Power Dynamics Analysis on Gen-Z Slang conversations

Revanth, Shivam, Keshava

IIIT Hyderabad

Contents

1 Introduction

In recent years, language models and classification systems have achieved impressive performance across a variety of natural language processing (NLP) tasks. These models are typically trained on datasets composed of standard or formal English, such as news articles, books, and structured conversations. However, language is dynamic and constantly evolving, particularly in online spaces where new linguistic forms emerge rapidly. One such evolution is Gen Z slang—a style of communication that is informal, highly contextual, and often opaque to those outside the demographic. This raises an important question: How do models trained on standard English datasets handle the classification and interpretation of text written in Gen Z slang?

This paper aims to explore the power dynamics and limitations of classification models when exposed to Gen Z-style language. Specifically, we investigate several core questions:

- Does the performance of standard-trained models decrease on Gen Z slang texts?
- Can we identify the potential reasons for the poor performance on Gen Z slang texts?
- What linguistic features or phrases contribute to this performance drop? Are they rooted in stylistic differences or semantic ambiguity?

To answer these questions, we construct and utilize a Gen Z Slang Dataset. We used Different models(RNN, LSTM, and BERT) to check the difference in performance and used the LIME method to analyze the phrases that are contributing to the performance drop and change in styles.[2] [4]

2 Enron Email Dataset

2.1 Data Preprocessing

The dataset used in this study was sourced from `datasets/emails.csv`. To facilitate structured analysis, each email was divided into two main components:

- **Information Part:** Encompasses headers and metadata.
- **Content Part:** Comprises the actual email body.

2.1.1 Header Information Extraction

Key metadata fields were parsed and extracted from the header of each email. These include:

`Message-ID, Date, From, To, Subject, Cc, Bcc, Content-Type, Content-Transfer-Encoding, X-From, X-To, X-cc, X-bcc, X-Folder, X-Origin, X-FileName`

2.1.2 Content Cleaning

To improve data quality, a series of cleaning steps were applied:

- Emails with invalid encoding (e.g., `text/plain; charset=us-ascii`) were removed.
- Forwarded content was identified and separated using the delimiter "`-------------`".
- Two binary flags were added:
  - `has_other_content`: Indicates presence of additional content beyond the main body.
  - `if_forwarded`: Flags whether the email was forwarded.

2.1.3 Email Content Filtering

To maintain consistency and relevance in the dataset, the following filters were applied:

- Emails containing fewer than two words were excluded.
- Emails with header-like patterns in the content (e.g., `From:`, `To:`, `Message from:`) were removed.
- Email content was truncated to a maximum of 1000 characters.
- Emails where the sender and recipient were the same were discarded.
- Emails with null values in the recipient fields were removed.

2.1.4 Recipient Standardization

To standardize recipient information:

- A new field, `Main_To`, was created by extracting the first recipient from the `To` field.
- The `Cc` field was normalized by replacing null values with the string `None`

2.2 Label Identification

The publicly available Enron Email Dataset doesn't contain the Hierarchical Labels. There are several research papers that tried to find the Hierarchical Labels, one such method is described in[1], in which the authors have manually found the labels and stored the dataset along with labels in a MongoDB database. But this is not publicly available, we also tried to contact the authors that have access to the dataste along with labels, unfortunately we didn't get any response from the authors. So we decided to find the labels using SNA (Social Network Analysis), which was an alternaate method in[1], but with 85% as compared to the Golden hierarchy dataset.

2.3 Social Network Analysis

To infer implicit hierarchies in email communications, we conducted a Social Network Analysis (SNA) using centrality measures from graph theory. The communication dataset was modeled as a directed graph where each node represents a person, and each directed edge from a sender to a recipient (including 'To' and 'Cc' fields) denotes an email communication. The edges were parsed without interpreting direction as a strict indicator of superiority but rather to quantify the extent of communication influence.

Graph Construction: The dataset was preprocessed by filling missing 'Cc' fields with empty strings. For each email, all recipients listed in 'To' and 'Cc' were extracted, and directed edges were added from the sender to each recipient. This resulted in a directed graph constructed using NetworkX.

