Detecting flirting in text using deep learning

Wayne Wang
University of Michigan
wswang@umich.edu

Jaehyun Shim
University of Michigan
jaeday@umich.edu

Zirui Zhao
University of Michigan
zhaojer@umich.edu

1 INTRODUCTION

1.1 Background Information & Motivation

Flirting is a social behavior in the form of spoken, written, and body language by one person to another to either suggest romantic interest or simply for amusement.

However, flirting can be intentionally subtle and indirect, so it is hard to decipher whether or not someone is actually expressing interest [12, 13, 23]. In a study of flirtation perception accuracy with over 100 participants, Hall et al. found that individuals only correctly detected flirting 28% of the time, and third-party observers were even less accurate [12]. Moreover, men tend to have a harder time in correctly identifying flirting compared to women [9].

Not being able to flirt puts one at a disadvantage. Studies have shown that people who scored low in identifying and engaging in flirting tend to be involuntarily single [1, 2], and have difficulties in initiating and maintaining an intimate relationship [3].

With smartphones nowadays, flirting often happens online via text messages [16, 23], and this have made flirting even more subtle due to the lack of nonverbal cues [11]. Hence, accurately detecting flirting in text messages is crucial to avoid potential embarrassment from misinterpretation but also to recognize genuine interest, potentially leading to new relationships.

Fortunately, with the advancement of artificial intelligence, machines can now perform tasks that typically require human intelligence. One such task is sentiment analysis, an application within the natural language processing (NLP) field, which involves systematically identifying, categorizing, or quantifying the affective states and subjective information in textual material.

Traditional sentiment analysis often involved classic supervised machine learning (ML) algorithms, including Naive Bayes (NB), Support Vector Machine (SVM), etc [22]. However, with the rapid advancement of deep learning in the 2010s, researchers began to employ neural networks in NLP, specifically, Recurrent Neural Networks (RNN) and its variants like Long Short-Term Memory (LSTM) [22]. These neural network models significantly outperform traditional ML models on sentiment analysis tasks [4, 6]. With the introduction of Transformers in 2017, the NLP field has shifted to using pre-trained transformer-based models or large language models (LLMs), like Bidirectional Encoder Representations from Transformers (BERT) and Generative Pre-trained Transformer (GPT), as a basis and then fine-tune the model to the new task [6].

Sentiment analysis has been used extensively in classifying whether a piece of text like a movie or product review is positive or negative [17, 20, 24], as well as detecting a broad range of emotions on social media posts [10, 18]. However, very few studies have explored sentiment analysis in the domain of flirting. Thus, creating an intelligent agent that can accurately detect flirting in text not only aids those seeking romantic relationships but also contributes to NLP by expanding sentiment analysis applications.

1.2 Problem Statement

In this study, our goal is to accurately detect flirting in text using advanced deep learning algorithms, specifically a Neural Network model and a Transformer model, and compare their performance and cost. We ask the following research questions:

RQ1: Do recent deep learning models perform better than traditional machine learning models in detecting flirting?

RQ2: How do deep learning models compare to each other, i.e. how does a Transformer model compare to a Neural Network model, in detecting flirting?

We propose the following hypotheses:

H1: Deep learning models, including both Neural Networks and Transformers, perform better than traditional ML models.

H2: Transformer models yield better performance than Neural Network models, but their cost may outweigh the performance gains.

1.3 Overview of Proposed Work

Our research contributes to the NLP field by applying state-of-the-art deep learning models to a new domain of flirting detection. We train LSTM (neural network) and fine-tune BERT (transformer) on a diverse dataset of flirty and neutral texts, while devising an unique standardization technique and performing rigorous hyperparameter tuning. Then, we analyze and compare the models' performance and cost using multiple evaluation metrics. Our results support both hypotheses.

2 RELATED WORK

Flirting Detection. Jurafsky et al. classified 3 interactional styles, "awkward", "friendly", and "flirtatious", in over 1000 real-life speed dating conversations [14]. Using this corpus, they manually extracted features including dialogue acts, lexical choice, and prosody, to feed into a logistic regression model, achieving an accuracy of 75% – an impressive baseline, given the challenging noise and the naturalistic setting of their data. Leveraging the same corpus and feature extraction, the same authors also developed an SVM model that focused exclusively on detecting flirting, which achieved an accuracy of 71.5%, surpassing both the baseline and human evaluators [19].

