Progress Report of the Free Topic Project

Yuqin Zhou
University of Copenhagen
qvk729@alumni.ku.dk

Abstract

This progress report provides an update on the Free Topic project, which aims to distill knowledge from datasets by modifying training instances while preserving their dimensionality and quantity. Specifically, the project seeks to find a theoretical framework for explaining how various modification strategies impact a model's performance when applied to datasets. During the experimental evaluation, reproducibility issues are encountered with some multimodal models, and the experimental design for stop words deletion needs to be revised. Theoretical analysis of the experimental results is conducted from two perspectives: long-range word correlation and word distribution. However, the use of long-range correlation requires longer training sentence lengths, and the word distribution assumes that the dataset follows Zipf's law, which is not observed in the chosen datasets (CMU-MOSI and MUSTARD). One potential direction is to continue investigating long-range correlation. To pursue this, a dataset with longer sentences for training the model would need to be identified, as well as a benchmark dataset to test the models trained on modified data and assess their ability to capture long-range correlation.

1 Background and Motivation

State-of-the-art neural network models often have vast amounts of parameters and require numerous training data, leading to the need for more efficient learning. Two directions have been explored to address this issue: *model distillation* and *dataset distillation*. Model distillation transfers the knowledge of a large model into a simpler one in order to improve the performance and/or efficiency of the smaller model (Hinton et al., 2015) . The alternative, dataset distillation, keeps the model architecture unchanged and instead attempt to distill the knowledge from a large training dataset into a small one (Wang et al., 2018).

To the best of my knowledge, dataset distillation is the closest related research area for this project and I would like to briefly outline its background. Currently, the mainstream method is to transform a real-world dataset into a synthetic dataset. The standard procedure is to take a large real dataset as input, and outputs a small synthetic, distilled dataset, and evaluates the synthetic method via testing models trained on this distilled dataset and tested on a separate real dataset. Techniques like label distillation (Bohdal et al., 2020), optimization (Nguyen et al., 2021), gradient Matching (Zhao et al., 2020), and parametrization (Kim et al., 2022) have been proposed for synthesizing data. These techniques have been applied in various fields, like recommender systems (Sachdeva et al., 2022) and federated earning (Goetz and Tewari, 2020). This type of work often performs a low-dimensional projection on instances to reduce their number of dimensions.

In addition to synthesizing data, dataset distillation can also be achieved through pruning unimportant instances, which is often referred as *core-set construction* (Tsang et al., 2005; Bachem et al., 2017), and *instance selection* (Olvera-López et al., 2010). Their work aim to choose a subset of the entire training data so that models trained on the subset perform similarly to models trained on the full dataset. An example of building the core set is Perception (Rosenblatt, 1957) algorithm, which can be viewed as the weighted sums of subsets of training examples (Wang et al., 2018).

To distill dataset, we can find that the previous research has attempted to synthesize instances or select valuable ones. However, little exploration has been done to "simplify" individual instances while preserving their dimensionality and quantity. For example, in the context of sentiment analysis, removing unimportant words or those that may mislead the model's judgment should also be viewed as one type of dataset distillation. To formalize

this idea, I will outline the research objectives and methodology in Sec. 2.

2 Research Goal and Methodology

The research goal of this project is to conduct **empirical** and **theoretical** research to study the impact of modifying instances on model performance. The project is divided into two phases, as outlined below.

2.1 Phase A: Experimental analysis

In this phase, I have conducted experiments to remove words from training instances selecting by different strategies, to observe changes in model performance.

It is important to note that this approach differs from previous research as it does not reduce the number of training instances or their dimension. Also, the selection and removal of words occurs during training, not validation or test, as I want to see if the model performance can be improved by eliminating unnecessary information during training. Modifying the validation or test set is often viewed as introducing noise and testing models' robustness (Liang et al., 2021), which is not relevant to our research objective.