Centrality Metrics: Four centrality measures were computed for all nodes:

- **Degree Centrality:** Reflects how many direct connections (emails sent) a person has.
- **Betweenness Centrality:** Measures the extent to which a node lies on communication paths between other nodes, identifying brokers or intermediaries.
- **Closeness Centrality:** Quantifies how quickly a node can reach other nodes, indicating efficiency of communication.
- **PageRank:** Assigns higher scores to nodes that are connected to other highly-scored nodes, useful for identifying influential individuals.

All centrality scores were normalized using min-max scaling to bring them into a comparable range. A weighted combination of these metrics was then computed to form a unified influence score for each node. The weights were empirically assigned as follows:

- Degree: 0.10
- Betweenness: 0.20
- Closeness: 0.20
- PageRank: 0.50

Hierarchy Inference: Using the combined influence scores, hierarchical relationships were inferred between senders and primary recipients ('Main_To') of each email. The following logic was applied:

- If sender's score > recipient's score $\Rightarrow$ *Sender higher*
- If sender's score < recipient's score $\Rightarrow$ *Recipient higher*
- If scores are equal or unidentifiable $\Rightarrow$ *Similar level* or *Unknown*

Here are a few samples of email content and their corresponding Hierarchical Label :

1. - **Content**:

```
Hi Al,

My suggestion is
that you capture
the change order
with change order
#1 for PSCO's break
out contract.  If
it needs to be paid
before year end
```

```
(which it probably
does) you can put
a payment date of
December 20th, and
it will be paid out
of TurboPark. This
approach has been
discussed with Lee
Johnson at GE, and
he is fine with it.

Thanks,

Kay
```

- **Hierarchical Label**: Sender higher

2. - **Content** :

```
Attached is another
redlined draft of
the Pastoria IM.
It is much more
close to completion
because it now
includes comments
from all necessary
parties.

Please carefully
review and send
me any changes no
later than Monday.
If you send them
today (Sunday) send
them via e-mail.
Beginning Monday
(12 PM CDT) I'll
be in SF, so if you
fax them send them
to 415.782.7827.
Please call me
at 713.629.0929
or 713.398.6412
should you have any
questions. Thanks.

PS for Dave Parquet:
Not all of your
proposed changes
were made simply
because I could not
ready many of them
and the changes
were too late at
night to call you.
I was asked to get
a draft out ASAP
so I will call you
today (Sunday) to
discuss them.
```

- Hierarchial Label : Recipient higher

3. - **Content** :

```
My complete and
heart-felt condolences
go out to all my
American friends.
India shares your
grief. I only hope
that out of this
calamity, America's
wrath shall bring
retribution without
end.

Annat Jain
```

- Hierarchial Label : Similar Level

3 Gen-Z Data Generation

3.1 Using Similarity Scores

So, we mentioned above we only Enron dataset and Gen-Z slang dataset which contains only word and its corresponding.

So, we naively approached in the below way:

- Convert the meaning of the Gen-Z word into Embeddings using the

SBERT(sentence BERT) model.

- or Each Message, create the phrases using the spaCy library and check whether there are any similar semantically similar phrases to that Gen-Z word meaning Embedding. We found the similarity between phrases and Gen-Z word embedding using cosine similarity.
- After finding the similarity scores between phrases and Gen-Z word Embedding, we replaced those phrases whose cosine similarity score value is greater than some threshold with this new Gen-Z word.
- Similarly, we have to repeat this method for each message in the Enron Email Dataset.
- We synthesized a Dataset, emails cleaned modified.csv [Datasets Link (2025)], which contains modified messages infused with Gen-Z slang.

Now, let us see how the message has been changed using this method:

- 1. Actual content: "test successful. way to go!!!"

  Modified content: "test successful. way to G2G!!!"

But this is not giving the meaning full sentences and breaking the continuity in the sentences.