However, both studies employed traditional ML models, which rely on domain experts to manually select and design features, a difficult and time-consuming process that may limit the system's ability to capture the intricate nuances of flirting. Our research aims to bridge this gap by employing more recent and advanced deep learning algorithms, which can automatically learn and prioritize relevant features, adapting the model for texts where some lexical features are unavailable and modeling intricate patterns and contextuality in the data that might be difficult to engineer manually, potentially resulting in improved accuracy [4, 6].

A Comparison of LSTM and BERT for Small Corpus. Ezen-Can found that LSTM had *higher* accuracy than BERT in classifying user intent when interacting with chatbots [7]. However, the author utilized a small, narrow-scoped dataset with uniformly short instances and restricted features, which hinder BERT's learning ability through tokenization. Moreover, the study's minimal exploration of model hyperparameters further limits insights into model performance.

We aim to improve upon these limitations by utilizing a larger dataset containing instances of various lengths and with diverse lexical features, and performing more extensive hyperparameter tuning.

SentimentGPT. Kheiri and Karimi investigated GPT models' performance in sentiment analysis of social media posts, employing three genres of GPT models: instruct-based (GPT3.5 Turbo), fine-tuned (Ada, Babbage, Curie), and embedding (pre-trained GPT models together with XGboost and Random Forest) [15]. Using the SemEval-2017 dataset on social media posts for sentiment analysis, the authors found that GPT models significantly outperform non-LLM models evaluated by F1-score; both fine-tuned GPT models and embedding GPT models achieve high recall scores.

Our study aims to address the following gaps presented in their study: (1) exploring lightweight BERT-based models for sentiment analysis as opposed to GPT which is resource-intensive and expensive; (2) evaluating the performance specifically in binary classification of sentiment (i.e. flirting), rather than categorical classification, thus bringing more nuanced challenges to the models; and (3) utilizing conversation-based datasets instead of social-media posts for better contextual understanding by LLMs.

3 METHODS

3.1 Dataset

Our initial dataset was directly sourced from Hugging Face [21], consisting of 2,114 instances. Our preliminary results on this dataset showed poor model performance, likely due to the lack of training data. Thus, we also employed a second dataset found on Github [8] and combined the two to form one larger dataset while ensuring no duplicates. The data in both datasets were sourced from real-life conversations on Tinder.

Together, the final dataset consisted of 4,213 instances of texts (input variable), each labeled with 0 as 'neutral' and 1 as 'flirty' (output variable); having 60% of the instances as neutral and 40% as flirty made the dataset fairly balanced. Though the increased dataset size was beneficial, we acknowledged that there may be some downsides to this combined method, since the datasets were manually labeled by two different authors, who might have different interpretations of textual flirtatiousness.

The texts in the dataset varied greatly in length (*Min*: 1, *Q*1: 4, *M*: 7, *Q*3: 11, *Max*: 135) and content, each containing a different mixture of English words, punctuation, emojis, and other symbols, thus covering a diverse set of linguistic features in flirting. The binary class labeling and diverse instances of the dataset made it suitable for our task, which was to classify whether a given piece of text is flirty or not. The data were split into training (3,683), validation (474), and test (581) sets. We used the validation dataset to assess the models' performance during training, allowing us to iteratively adjust model configurations to optimize its performance on unseen data. The separate test set provided a valid, unbiased assessment of the two models' performance in real-world scenarios.

The textual data in the dataset were preprocessed via standardization (or normalization), tokenization, and vectorization, which involved cleaning up the raw text, splitting each text (e.g. a sentence) into a sequence of tokens (e.g. individual words), and converting the tokens into numbers that can be fed into the model, respectively. To achieve these three steps for the LSTM model, we utilized TensorFlow's `TextVectorization` layer, while defining additional hyperparameters like max tokens and sequence length. For BERT, we simply employed Hugging Face's default `BertTokenizer`, specifying it to truncate/pad input sequences to the max length.

3.2 Models

We developed both models on Google Colab, exploiting state-of-the-art machine learning library and training resources (e.g. GPUs).

3.2.1 *LSTM.*

We built our LSTM model using TensorFlow Keras. The initial model followed a standard architecture, starting with a `TextVectorization` layer to convert the raw textual input into a fixed-length sequence of token ids (numbers), followed by an `Embedding` layer that maps each token id to a dense vector which captures its meaning, then an `LSTM` layer to process the sequence, and finally a single-neuron `Dense` layer to output a number between 0 and 1, indicating whether the input text is flirty (if > 0.5) or neutral (if ≤ 0.5). Training for 10 epochs on Google Colab CPU, the model achieved a validation accuracy of 0.721 and loss of 0.553 on the initial (validation) dataset, while demonstrating severe overfitting.

