Impact of Textual Noise on Sentiment Detection

Pouria Zarrehparvar
Pouria.zarhparvar@ut.ac.ir

1 Problem statement

The goal of this project is to investigate the impact of different types of noise in textual data on sentiment detection using large language models (LLMs) in the context of semantic communication. Specifically, we focus on the introduction of special characters and Homoglyph (Unicode Character Look-Alikes) as noise. This study aims to enhance our understanding of how noisy text can influence the semantic accuracy and quality of system. my motivation stems from the increasing prevalence of noisy data in real-world applications, such as social media, and the need for robust models that can handle such imperfections.

2 What you proposed vs. what you accomplished

- ~~Collect and preprocess dataset~~
- ~~Noisy channel modeling and adding noise to data~~
- ~~Build model using pre-trained models (specific baseline model) on collected dataset and examine its performance~~
- *use text-to-image models to find how textual noise can affects the perfomance of text-to-image LLMs* : I failed to do this because I need more time.
- *Make fancy model perform better than baseline model*: I failed to do this because I need more time.
- ~~Perform in-depth error analysis to figure out what kinds of examples our approach struggles with~~

3 Related work

Understanding and mitigating the impact of noisy text on sentiment detection and text-to-image generation has evolved significantly over the years. This project builds on a rich history of research and development in both natural language processing (NLP) and multimodal learning.

foundational work in sentiment analysis, where Go et al. (2009) introduced the Sentiment140 dataset, a pivotal resource comprising 1.6 million tweets annotated for sentiment. This dataset has since been a cornerstone for numerous studies in sentiment detection (Go et al., 2009).

In parallel, the advent of deep learning revolutionized NLP, leading to the development of powerful models capable of understanding and generating text. One notable advancement was the creation of CLIP by Radford et al. (2021), a model that learns visual concepts from natural language descriptions, bridging the gap between text and images (Radford et al., 2021). This was a significant step towards integrating text and visual data, yet it primarily focused on clean text inputs.

As the volume of user-generated content on social media grew, researchers began to encounter the challenge of noisy text. Pruthi et al. (2019) explored the robustness of neural networks to typographical errors, highlighting the vulnerability of models to even minor text perturbations (Pruthi et al., 2019). This study underscored the need for models that can withstand noisy inputs, a theme central to our current project.

Building on this, Belinkov and Bisk (2018) conducted a comprehensive survey on improving the robustness of NLP models to noise, proposing various strategies for handling textual imperfections (Belinkov and Bisk, 2018). Their work laid the groundwork for subsequent research aimed at enhancing model resilience.

In the realm of text normalization, Li et al. (2019) presented methods for normalizing noisy text using sequence-to-sequence models, demonstrating how preprocessing can mitigate the adverse effects of noise (Li et al., 2019). Our project extends this approach by injecting specific types of noise, such as special characters and Homoglyphs, into the Sentiment140 dataset to study their impact on downstream tasks.

The challenge of integrating noisy text into multimodal models was further explored by Zhang et al. (2020), who discussed Homoglyph attacks in cybersecurity. Their insights into how similar-looking characters can deceive systems have informed our approach to evaluating text-to-image models (Zhang et al., 2020).

Moreover, character-level convolutional networks for text classification, as proposed by Zhang et al. (2015), offer another perspective on handling noisy data. Their work on character-level models highlights the potential for these architectures to manage textual noise effectively (Zhang et al., 2015).

Our project also draws inspiration from the work of Lu et al. (2019) on multimodal transformers for image captioning and visual question answering (VQA). They demonstrated the power of transformers in handling multimodal data, a principle we apply to study the effect of noisy text on text-to-image generation (Lu et al., 2019).

Finally, the paper by Goodfellow et al. (2015) on adversarial examples in deep learning provides a theoretical framework for understanding how small perturbations can drastically affect model outputs. This concept is closely related to our study of how noise influences sentiment detection and image generation (Goodfellow et al., 2015).

Together, these studies form the historical backdrop of our project, illustrating the evolution of techniques to handle noisy text and their application in sentiment detection and multimodal learning. Our contribution lies in systematically injecting noise into textual data and evaluating its effects on state-of-the-art models, thereby advancing the field of robust NLP and multimodal communication.