3.2 Using Machine translation

One of the techniques that can be used to generate Gen-Z emails is using Transformer that trains on the normal sentence and its corresponding Gen-Z form. Eventually, we can give any text as input to the transformer and get the Gen-Z form. This is a style translation problem in NLP. But the challenge here is, as mentioned above, we don't have any corpus corresponding to Gen-Z sentences. In order to mitigate this challenge, we have used LLMs to translate a given text into Gen-Z form using API calls. We have used two datasets independently for this task:

- Enron Email Dataset
- Twitter Dataset

3.2.1 LLM Sampling

To generate Gen Z-style paraphrases of textual data, we employed large language models (LLMs) via the Groq API. Our approach used a set of curated prompts designed to strictly enforce constraints such as tone preservation, no addition of emojis or hashtags, and accurate slang translation. The core objective was to produce a Gen Z slang version of each sentence while retaining its original meaning and intent.

We created a Python-based pipeline to automate the generation process using multiple API keys and a list of diverse LLM models provided by Groq, including:

- `gemma2-9b-it`
- `llama-3.3-70b-versatile`
- `llama3-8b-8192`
- `meta-llama/llama-4-scout-17b-16e-instruct`
- `qwen-qwq-32b`
- `mistral-saba-24b`
- `and others`

The dataset used for generation comprised email texts sourced from `datasets/sample_emails.csv` and text sourced from `datasets/twitter_dataset.csv`. First the email data is used and the responses were saved in `datasets/genz_emails_final.csv` and the responses correspondind to twitter data was saved in `datasets/gen_data.csv`. A subset of the data was selected and batched into groups of 15 samples for efficient API querying. For each batch, the API key and model were selected in a round-robin manner to balance the request load and avoid rate limits or overuse of a single key.

The core generation function, `Genz_generation`, initialized a Groq client and sent a tailored prompt to the selected model. It handled exceptions and fallback mechanisms: in case of failure, the next available model was tried, and if failures persisted across all models, the sampling process was halted.

Each successfully generated batch was stored in a CSV file with the original text and its corresponding Gen Z translation. The pipeline ensured:

- consistent format of output using strict prompting,
- robust handling of API errors,
- fair distribution of requests among API keys and models.

This LLM sampling procedure provided a reliable and scalable way to augment the dataset with stylistic variations suitable for downstream tasks such as style transfer, classification, or generation evaluation.

3.2.2 GenZ Style Transformer

In this project, we utilize a Transformer-based model—specifically, the T5 (Text-To-Text Transfer Transformer)—to translate formal English sentences into informal, Gen Z-style language. The task is framed as a text-to-text generation problem, where the input is a sentence in formal English and the output is its Gen Z-styled equivalent.

Preprocessing

Each sentence is first preprocessed to standardize the input format. The text is converted to lowercase, multiple whitespaces are reduced to a single space, and special characters (excluding punctuation like commas, periods, and hyphens) are removed. This normalization step ensures consistency and improves the quality of tokenization.

Dataset Splitting

The dataset is split into training and validation sets using a 90-10 split ratio. Two new columns—`original_processed` and `generated_processed`—contain the cleaned formal input and the target Gen Z style output, respectively.

Model and Tokenizer

We load the `t5-small` model and its associated tokenizer from the Hugging Face Transformers library. The model is fine-tuned for 5 epochs using the AdamW optimizer and a linear learning rate scheduler. During training, we apply teacher forcing by feeding the target sequence as input for loss computation. To avoid penalizing padded tokens, the padding token IDs in the label sequence are replaced with `-100`, which PyTorch ignores when computing the loss.

Training and Evaluation For training, evaluation inference we used two sets of datasets and they are :

- Only Genz Email data
- Genz Email data and Genz Twitter Data

The model is trained for 5 epochs with a batch size of 4. After each epoch, the model's performance is evaluated on the validation set using the average loss. The best model (based on validation loss) is saved to disk and reloaded for inference.

The training and evaluation process is

done independently for both the datasets and the corresponding models are saved as follows :

- For Email data alone (***Model-1***) : genz_t5_model.pt and similarly the tokenizer can be found in folder genz_t5_tokenizer
- Using both data (***Model-2***) : genz_t5_model_new.pt and similarly the tokenizer can be found in folder genz_t5_tokenizer_new

3.2.3 Inference

Model-1

Here are few examples of input and transformed output obtained by using transformer trained on Email and twitter data :

1. **Original Email** :
   Attached is the revised Committed Reserves Confirm with the language we discussed. Please modify the confirm templates for the Sept. wellhead deals (and those remaining from August) to reflect the revision. Let me **know** if you have any questions.

