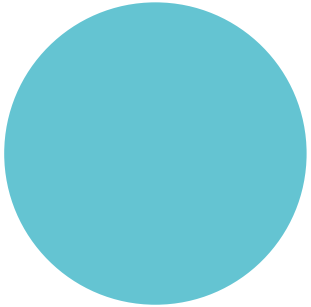
If LLMs are effective in sarcasm detection tasks, can they be used as synthetic sentence generators?

Since we have a dataset of real sarcastic tweets (Abu Farha et al. 2022) for evaluation, can we use these generated synthetic sentences to train and enhance the model for contrast detection?

The training step

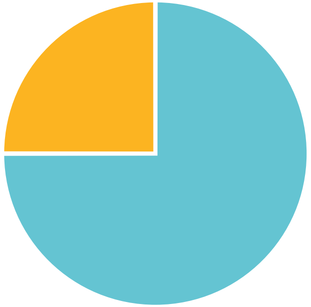
The training step

Only real tweets



50% of synthetic and 50% of real tweets

25% of synthetic and 75% of real tweets



75% of synthetic and 25% of real tweets

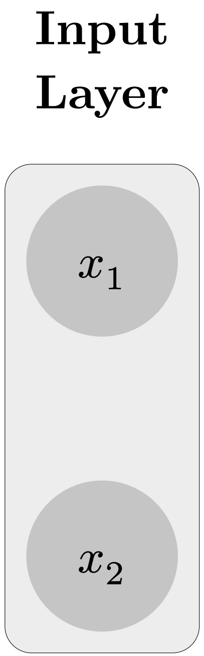
The training step

Since GPT-4o is the most advanced LLM as of the day I write this, it has been chosen as the synthetic sentence generator. (OpenAI 2024a)