4 Your dataset

The primary dataset used in this study is the Sentiment140 dataset, a well-established resource for sentiment analysis in the field of natural language processing (NLP). This dataset was introduced by Go et al. (2009) and consists of 1.6 million tweets, each labeled for sentiment. The labels are categorized into positive, negative, and neutral sentiments. The tweets were collected using the Twitter API, and sentiment labels were automatically assigned based on emoticons present in the tweets. This large-scale dataset provides a diverse and extensive collection of real-world textual data, making it ideal for training and evaluating sentiment detection models.

One of the key challenges of working with the Sentiment140 dataset is its inherent variability. Tweets, by nature, are often informal and may include slang, abbreviations, and other forms of non-standard language. This variability can complicate the task of sentiment detection, as models must be robust enough to understand and process these irregularities. Additionally, the dataset contains a significant amount of noise in the form of special characters, emoticons, and other non-alphabetic symbols, which can further challenge the accuracy of sentiment classification models.

In this project, we extend the complexity of the Sentiment140 dataset by intentionally injecting specific types of noise, such as special characters and Homoglyphs (Unicode Character Look-Alikes). For example, the phrase ”I love this product!” might be altered to ”I l0v3 th!$ pr0duct!” to simulate the impact of noisy text. This approach allows us to systematically study how different types of noise affect sentiment detection performance and how robust the models are to such perturbations.

The Sentiment140 dataset includes various statistical properties that are pertinent to our task. The dataset contains approximately 10 million words distributed across 1.6 million tweets. Each tweet, on average, consists of around 10-15 words, although there is significant variation in tweet length. The dataset's large size and the diversity of its content provide a comprehensive foundation for training deep learning models.

Below are a couple of examples of input and output pairs from the dataset:

- **Original Tweet:** ”I love this product!”
  - **Noisy Tweet:** ”I l0v3 th!$ pr0duct!”
  - **Sentiment Label:** Positive

- **Original Tweet:** ”This is the worst experience ever.”

  - **Noisy Tweet:** "Th!$!s th3 w0rst exp3ri3nce 3v3r."
  - **Sentiment Label:** Negative

These examples illustrate how noise is introduced into the dataset and the corresponding sentiment labels. The task involves training models to accurately classify the sentiment of tweets, even when they contain significant noise.

The Sentiment140 dataset is a crucial component of this study, as it provides a realistic and challenging environment for evaluating the robustness of sentiment detection models. By systematically injecting noise and analyzing the impact on model performance, we aim to gain deeper insights into the capabilities and limitations of current NLP techniques in handling noisy data.

4.1 Data preprocessing

In the data preprocessing phase, we focused on preparing the Sentiment140 dataset for our experiments. This involved several steps, starting with the tokenization of tweets to split the text into individual words or tokens. We then injected noise into the dataset by introducing special characters and Homoglyphs (Unicode Character Look-Alikes) into the tweets. This was done to simulate the kinds of noise commonly found in real-world text data, especially on social media platforms. Additionally, we applied normalization techniques to convert text into a standard format, which involved lowercasing all characters and removing unnecessary whitespace. These preprocessing steps are crucial to ensure that the dataset accurately represents the noisy conditions under which our models will be evaluated, thereby enabling a thorough assessment of their robustness.

5 Baselines

For our baseline models, we selected two primary approaches to serve as reference points for evaluating the impact of noise on sentiment detection. The first baseline is a sentiment analysis model trained on the original, clean Sentiment140 dataset. This model provides a performance benchmark in the absence of noise. The second baseline is a similar sentiment analysis model trained on the noisy version of the Sentiment140 dataset, where special characters and Homoglyphs have been introduced. By comparing the performance of these two models, we can quantify the degradation in accuracy and other performance metrics due to the presence of noise. These baselines were chosen to highlight the differences between clean and noisy data conditions and to serve as a foundation for further improvements and noise mitigation strategies.