2.1.1 Deleting strategies

We employs different strategies to select words during training and compare their impact on the model performance on test sets.

Randomly deleting all words. The first strategy involves randomly deleting all words from sentences, with each word having a probability of $p \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ of being deleted. This probability value, which I refer to as the "modification level". The model performance is expected to gradually worse with higher modification levels.

Randomly deleting stop words. Our second strategy is to delete *stop words*, such as "the", from sentences with different modification levels. The model's performance is expected to slightly improve with increasing modification levels, as all stop words containing little information have been deleted.

Techniques in information retrieval. Our third strategy is query expansion techniques used in information retrieval such as KL divergence (Plachouras et al., 2004). For example, given a sequence of N words $X = [x^{(1)}, x^{(2)}, \cdots, x^{(N)}]$, we can use KL divergence to assign a weight to each word in its sentence as follows :

$$w(x^{(t)}) = P_X \log_2 \frac{P_X}{P_C} \tag{1}$$

where $P_X = \frac{Count_X(x^{(t)})}{len(X)}$, $P_C = \frac{Count(x^{(t)})}{N}$, $Count_X(x^{(t)})$ is the frequency of $x^{(t)}$ in the sentence X, $len(X)$ is the number of words in X, $Count(x^{(t)})$ is the frequency of $x^{(t)}$ in the datasets, and N is the number of words in the dataset. We can find that KL divergence measures the difference between the distribution of a word in a sentence vs the entire dataset.

Words that are more unique to a sentence receive a higher weight. Then, I can remove words by sorting their weights from lowest to highest score. The model's performance is expected to improve with low modification levels, before gradually worsening as the modification level increases.

2.1.2 Multimodal datasets

In addition to experimenting with different deletion strategies, a second approach that enriched our research is the use of multimodal datasets. This allows us to examine the performance of the model with different types of information. For instance, I could modify the text modality while keeping the visual and acoustic modalities intact, then run experiments using only the text modality to compare the results. This helped us to understand how the model behaves with or without limited information (i.e. just visual and acoustic).

The experiment details and results are provided in Sec. 3. Before discussing the results, we'll introduce our plan for identifying theoretical frameworks to explain them.

2.2 Phase B: Theoretical analysis

Our second research goal is to find a proper theoretical framework to explain the influence of the modification levels using different strategies. Two mathematical tools have been explored: Zipf'law and long-range correlation (Tanaka-Ishii).

2.2.1 Zipf's law

The frequency distribution of words follows a simple mathematical form known as Zipf's law (Piantadosi, 2014). It states that the frequency of a word is inversely proportional to its rank, resulting in a power law distribution. This law is known to hold

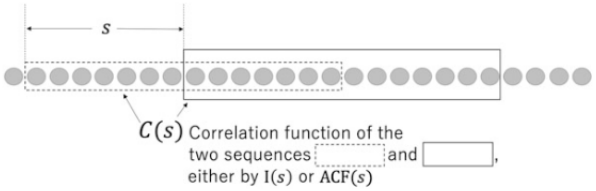


Figure 1: Schematic illustration of long-range correlation analysis (Tanaka-Ishii). $C(s)$ is a function to measure the correlation between two sequences. $I(s)$ or $ACF(s)$ denotes the *mutual information* or the *autocorrelation function* between two sequences separated by a distance s.

true for human language, leading me to question if I calculate the fitness score between the word distribution in a modfied dataset and the Zipf's law.

This fitness score shall reflect the deviation of the word distribution from Zipf's law, and is expected to decrease with increasing modification levels (assuming a score of 1 being a perfect fit). For example, if I randomly delete all words in a sentence with increasing modification levels, the modified sentences shall be gradually disrupted and not follows Zipf's law when the modification reaching at a certain level.