   GenZ Translated :
   the revised Committed Reserves Confirm. Please modify the confirm templates for the Sept. wellhead deals (and those remaining from August) to reflect the revision. Let me know if you have any questions.

2. **Original Email** :
   Please let me know if this will be problem.

   GenZ Translated :
   Bitte let me know if this will be problem.

3. **Original Email** :
   Dad, I was talking with Kathleen this weekend and she had some ideas and suggestions about Enron. I asked her to put them in writing so that I could share them with you.

   GenZ Translated :Dad, I was talking with Kathleen this weekend and she had some ideas and suggestions about Enron.

Model - 2

Here are few examples of input and transformed output obtained by using transformer trained on Email and twitter data :

1. **Original Email** :
   Attached is the revised Committed Reserves Confirm with the language we discussed. Please modify the confirm templates for the Sept. wellhead deals (and those remaining from August) to reflect the revision. Let me **know** if you have any questions.

   GenZ Translated :
   attached is the revised Committed Reserves Confirm with the language we discussed. please tweak the confirm templates for the Sept. wellhead deals (and those remaining from August) to reflect the revision. **hit** me up if you've got any questions.

2. **Original Email** :
   Please let me know if this will be problem.

   GenZ Translated :
   hit me up 'if this will be sus.

3. **Original Email** :
   Dad, I was talking with Kathleen this weekend and she had some ideas and suggestions about Enron. I asked her to put them in writing so that I could share them with you.

   GenZ Translated :Dad, I was vibing with Kathleen this weekend and she had some ideas and suggestions about Enron. I asked her to put them in writing so that I could share them with you.

3.2.4 Conclusion

Model-1, trained only on email data, retains a formal tone with minimal stylistic changes, resulting in outputs close to the original content. In contrast, Model-2, trained on both email and Twitter data, generates more informal and stylistically transformed outputs using GenZ slang (e.g., *"hit me up"*, *"vibing"*). This indicates that incorporating social media data enhances the model's ability to produce culturally adaptive and expressive language, making Model-2 better suited for informal or generational translation tasks.

---

4 Methods

Overview of our approach to power-relation classification in emails.

4.1 LSTM Classifier

Our classifier is a single-layer LSTM network built as follows:

- **Vocabulary & Embeddings:** Constructed from tokens appearing at least twice in the training split; reserved indices for padding and unknown tokens. Embedding dimension set to 128.
- **Model Architecture:**
  - *Embedding layer* (vocab size × 128)
  - *LSTM layer* with hidden size 128 and dropout 0.3
  - *Fully connected output* layer mapping to three classes
- **Optimization:** Trained for 5 epochs using Adam (learning rate 1e-3), batch size 64, and cross-entropy loss.
- **Evaluation:** After training on the formal (sample) emails, we assess performance on both the held-out formal test split and the entire Gen-Z–translated set, reporting accuracy, precision, recall, F1-score, and confusion matrices.

4.2 BERT

Our classifier is a fine-tuned BERT-based model built as follows:

- **Tokenizer & Input Representation:** Constructed using the `bert-base-uncased` tokenizer from HuggingFace; input text is lowercased, tokenized, truncated to 128 tokens, and padded to fixed length.
- **Model Architecture:**
  - Pre-trained BERT encoder (`bert-base-uncased`)
  - Classification head on top of [CLS] token embedding
  - Output layer mapping to three hierarchy classes (*Sender higher*, *Similar level*, *Recipient higher*)
- **Optimization:** Trained for 3 epochs using the HuggingFace `Trainer` API with AdamW optimizer (learning rate 2e-4 via `TrainingArguments`), batch size 2 (with gradient accumulation of 4 for effective batch size 8), weight decay 0.01, and cross-entropy loss.

- **Evaluation:**
  After training on the formal emails, we evaluate performance on a held-out 20% validation set, reporting accuracy and loss per epoch. Final model and tokenizer are saved for downstream use.