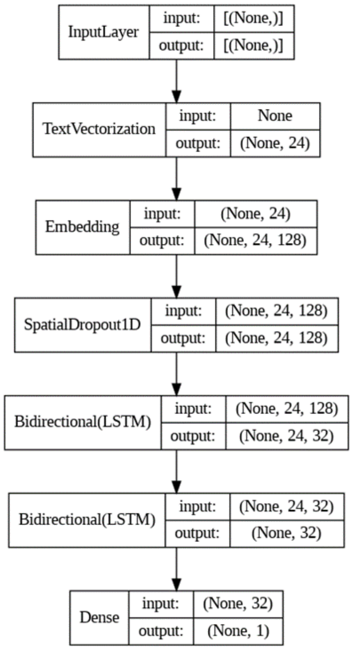

Figure 1: LSTM Architecture

After rigorous hyperparameter tuning, experimentation with the architecture, and adding a second dataset (as discussed in 3.1), we improved the model's validation accuracy and loss to 0.941

and 0.240 (measured on the final dataset), respectively. Though the dataset enhancement played a significant role, we found that certain hyperparameters also greatly contributed to this improvement. Notably, for the standardization step (in the `TextVectorization` layer), instead of removing all special characters/symbols, which is the common preprocessing technique for NLP tasks, we found that keeping the special characters, including punctuation and emojis, and treating each of them as a token led to the highest validation accuracy, likely due to the importance of punctuation and emojis in flirting. Additionally, a smaller model architecture, i.e. fewer LSTM units and layers, together with a moderate dropout rate in a `SpatialDropout1D` layer and LSTM layer(s) not only led to a better validation accuracy and loss, but also reduced overfitting during training, resulting in better generalization to unseen data. Our final LSTM architecture is shown in Fig. 1.

3.2.2 BERT. We relied on the Hugging Face's `transformers` library, which builds upon `PyTorch` as the underlying training architecture, where we imported and fine-tuned a BERT model. We first instantiated a `BertTokenizer` to convert raw text data into tokenized sequences. Then, the pre-trained base BERT model (uncased) [5] was loaded using `BertForSequenceClassification` with the number of labels as 2, indicating the model is performing a binary text classification task, which outputs the class id (0 = 'neutral' and 1 = 'flirty') and an associated probability. Then, training arguments (hyperparameters) were provided – we started with default hyperparameters from the model library without performing parameter searching as a baseline. After defining metric computation methods (see 3.3), we performed 5 epochs of fine-tuning using the (final) training dataset on Google Colab GPU. This pipeline is illustrated in Fig. 2.

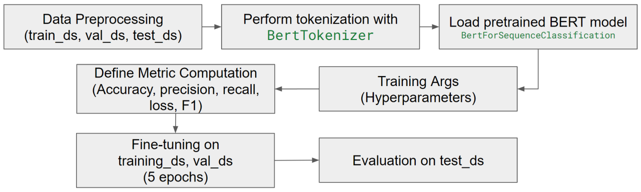
Data Preprocessing
(train_ds, val_ds, test_ds)
Perform tokenization with
BertTokenizer
Load pretrained BERT model
BertForSequenceClassification
Define Metric Computation
(Accuracy, precision, recall,
loss, F1)
Training Args
(Hyperparameters)
Fine-tuning on
training_ds, val_ds
(5 epochs)
Evaluation on test_ds

Figure 2: Pipeline for BERT Training

3.3 Evaluation

After both models finished training, we evaluated their performance on the same test dataset (as discussed in 3.1) via a comprehensive set of metrics. We gauged overall accuracy by the ratio of correctly predicted instances to the total predictions, defining a prediction as correct when the model's output matched the true label from the dataset. The loss was computed using binary cross-entropy loss function, reflecting the disparity between the model's predictions and the actual outcomes. Precision assessed the proportion of true positives among all positive predictions, whereas recall captured the model's ability to identify all true positives. The F1-score, the harmonic mean of precision and recall, provided a balanced view of the model's predictive performance. Lastly, the confusion matrix detailed the specific counts of the models' misclassifications. The SVM model from Jurafsky et al. [19] discussed in the Related Work section was used as the baseline.