Since human language follows Zipf's law, I assume that the fitness score is a metric that can reflects the deviation of the word distribution after modification and natural human language. Therefore, if there is a strong correlation between the fitness score and test performance, it suggests that the model requires natural language training data. However, if no such relationship exists, it implies that the model is capable of binary classification regardless of the meaningfulness and linguistic structure of its training examples. Thus, successful results from this experiment would be an intriguing discovery, providing insight into how neural network models learn from training data.

2.2.2 Long-range correlation

From a holistic persepctive, Zipf's law offers a mathematical framework for analyzing word distribution patterns. Another approach is to focus on word dependencies in sentences. One research area to reveal word dependencies is known as *long-range correlation* (Tanaka-Ishii).

The Long-range correlation analysis, as depicted in Fig. 1, examines the correlation change between two sequences of text with respect to the distance s (Tanaka-Ishii). The correlation is measured using a function denoted as $C(s)$, which can be either the mutual information formula or the autocorrelation function. For instance, if mutual information is used, the formula is defined as follows

$$\mathrm{I}(s) \equiv \sum_{a,b} P\left(X_i = a, X_{i+s} = b\right) \log \frac{P\left(X_i = a, X_{i+s} = b\right)}{P\left(X_i = a\right) P\left(X_{i+s} = b\right)} \tag{2}$$

where X_i and X_{i+s} are words or characters separated by a distance s, and a and b indicate a particular word type or character. Using this function to measure a text has some correlation, the value of $C(s)$ follows a power function with respect to the distance s between two of its sequences:

$$C(s) \propto s^{-\gamma}, s > 0, 0 < \gamma < 1, \tag{3}$$

where $-\gamma$ is the power exponent indicating the degree of decay of $C(s)$ with respect to s. As words in a sentence are deleted at increasing levels, the resulting training instances shall become randomly generated text, lacking long-range correlation. There must be a critical point beyond which the dataset no longer exhibits this correlation, which can be determined by observing if $C(s)$ follows a power law. This is similar to the critical behaviour of language which have been extensively discussed (Ebeling and Pöschel, 1994; Ebeling and Neiman, 1995; Altmann et al., 2012; Mora and Bialek, 2011).

Based on these deductions, I believe that the critical point may provide insight into the change in a model's performance on a test set. This could be helpful to decide if the model is susceptible to long-range correlation in the training instances.

3 Experimental evaluation

I shall here introduce my attempts to complete my empirical research goal as discussed in Sec. 2.1. I have conducted approximately 3000 experiments, which can be accessed at https://wandb.ai/yuqinzhou/Free_1.2%20MOSI/table?workspace=user- and https://wandb.ai/yuqinzhou/Free_2.0%20MOSI?workspace=user-. Due to the substantial number of experiments, I will only present a selection of representative results for discussion.

3.1 Experiments settings

3.1.1 Datasets, Tasks, and Metric.

I utilize two multimodal datasets, CMU-MOSI (Zadeh et al., 2016) and MUSTARD (Castro et al., 2019) , which incorporate three modalities: *language (ℓ)*, *video (v)*, and *audio (a)*. A comprehensive description of the datasets can be found in Appendix C.2.1 of Liang et al. (2021). In this text, I present some observations from our experience with the two datasets.

Number of instances. The two datasets were chosen because of their small number of instances and the limitations of my computational resources. However, when the number of instances is too small (such as the 412 training instances in MUSTARD), the results tend to be unstable and difficult to extract meaningful information, as demonstrated in Fig. 2. CMU-MOSI has a larger number of training instances (1284), which results in more stable outcomes. Therefore, in the following sections, I will only present experiments utilizing CMU-MOSI for discussion.