5 Results

5.1 Results for RNN

Overall Accuracy: 58%

Precision:

- Sender higher: 0.59 (59%)
- Similar level: 0.22 (22%)
- Recipient higher: 0.53 (53%)
- Macro average precision: 0.45 (45%)
- Weighted average precision: 0.56 (56%)

Recall:

- Sender higher: 0.90 (90%)
- Similar level: 0.09 (9%)
- Recipient higher: 0.16 (16%)
- Macro average recall: 0.38 (38%)
- Weighted average recall: 0.58 (58%)

5.2 Results for LSTM

Our single-layer LSTM generalized remarkably well across both standard and Gen-Z–infused emails. On the held-out formal test split, it achieved an overall accuracy of 78.2%, with F1-scores of 0.81 for *Sender higher*, 0.73 for *Recipient higher*, and 0.91 for *Similar level*. On the Gen-Z transformed set, performance further improved to 84.3% accuracy, delivering F1-scores of 0.87, 0.81, and 0.75 for the respective classes. These results demonstrate the model's robustness under informal, slang-laden language variations.

5.3 Results for BERT

On Normal English data:

Accuracy: 93.21%

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Recipient higher	0.9328	0.9121	0.9224	1081
Sender higher	0.9272	0.9464	0.9367	1306
Similar level	0.9823	0.9569	0.9694	116
Accuracy			**0.9321**	2503
Macro Avg	0.9475	0.9385	0.9428	2503
Weighted Avg	0.9322	0.9321	0.9320	2503

On Gen-Z Slang Data:

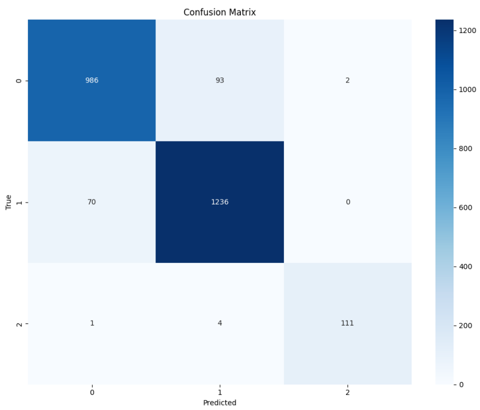
Accuracy: 85.30%

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Recipient higher	0.8571	0.7993	0.8272	1081
Sender higher	0.8401	0.8974	0.8678	1306
Similar level	0.9900	0.8534	0.9167	116
Accuracy			**0.8530**	2503
Macro Avg	0.8958	0.8500	0.8706	2503
Weighted Avg	0.8544	0.8530	0.8525	2503

6 Analysis

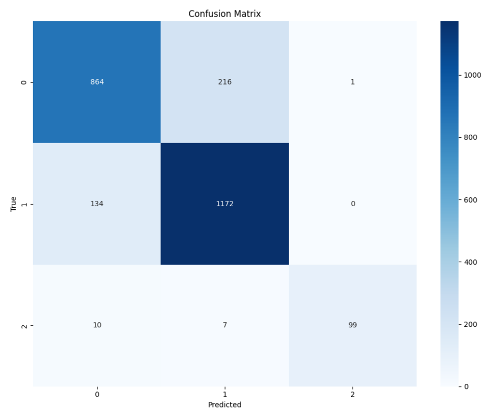
6.1 LIME

We analyzed how GenZ slang affects the determination of hierarchical order using the LIME library. Below are a few examples comparing the original and GenZ-translated versions.