Additionally, as our project was application-centric, we were interested in evaluating the two models in a real-world setting, where energy consumption might behave as the cost for performance. Thus, we also measured the energy consumption/cost for the two models using `codecarbon` library's kWh metric. While this library also provided a metric on carbon footprint, we decided not to use it since its calculation was dependent on the geological region where the model would be running. To have a holistic view of the power consumption of each model, we decided to use the kWh metric, whose measurement is consistent across different factors, only dependent on the models themselves. The measurement is performed on queries instead of training, since while training is one-off, if deployed to end-users, the amount of queries can easily scale up.

4 RESULTS AND ANALYSIS

The performance of the LSTM and BERT models on the test dataset is shown in Table 1. Clearly, both the LSTM model and the BERT model achieved significantly higher accuracy (both over 90%) compared to the baseline SVM model of 71.5% [19], indicating that these advanced deep learning models outperform traditional ML models in flirting detection, thus supporting our hypothesis (i).

	Accuracy	Loss	Precision	Recall	f1-score
LSTM	0.941	0.268	0.976	0.847	0.907
BERT	0.957	0.263	0.994	0.878	0.933

Table 1: Performance of LSTM and BERT on Test dataset

Comparing the two models, BERT consistently demonstrates better performance than LSTM in all five evaluation metrics, which reinforces the powerful learning ability of the Transformer architecture in natural language understanding and classification tasks. To elaborate on each metric, a higher test accuracy of BERT at 95.7% compared to 94.1% of LSTM indicates BERT can more accurately predict whether a given piece of text is flirty or not in general. Both models have a fairly low loss value, meaning that they both have low overfitting during training and generalize well to new, unseen data.

Both models exhibit extremely high precision, meaning that when either model predicts a text is flirty, it is almost always going to be correct, with BERT being correct 99.4% of the time and LSTM being 97.6%. The models' great ability in avoiding false positives ensures that users can safely "make a move" whenever the model predicts flirty. On the other hand, both models have relatively lower recall compared to other metrics, meaning that given a text is truly flirty, the model can correctly predict 'flirty' 87.8% of the time if it is BERT and 84.7% if it is LSTM; in a real world context, this implies that the models may occasionally miss out certain flirting cues, with BERT being less likely to miss. This relatively poor ability in identifying true positives is not always bad, since it makes the models more "conservative" and can avoid potential embarrassment for the users. The higher F1 score of BERT (93.3%) than LSTM (90.7%) simply indicates that overall, BERT has a better balance between precision and recall, working well where both false positives and false negatives are important. These results align with the first part of our hypothesis (ii), that BERT exhibits superior performance across all five evaluation metrics compared to LSTM.

	LSTM		BERT	
	Predicted negative	Predicted positive	Predicted negative	Predicted positive
True negative	380	4	383	1
True positive	30	167	24	173

Table 2: Confusion Matrix of LSTM and BERT

To examine the model performance in further details, we computed the confusion matrix for both models in Table 2. We observed a notably lower count of misclassifications, in terms of both false positives and false negatives, with BERT in contrast to LSTM. This further shows BERT's ability to learn intricate linguistic representations thanks to its advanced attention mechanism and pre-training techniques. After investigating the specific text instances in the test dataset in which the models made incorrect predictions, we found that the misclassifications made by BERT are a subset of those made by LSTM. For instance, "Oops that was for Julie sorry" is a false negative prediction made by both models; other misclassifications are similar to this in the sense that they are all very subtle even when knowing the true labels, posing challenges for both the models and human evaluators alike. Moreover, since the data were sourced from Tinder, conversations with each text message being its own instance in the dataset, some instances, though labeled as flirty, might only make sense as flirty when considering the entire conversation, e.g., "Lol funny huh?" and "is it a golden retriever? mine's name is honey"; this, being an inherent limitation of the dataset, underscores the difficulty in understanding text messages out of context and might account for both models' relatively low recall rate. This observation serves as a notable consideration in the interpretation of our study findings and suggests avenues for future research aimed at addressing these complexities comprehensively.