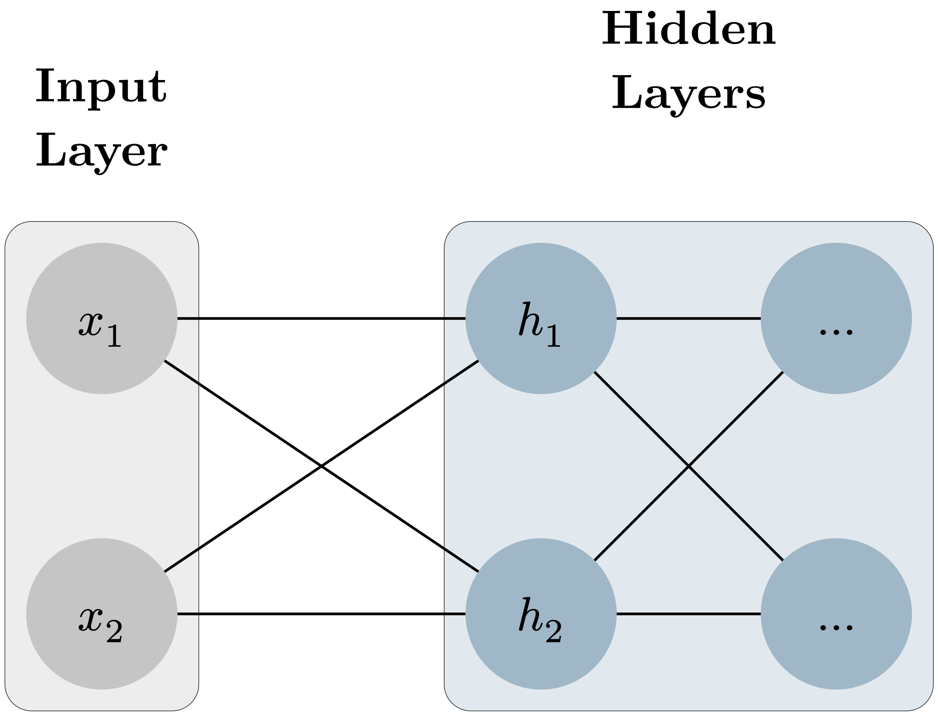
Recurrent Neural Networks in a nutshell

MLPs can’t help much

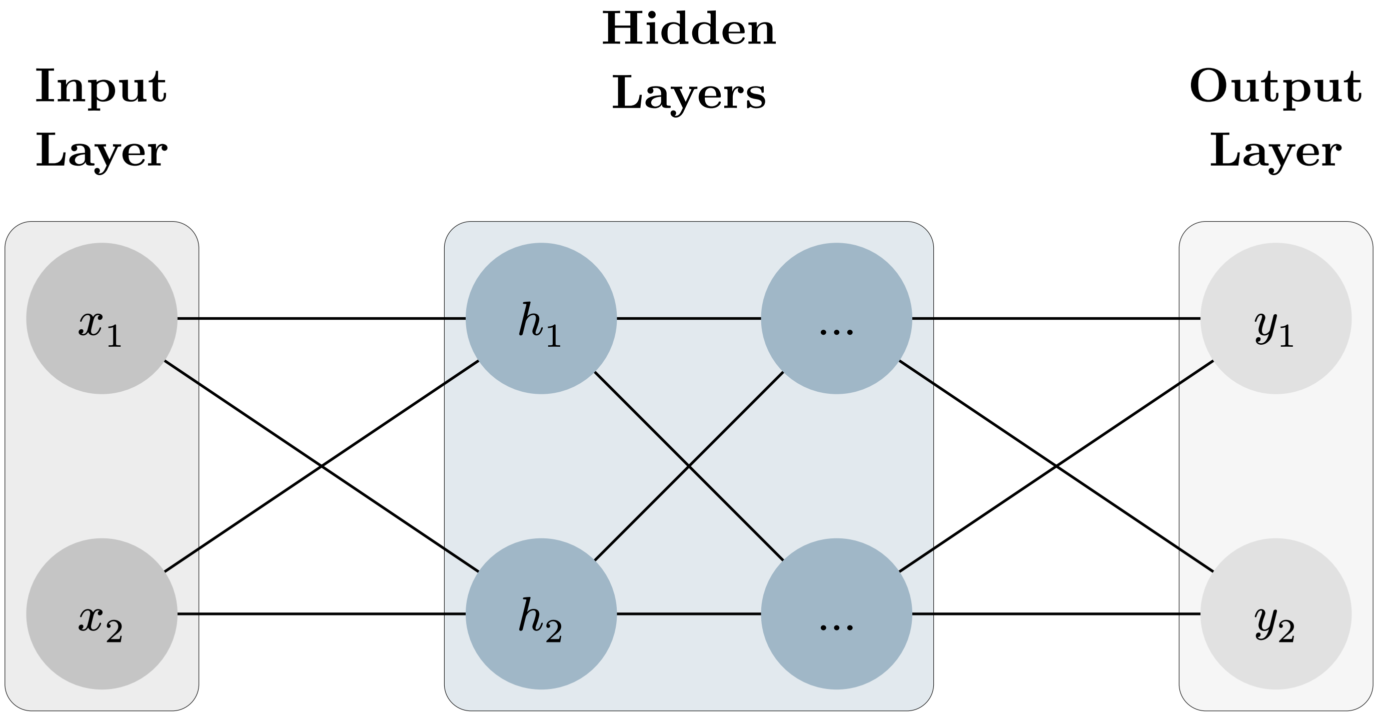
MLPs can't help much



MLPs can't help much