[b]

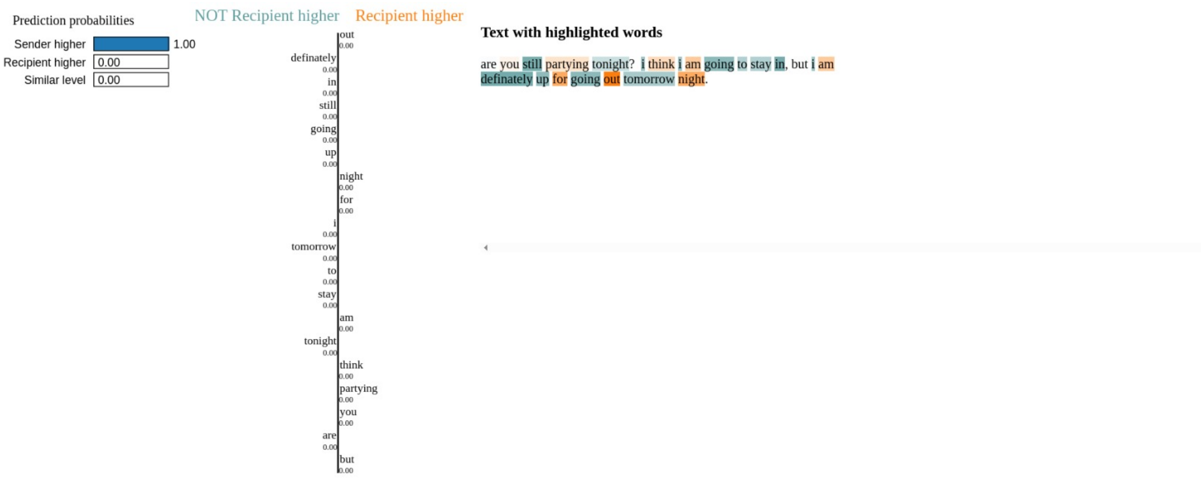
Figure 1: Evaluation on Normal Emails



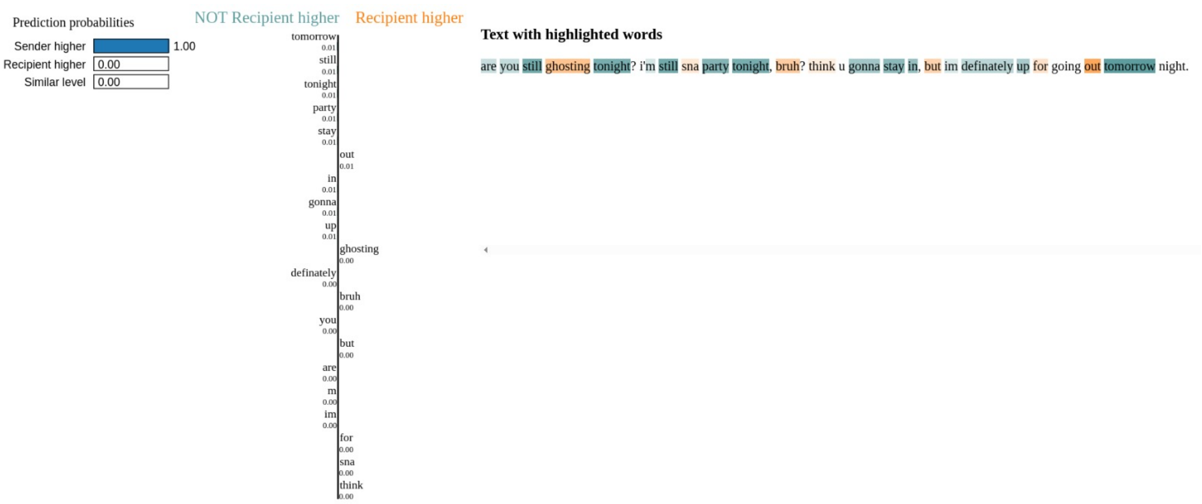
[b]