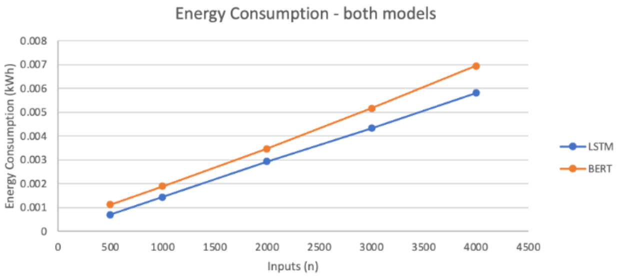

Figure 3: Energy Consumption Graph for LSTM and BERT

In our examination of the energy consumption patterns exhibited by LSTM and BERT models, our analysis uncovers a distinct disparity in their respective energy utilization profiles. Fig. 3 illustrates that BERT consumes notably higher energy resources compared to LSTM. While both models demonstrate a linear relationship between energy consumption and input size, it becomes evident that BERT's energy consumption rate accelerates more steeply as inputs approach the 3000 mark. This observation underscores a crucial implication: when considering deployment at scale, particularly in scenarios where horizontal scaling is necessary to accommodate millions of daily requests or more, BERT's energy demands would significantly surpass those of LSTM. This finding lends empirical support to the remainder of our hypothesis (ii), positing that the cost associated with deploying BERT outweighs its performance advantages over LSTM in the context of flirty sentiment analysis. Given the marginal discrepancy observed across various performance metrics between LSTM and BERT, the pragmatic choice leans towards leveraging LSTM over BERT for an overall optimal resource allocation and efficiency.

5 CONCLUSION

In this project, we implemented the training and fine-tuning pipelines for LSTM and BERT to perform binary classification on potentially flirtatious text messages. We evaluated two approaches' performance using multiple performance and energy consumption metrics that resulted in two key findings. First, both of our LSTM and BERT models outperform the baseline SVM model in the previous study by achieving higher accuracy. Second, while BERT performs slightly better than LSTM, its energy consumption cost outweighs the slight performance gain.

We consider several ethical impacts with our project. In our endeavor to harness AI models for assisting individuals in navigating flirtatious interactions, we aspire to not only enhance communication skills but also foster stronger interpersonal connections, while also helping study flirtatious behaviors and contexts across various cultures and genders which would improve our understanding of cultural variations and gender roles. However, to ensure ethical deployment - firstly, it is paramount to prioritize user privacy and obtain informed consent for data usage, to uphold robust privacy standards and foster trust in the technology; secondly, it is also important to explore aspects of energy efficiency of each model to reduce power consumption.

Overall, we had a very positive research project experience, as our model development and evaluation all went smoothly, and our group dynamics were incredible. We met our expected goal, which was to build and evaluate models to detect flirting, and completed all tasks and milestones mentioned in the progress report, including dataset enhancement, exploration of different data preprocessing (standardization) techniques, hyperparameter tuning, and data collection for other evaluation metrics. The only task we did not explore was employing alternative BERT models, because our existing BERT model already demonstrated exceptional performance. This is left as a possible future work.

Future work should also consider creating a dataset by manually sourcing first-hand data to further boost data quality and validity, developing models to classify the flirtatiousness of an entire conversation instead of individual text messages, and utilizing Generative AI in flirting detection and evaluating its performance against our models.

6 CODE AVAILABILITY

Our code is available at https://github.com/waynew99/592-final-project-team6.

REFERENCES

[1] Menelaos Apostolou. 2021. Involuntary singlehood and its causes: The effects of flirting capacity, mating effort, choosiness and capacity to perceive signals of interest. *Personality and Individual Differences* 176 (July 2021). https://doi.org/10.1016/j.paid.2021.110782

[2] Menelaos Apostolou and Elli Michaelidou. 2024. Why people face difficulties in attracting mates: An investigation of 17 probable predictors of involuntary singlehood. *Personality and Individual Differences* 216 (Jan. 2024), 1–6. https://doi.org/10.1016/j.paid.2023.112422

[3] Menelaos Apostolou, Irene Papadopoulou, Michael Christofi, and Demetris Vrontis. 2019. Mating performance: Assessing flirting skills, mate signal-detection ability, and shyness effects. *Evolutionary Psychology* 17, 3 (Sept. 2019). https://doi.org/10.1177/1474704919872416

[4] Yogesh Chandra and Antoreep Jana. 2020. Sentiment analysis using machine learning and deep learning. In *2020 7th International Conference on Computing for Sustainable Global Development (INDIACom 2020)*. 1–4. https://doi.org/10.23919/INDIACom49435.2020.9083703

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[6] Kaushik Dhola and Mann Saradva. 2021. A comparative evaluation of traditional machine learning and deep learning classification techniques for sentiment analysis. In *2021 11th International Conference on Cloud Computing, Data Science Engineering (Confluence 2021)*. 932–936. https://doi.org/10.1109/Confluence51648.2021.9377070

[7] Aysu Ezen-Can. 2020. A comparison of LSTM and BERT for small corpus. (2020). arXiv:2009.05451