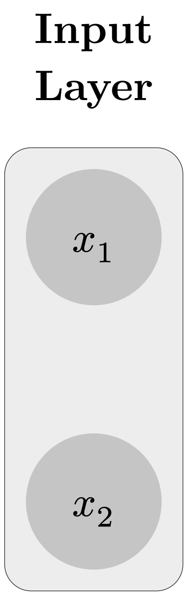


MLPs can't help much

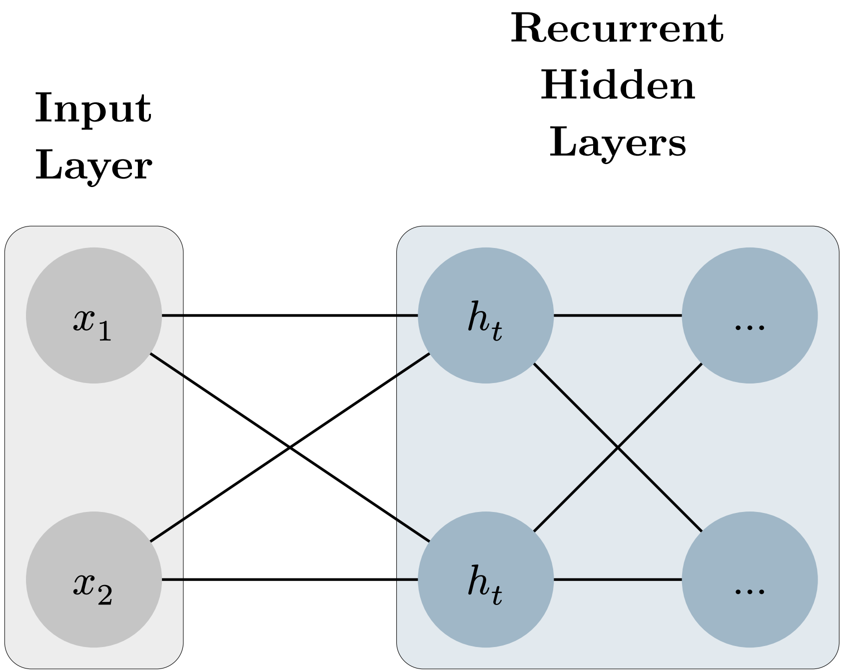


RNNs, maybe?

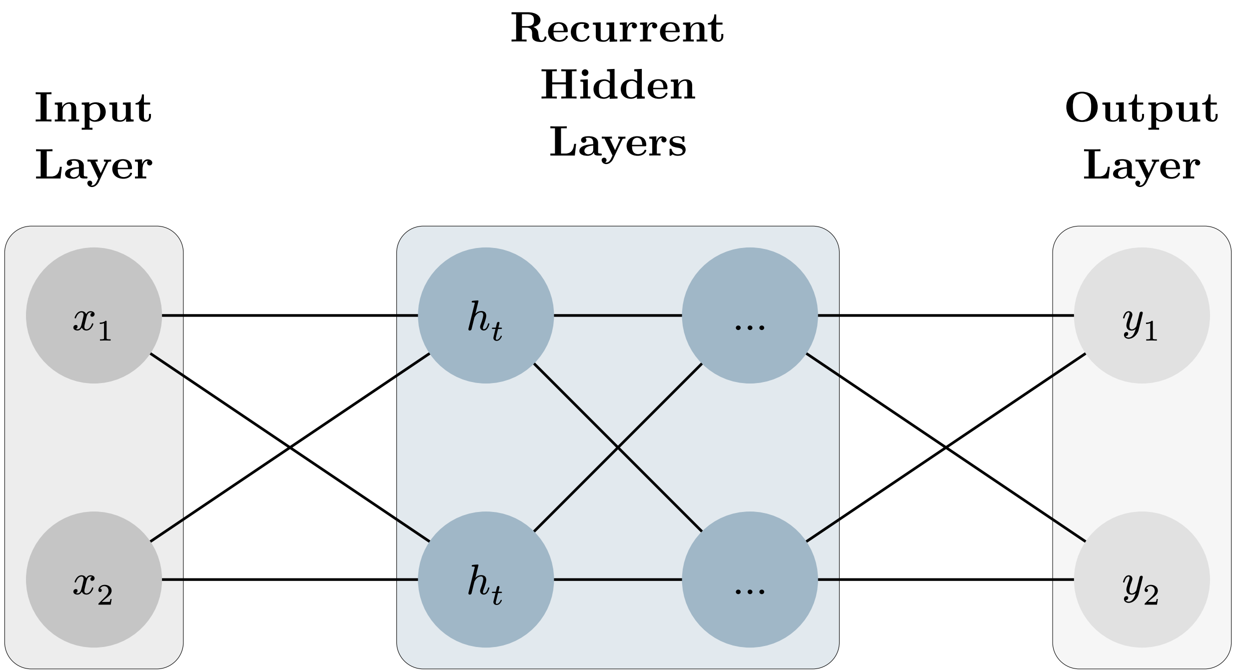
RNNs, maybe?