Figure 2: Evaluation on Gen-Z Emails

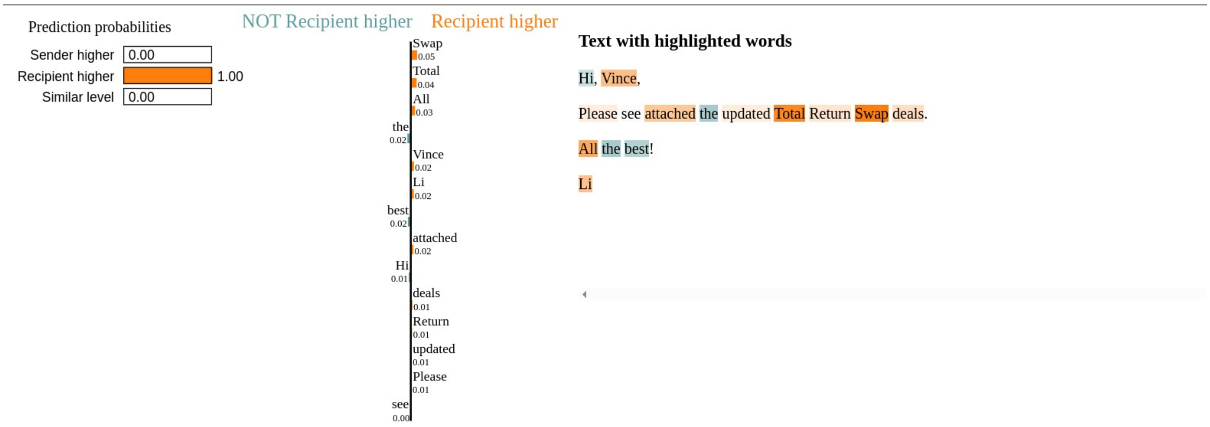
Figure 3: BERT performance comparison across Normal and Gen-Z email datasets.



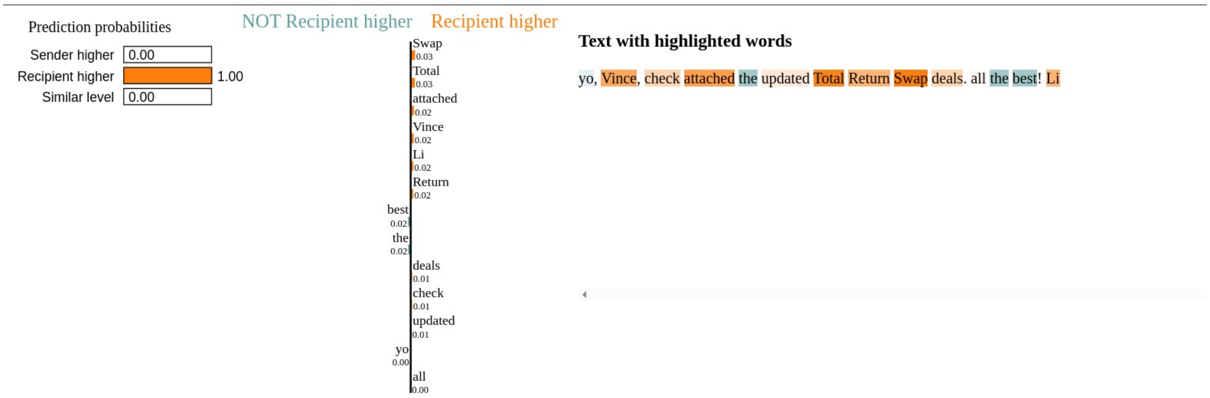
LIME Analysis: Original Sample-1



LIME Analysis: Generated Sample-1



LIME Analysis: Original Sample-2



LIME Analysis: Generated Sample-2

6.2 Phrase Extraction

We tried to extract the phrases from the datasets by giving scores to each phrase. If the score is positive for a phrase, then it mean that phrase was frequently used by the Senior. If the score for that phrase is negative, then it means that phrase was regularly used by Subordinates. We have the scores to each phrase based on the method described in[3]

1. **Words Used by Senior vs Subordinate in standard dataset:**

Senior Words		**Subordinate Words**	
corp legal department	2.86	click here	-1.91
department smith	2.84	user	-1.77
department smith street	2.84	the report	-1.77
legal department smith	2.84	this request	-1.70
thanks ben	2.73	have received this	-1.66
ebiz	2.61	click on	-1.58
this distribution	2.61	is now	-1.55
veronica	2.48	received this	-1.55
america corp legal	2.46	password	-1.51
corp legal	2.46	available for	-1.50
best jeff	2.44	act	-1.49
veronica espinoza	2.41	tana	-1.48
espinoza at	2.33	delivery	-1.46
veronica espinoza at	2.33	click	-1.44
couple of	2.24	hpl	-1.43
regards delainey	2.24	create	-1.43
report to	2.24	all of the	-1.41
thomas	2.24	and other	-1.41
thanks kay	2.19	below is	-1.41