[8] Alyssa Fernandez. 2021. Flirtation analysis. https://github.com/alyssafrndz/Flirtation-analysis/blob/main/flirting_rated.csv

[9] Betty La France, David D. Henningsen, Aubrey Oates, and Christina M. Shaw. 2009. Social-sexual interactions? Meta-analyses of sex differences in perceptions of flirtatiousness, seductiveness, and promiscuousness. *Communication Monographs* 76, 3 (Aug. 2009), 263–285. https://doi.org/10.1080/03637750903074701

[10] Bharat Gaind, Varun Syal, and Sneha Padgalwar. 2019. Emotion detection and analysis on social media. arXiv:1901.08458

[11] Karl Grammer. 1990. Strangers meet: Laughter and nonverbal signs of interest in opposite-sex encounters. *Journal of Nonverbal Behavior* 14, 4 (Dec. 1990), 209–236. https://doi.org/10.1007/BF00989317

[12] Jeffrey A. Hall, Chong Xing, and Seth Brooks. 2015. Accurately detecting flirting: Error management theory, the traditional sexual script, and flirting base rate. *Communication Research* 42, 7 (April 2015), 939–958. https://doi.org/10.1177/0093650214534972

[13] David Dryden Henningsen. 2004. Flirting with meaning: An examination of miscommunication in flirting interactions. *Sex Roles* 50 (April 2004), 481–489. https://doi.org/10.1023/B:SERS.0000023068.49352.4b

[14] Dan Jurafsky, Rajesh Ranganath, and Dan McFarland. 2009. Extracting social meaning: identifying interactional style in spoken conversation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (Boulder, Colorado) *(NAACL '09)*. Association for Computational Linguistics, USA, 638–646.

[15] Kiana Kheiri and Hamid Karimi. 2023. SentimentGPT: Exploiting GPT for advanced sentiment analysis and its departure from current machine learning. (2023). arXiv:2307.10234

[16] James Lykens, Molly Pilloton, Cara Silva, Emma Schlamm, Kate Wilburn, and Emma Pence. 2019. Google for sexual relationships: Mixed-methods study on digital flirting and online dating among adolescent youth and young adults. *JMIR Public Health and Surveillance* 5, 2 (May 2019). https://doi.org/10.2196/10695

[17] Wedjdane Nahili, Khaled Rezeg, and Okba Kazar. 2021. Sentiment analysis on product reviews data using supervised learning: A comprehensive review of recent techniques. In *Proceedings of the 10th International Conference on Information Systems and Technologies* (Lecce, Italy) *(ICIST '20)*. Association for Computing Machinery, New York, NY, USA, Article 5, 6 pages. https://doi.org/10.1145/3447568.3448513

[18] Pansy Nandwani and Rupali Verma. 2021. A review on sentiment analysis and emotion detection from text. *Social Network Analysis and Mining* 11, 81 (Aug. 2021). https://doi.org/10.1007/s13278-021-00776-6

[19] Rajesh Ranganath, Dan Jurafsky, and Dan McFarland. 2009. It's not you, it's me: detecting flirting and its misperception in speed-dates. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1* (Singapore) *(EMNLP '09)*. Association for Computational Linguistics, USA, 334–342.

[20] Tun Thura Thet, Jin-Cheon Na, Christopher S.G. Khoo, and Subbaraj Shakthikumar. 2009. Sentiment analysis of movie reviews on discussion boards using a linguistic approach. In *Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion* (Hong Kong, China) *(TSA '09)*. Association for Computing Machinery, New York, NY, USA, 81–84. https://doi.org/10.1145/1651461.1651476

[21] IE University. 2023. Flirty or not. https://huggingface.co/datasets/ieuniversity/flirty_or_not

[22] Mayur Wankhade, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. 2022. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review* 55 (Feb. 2022), 5731–5780. https://doi.org/10.1007/s10462-022-10144-1

[23] Monica T. Whitty. 2004. Cyber-flirting: An examination of men's and women's flirting behaviour both offline and on the internet. *Behaviour Change* 21, 2 (June 2004), 115–126. https://doi.org/10.1375/bech.21.2.115.55423

[24] Zhao Yang. 2021. Sentiment analysis of movie reviews based on machine learning. In *2020 2nd International Workshop on Artificial Intelligence and Education* (Montreal, QC, Canada) *(WAIE 2020)*. Association for Computing Machinery, New York, NY, USA, 1–4. https://doi.org/10.1145/3447490.3447491