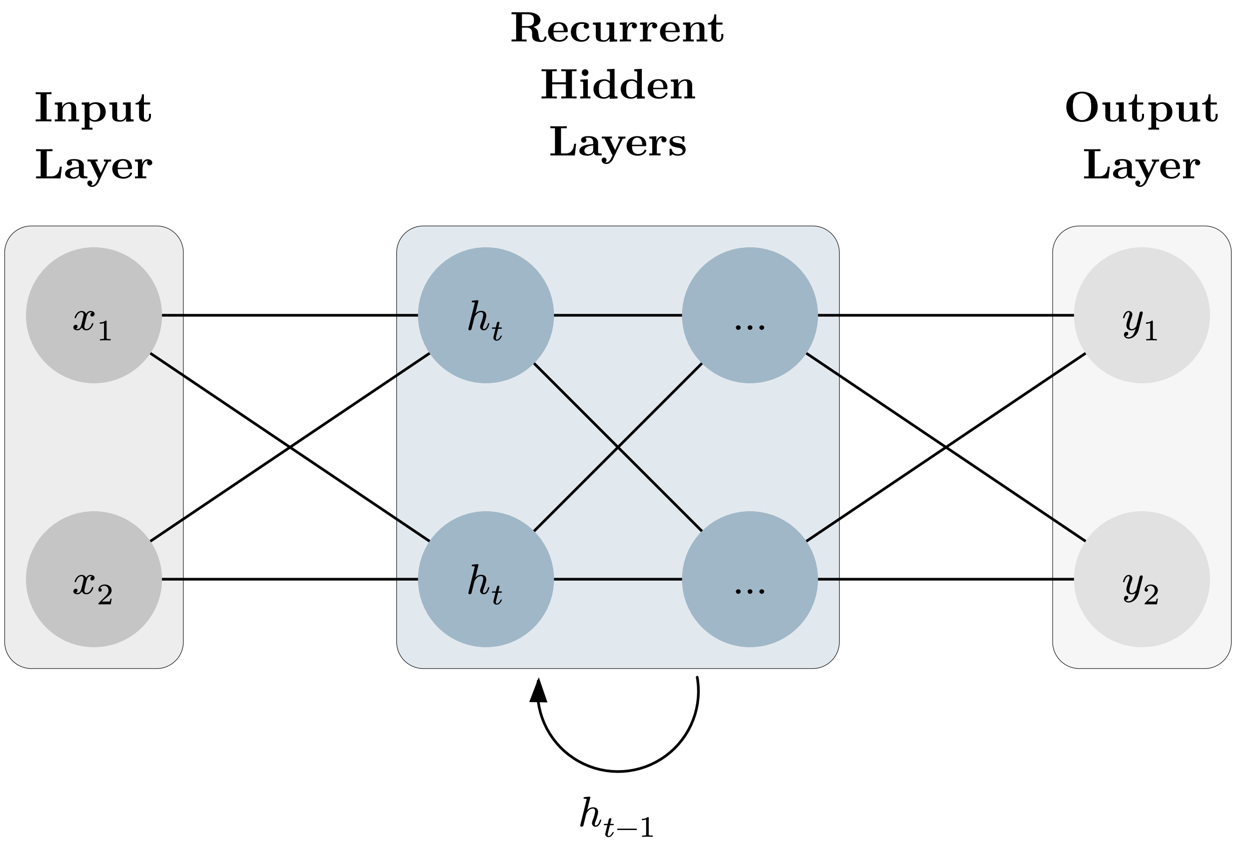
RNNs, maybe?



RNNs, maybe?


Input
Layer
Recurrent
Hidden
Layers
Output
Layer
x_1
x_2
h_t
h_t
...
...
y_1
y_2

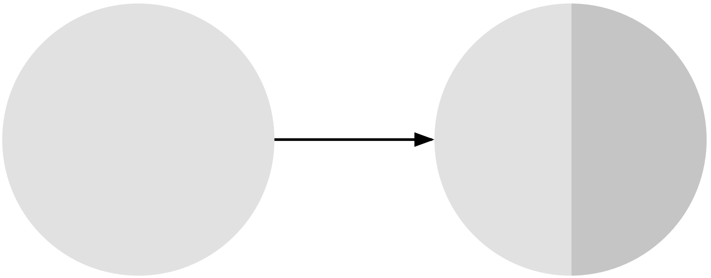
RNNs, maybe?