1. **Words Used by Senior vs Subordinate in GenZ slang dataset:**

Senior Words		Subordinate Words	
department smith	2.83	delivery	-1.91
legal department smith	2.83	total	-1.84
corp legal department	2.80	account	-1.71
department smith street	2.80	create	-1.59
thanks ben	2.54	and take	-1.56
wholesale services	2.47	live on	-1.56
dperlinenroroncom	2.43	this request	-1.56
dperlinenroroncom phone	2.43	questions hit me	-1.53
dperlinenroroncom phone fax	2.43	currently	-1.51
houston texas dperlinenroroncom	2.43	hit the	-1.51
texas dperlinenroroncom	2.43	password	-1.47
texas dperlinenroroncom phone	2.43	requested	-1.44
america corp legal	2.40	approval	-1.42
corp legal	2.40	this email	-1.42
best jeff	2.40	central time	-1.42
enron wholesale	2.31	this message	-1.40
enron wholesale services	2.31	click	-1.40
form of	2.23	staff	-1.39
peeps	2.23	status	-1.39
eba	2.13	intended	-1.37
eba houston	2.13	hit up the	-1.33
get copy	2.13	on it	-1.33
		inc	-1.27
		during the	-1.27
		lowkey trying	-1.27
		lowkey trying to	-1.27

But this method did not give any useful insights. It displayed some names also in some of the phrases. We have to remove some more words from the dataset before finding scores.

6.3 Observations

- **Sample 1:**
  1. **True Label:** Sender Higher
  2. LIME predicts *Sender Higher* for both original and generated texts.
  3. In the generated version, the term *ghosting* contributes positively toward the *Receiver Higher* class, indicating a potential shift in interpretability introduced by GenZ slang.
- **Sample 2:**
  1. **True Label:** Receiver Higher
  2. LIME predicts *Receiver Higher* for both original and generated texts.
  3. However, in the generated text, the word *yo* has a negative weight toward the *Sender Higher* class, highlighting how slang can subtly influence model explanations.

7 Conclusion

Our findings indicate that standard-trained language models struggle with Gen Z slang due to a combination of domain shift, semantic ambiguity, and stylistic divergence. Many slang terms (e.g., *idk*, *sus*) are absent from training data, leading to vocabulary mismatch. Additionally, figurative expressions like *"He ghosted me"* or *"That's a big W"* often defy literal interpretation, posing challenges for models not fine-tuned on such language. Finally, informal syntax, abbreviations, and cultural references introduce stylistic variance that further reduces model performance. Addressing these issues requires continued adaptation of models to contemporary and informal language use.

Through our experiments, we observed some differences in the types of phrases used in Gen Z emails compared to standard or professional emails. These differences significantly affect the overall style and tone of the text. Gen Z messages often include informal expressions, abbreviations, and culturally specific slang, which create a more casual and conversational tone. In contrast, standard emails tend to use formal, structured language. This shift in linguistic style leads classifiers, especially those trained on conventional data—to misinterpret the intent or content of Gen Z messages, resulting in misclassifications. Our findings suggest that these stylistic and tonal variations are a key factor contributing to the performance drop in models not exposed to such language during training.

References

[1] Apoorv Agarwal, Adinoyi Omuya, Aaron Harnly, and Owen Rambow. A comprehensive gold standard for the Enron organizational hierarchy. In Haizhou Li, Chin-Yew Lin, Miles Osborne, Gary Geunbae Lee, and Jong C. Park, editors, *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 161–165, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[2] Philip Bramsen, Martha Escobar-Molano, Ami Patel, and Rafael Alonso. Extracting social power relationships from natural language. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 773–782, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[3] Eric Gilbert. Phrases that signal workplace hierarchy. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, page 1037–1046, New York, NY, USA, 2012. Association for Computing Machinery.

[4] Vinodkumar Prabhakaran Gourinivas. Social power in interactions: Computational analysis and detection of power relations. 2015.