6 My approach

Our approach involves systematically injecting noise into the Sentiment140 dataset and analyzing its impact on sentiment detection and text-to-image generation models. We used Google Colab with free GPUs to perform our experiments, leveraging libraries such as Hugging Face Transformers and PyTorch for model implementation. The sentiment analysis model was fine-tuned on both the clean and noisy versions of the dataset to compare performance. For text-to-image generation, we utilized the CLIP model to assess how noisy text inputs affect image generation quality. The primary challenge we faced was managing computational resources within the constraints of Colab's environment, which required optimizing code efficiency and managing memory usage effectively. Additionally, implementing noise injection without losing the semantic meaning of text was crucial, ensuring that the injected noise accurately represented real-world scenarios while still being interpretable by the models.

7 Error analysis

It can be seen that if there is noise in the data, the noises that are of the type of adding an additional character to the text are less likely to reduce the accuracy of large models, and most of all the noises of the type of adding a random character instead of one of the characters cause reduce the accuracy of the model, and the noise of the displacement of one of the letters with its similar equivalent is placed between the two mentioned types of noise in terms of reducing the accuracy of the model. Somehow, in the first two methods, the meaning of the words can be recovered to a good extent depending on the noise rate. Randomly changing characters can significantly alter words, making them unrecognizable and disrupting context.

8 Conclusion

The project investigates the impact of various types of textual noise on sentiment detection using bertbase pre-trained model and The three types

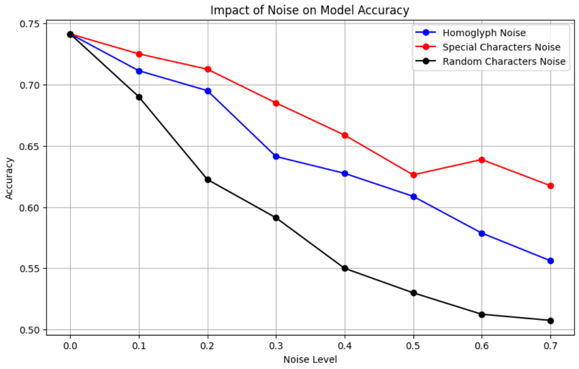

Figure 1: Bertweet Accuracy with different noise rate and types of noise.

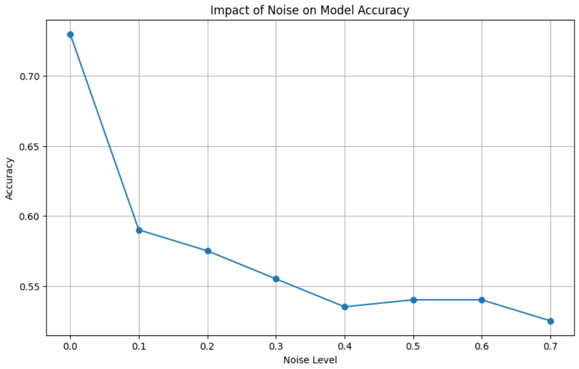

Figure 2: distilled Bert without fine-tune Accuracy with different noise rate .

of noise considered are: 1-Adding special characters to the text. 2 - Randomly changing characters in the text. 3-Replacing characters with similar-looking characters (homoglyphs). Adding special characters to text has the least impact on model accuracy. The model can recover the meaning of the words to a good extent, depending on the noise level. Replacing characters with similar-looking ones falls between the other two types in terms of impact on accuracy. While the meaning of words can often be recovered, it is less effective than dealing with special characters noise but more effective than random characters noise.

References

Belinkov, Y. and Bisk, Y. (2018). Synthetic and natural noise both break neural machine translation. In *Proceedings of the 7th International Conference on Learning Representations*.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples. *International Conference on Learning Representations*.

Li, J., Gong, D., Peng, H., Meng, Y., Xu, J., and Wu, F. (2019). Neural text normalization with sequence-to-sequence models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

Lu, J., Batra, D., Parikh, D., and Lee, S. (2019). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*.

Pruthi, D., Dhingra, B., and Lipton, Z. C. (2019). Combating adversarial misspellings with robust word recognition. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*.

Zhang, T., Liu, K., Liu, P., and Zhang, J. (2020). Hotstuff: Homoglyph-based cyber attack defense for context-free grammar based languages. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems*.