Recurrent
Hidden
Layers
Input
Layer
Output
Layer
x_1
x_2
h_t
h_t
...
...
y_1
y_2
h_{t-1}

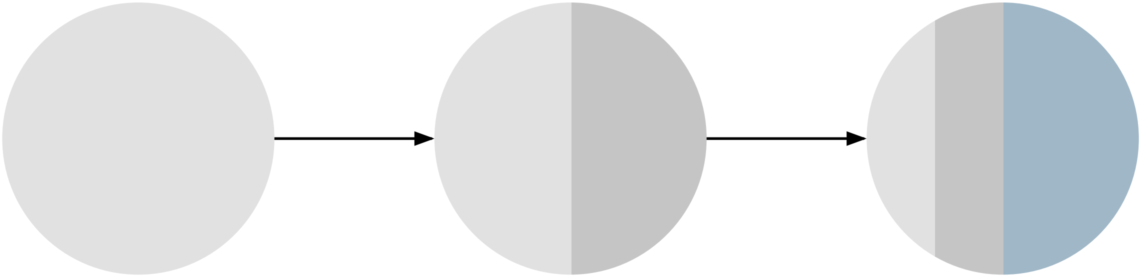
The vanishing gradient problem

The vanishing gradient problem

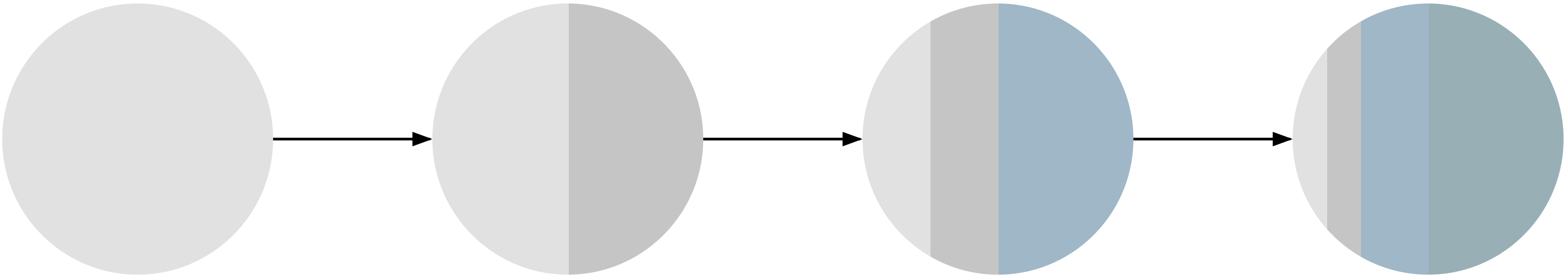
The vanishing gradient problem



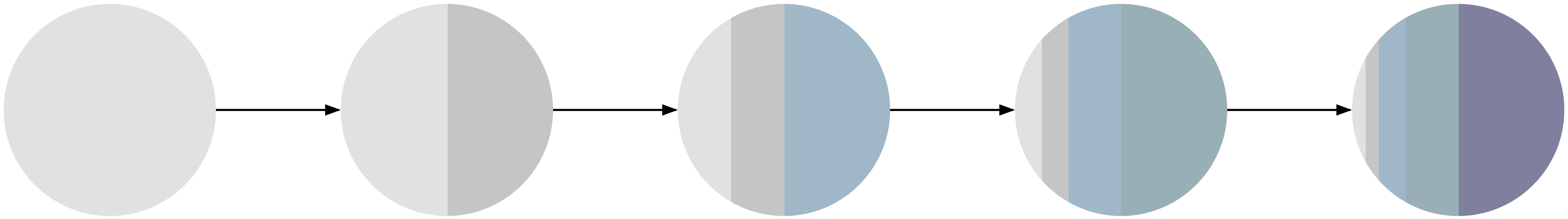
The vanishing gradient problem



The vanishing gradient problem



The vanishing gradient problem



What about Long short-term memory neural networks?

What about Long short-term memory neural networks?

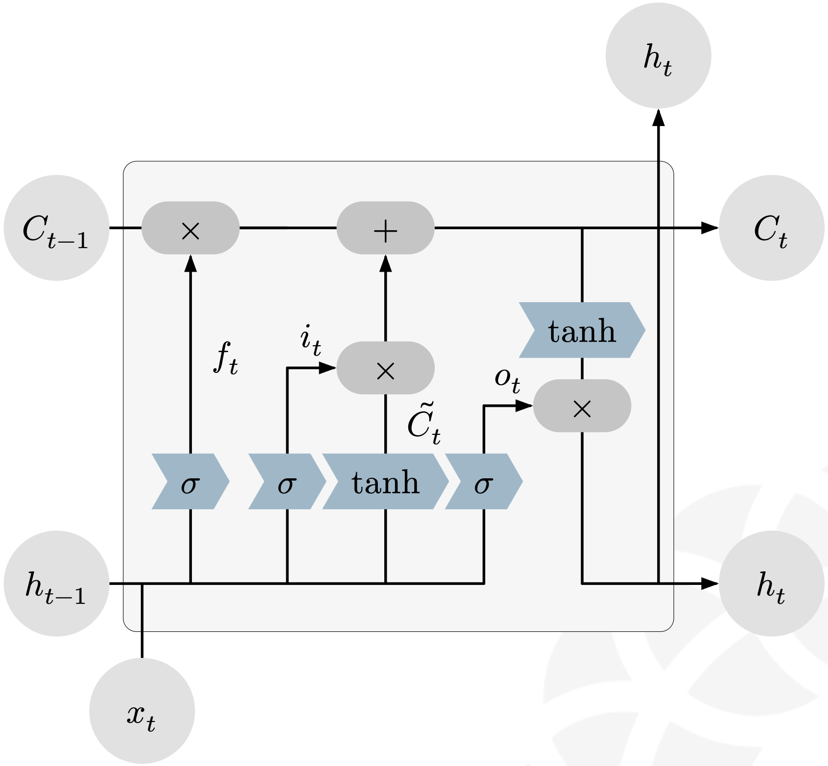
LSTMs implement a more robust mechanism that allows them to retain information over longer periods of time.

By maintaining the state C_t and using the input gate i_t, forget gate f_t, and output gate o_t, they can decide what to add or remove, as well as what to output for the next iteration.

What about Long short-term memory neural networks?

LSTMs implement a more robust mechanism that allows them to retain information over longer periods of time.

By maintaining the state C_t and using the input gate i_t, forget gate f_t, and output gate o_t, they can decide what to add or remove, as well as what to output for the next iteration.



Evaluation

Performance metrics

Performance metrics

The evaluation sample must contain **only real sentences**. These sentences will represent 20% of the total sentences of the chosen dataset. (Abu Farha et al. 2022)

Performance metrics

The evaluation sample must contain **only real sentences**. These sentences will represent 20% of the total sentences of the chosen dataset. (Abu Farha et al. 2022)

The effectiveness of the LSTM model will be evaluated using F1-score (8), accuracy (5), precision (6), and recall (7).

Performance metrics

The evaluation sample must contain **only real sentences**. These sentences will represent 20% of the total sentences of the chosen dataset. (Abu Farha et al. 2022)

The effectiveness of the LSTM model will be evaluated using F1-score (8), accuracy (5), precision (6), and recall (7).

For the validation step, a 10-Fold Stratified Cross-Validation (12) will be used to calibrate the hyperparameters of the Neural Network.

Performance metrics

The evaluation sample must contain **only real sentences**. These sentences will represent 20% of the total sentences of the chosen dataset. (Abu Farha et al. 2022)

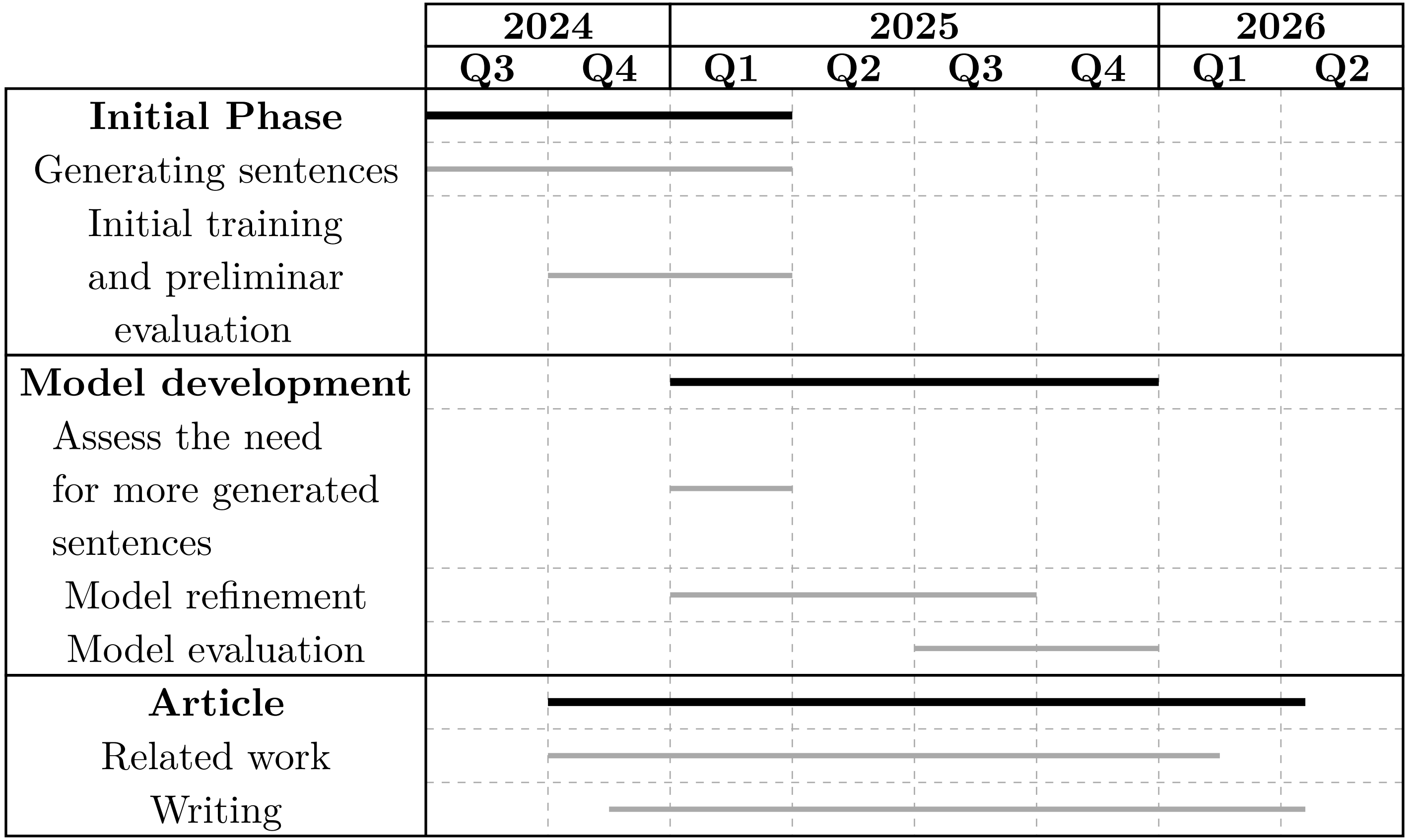
The effectiveness of the LSTM model will be evaluated using F1-score (8), accuracy (5), precision (6), and recall (7).

For the validation step, a 10-Fold Stratified Cross-Validation (12) will be used to calibrate the hyperparameters of the Neural Network.

The effectiveness of the model will be benchmarked against other models as well: GPT 4, GPT 4.0o, BERT, Llama 2 and Llama 3 (Achiam et al. 2023; OpenAI 2024b; Devlin et al. 2019; Touvron et al. 2023; Meta 2024).

Timeline

Timeline


2024
2025
2026
Q3
Q4
Q1
Q2
Q3
Q4
Q1
Q2
Initial Phase
Generating sentences
Initial training and preliminar evaluation
Model development
Assess the need for more generated sentences
Model refinement
Model evaluation
Article
Related work
Writing

Bibliography

[1] R. J. Kreuz, *Irony and sarcasm*. The Mit Press, 2020. [Online]. Available: https://mitpress.mit.edu/9780262538268/

[2] S. M. Sarsam, H. Al-Samarraie, A. I. Alzahrani, and B. Wright, “ Sarcasm detection using machine learning algorithms in Twitter: A systematic review ,” *International Journal of Market Research*, vol. 62, no. 5, pp. 578–598, 2020, doi: 10.1177/1470785320921779.

[3] D. Maynard and M. A. Greenwood, “Who cares about Sarcastic Tweets? Investigating the Impact of Sarcasm on Sentiment Analysis.,” in *International Conference on Language Resources and Evaluation*, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:14079970

[4] D. Hazarika, S. Poria, S. Gorantla, E. Cambria, R. Zimmermann, and R. Mihalcea, “CASCADE: Contextual Sarcasm Detection in Online Discussion Forums,” in *Proceedings of the 27th International Conference on Computational Linguistics*, E. M. Bender, L. Derczynski, and P. Isabelle, Eds., Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1837–1848. [Online]. Available: https://aclanthology.org/C18-1156

[5] M. Shrivastava and S. Kumar, “A pragmatic and intelligent model for sarcasm detection in social media text,” *Technology in Society*, vol. 64, p. 101489–101490, 2021, doi: https://doi.org/10.1016/j.techsoc.2020.101489.

[6] G. Bai *et al.*, “Beyond Efficiency: A Systematic Survey of Resource-Efficient Large Language Models.” 2024.

[7] I. Abu Farha, S. V. Oprea, S. Wilson, and W. Magdy, “SemEval-2022 Task 6: iSarcasmEval, Intended Sarcasm Detection in English and Arabic,” in *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 802–814. [Online]. Available: https://aclanthology.org/2022.semeval-1.111

[8] OpenAI, “simple-evals: Repository for Evaluating Language Models.” Accessed: Jun. 10, 2024a. [Online]. Available: https://github.com/openai/simple-evals

[9] O. J. Achiam *et al.*, “GPT-4 Technical Report,” 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257532815

[10] OpenAI, “Hello GPT-4o.” Accessed: May 26, 2024b. [Online]. Available: https://openai.com/index/hello-gpt-4o/

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." 2019.

[12] H. Touvron *et al.*, "Llama 2: Open Foundation and Fine-Tuned Chat Models," *ArXiv*, 2023, [Online]. Available: https://api.semanticscholar.org/CorpusID:259950998

[13] Meta, " Introducing Meta Llama 3: The most capable openly available LLM to date ." [Online]. Available: https://ai.meta.com/blog/meta-llama-3/

Appendix

Long Short-Term Memory Networks

EACH
campus capital USP LESTE
Escola de Artes, Ciências e Humanidades
Universidade de São Paulo

Cell state: This is the "memory" part of the LSTM, carrying relevant information throughout the processing of the sequence. Plays a major role in transferring past knowledge to future states.

Input gate: Decides how much of the newly computed state for the current input x_t should be added to the cell state.

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \end{aligned} \tag{1}$$

Forget gate: Decides how much of the current cell state should be kept. Anything that was not forgotten is passed along to the next step.

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big) \tag{2}$$

Update Cell State: This step combines the old state C_{t-1} and the new candidate values, modulated by the forget gate and the input gate.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{3}$$

Output: Decides what part of the cell state should be output at this step

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t \cdot \tanh(C_t) \end{aligned} \tag{4}$$

Evaluation metrics

- TP: True Positives
- TN: True Negatives
- FP: False Positives
- FN: False Negatives

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{5}$$

$$precision = \frac{TP}{TP + FP} \tag{6}$$

$$recall = \frac{TP}{TP + FN} \tag{7}$$

$$f1 = \frac{2 \cdot (precision \cdot recall)}{precision + recall} \tag{8}$$

Activation functions

Sigmoid:

$$\sigma(x) = \frac{1}{1} + e^{-x} \tag{9}$$

Tanh:

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{10}$$

Cross Validation

A dataset D is split into k equal parts. The model is trained and validated k times, with one of the parts D_k being used as the validation set V_i in each iteration, while the other parts T_i are used for training. Performance metrics are calculated at each step. The overall error rate is the average of the error rates from each step, where e_i is the error rate in the i-th iteration:

$$E = \frac{1}{k} \cdot \sum_{i=1}^{k} e_i \tag{11}$$

Stratified Cross Validation

The dataset D is first stratified into k parts $(D_1, D_2, ..., D_k)$, ensuring that the sample ratio for each class in each part D_i is as close as possible to the ratio of that class in the complete dataset D. If C represents a class label, then the class ratio C in each D_i, denoted as p_{C_i}, should closely match p_C, the ratio of the class C throughout the dataset D:

$$p_{C_i} \approx p_C, \quad \forall i = 1, 2, ..., k \tag{12}$$

Once the data is stratified, cross validation proceeds as usual. Each part D_i is used once as a validation set V_i, while the remaining combined parts form the training set T_i.

Questions?