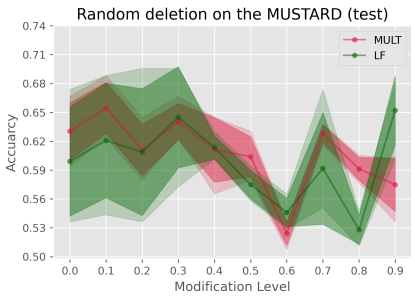


Figure 2: Test performance w.r.t modification level on the MUSTARD datasets. The solid line represents the mean over 3 experiments , the dark color area corresponds to ± 1 SEM, and the light color corresponds to the maximum and minimum values. LF stands for Late fusion using GRU, and MULT stands for Multimodal Transformer, which I shall provide a detailed explanation in 3.2.

Dimensions of modalities in CMU-MOSI. CMU-MOSI have two versions available: the raw version, where the dimensions of each modality are as follows: $d_\ell = 300$, $d_v = 35$, $d_a = 74$, and the processed version, which has dimensions of $d_\ell = 300$, $d_v = 20$, $d_a = 5$. According to Liang et al. (2021), they use the raw version in their paper, but their code appears to use the processed version. When I first run experiments with the raw version, most models' performance are worse compared to Table 13 in (Liang et al., 2021). However, when I conduct experiments with the processed version, most models improved their performance and matched the results from the original paper. As a result, in all the following experiments, I will only use the processed version of CMU-MOSI.

Tasks and Metric. At the test stage, both CMU-MOSI and MUSTARD are binary classification tasks, with MUSTARD focusing on sarcasm detection and CMU-MOSI on sentiment analysis. The evaluation metric used during the test stage is the standard accuracy metric. It's worth noting that during the training stage, the label of CMU-MOSI is a continuous variable, annotated on a scale ranging from -3 to $+3$ and averaged across multiple annotators. However, the impact of using different metrics between training and testing on our experiments is unclear.

3.2 Unimodal Models, Fusing paradigms, and Hyperparameters

Unimodal Models and Fusing paradigms. I have tested two approaches. **(1)** I employ Gated Recurrent Units (GRUs) (Chung et al., 2014) or Transformer (Vaswani et al., 2017) as the unimodal model. To fuse the information from different modalities, I employ the *Late Fusion* (LF) approach, which concatenates the representations from each modality after they have been processed by the unimodal model and feeds the result into a multi-layer perception (Bengio et al., 2003). **(2)** In addition to the LF approach, I also utilize the state-of-the-art multimodal method, *Multimodal Transformer* (MULT) (Tsai et al., 2019), which employs a Crossmodal Transformer block to generate a unified multimodal representation.

Hyperparameters. To ensure that each approach is evaluated under the same training conditions at every modification level, I perform a hyperparameter search, as shown in Table 1. All experiments are repeated 10 times and a mean and Standard Error of Mean (SEM) is computed.

3.3 Experiment results

I will now present the experiments results and the problems encountered.

3.3.1 Reproducibility issues

When doing the experiments, my first goal is to make the re-implemented models perform similarly to the previous work (Liang et al., 2021).

Component	Model	Parameter	Value
Unimodal models	GRU	Hidden sizes	[64, 128, 512, 1024]
		Num of layers	2
		Dropout	[0.1, 0.2]
	Transformer	Hidden sizes	[20, 30, 40, 50]
		Num heads	5
		Dropout	0.2
Head	MLP	Hidden sizes	[5, 20, 32, 64, 128, 256]
		Num of layers	2
		Dropout	0.2
Fusion	MULT	Hidden size	[30, 40, 50]
		Num heads	[6, 8, 10]
Training		Loss	MAE or Cross Entropy
		Num epochs	100
		Early stop	True
		Activation	ReLU
		Optimizer	AdamW
		Weight Decay	1×10^{-2}
		Learning rate	1×10^{-4}

Table 1: Table of hyperparameters for prediction on the CMU-MOSI and MUSTARD dataset. The `loss` function is set to cross entropy for the MUSTARD and Mean Absolute Error (MAE) for the CMU-MOSI. `Early stopping` is applied if valid performance does not improve over 15 epochs.

As shown in Fig. 3, the test performance of the LF (ℓ, v, a) and the best unimodal (ℓ) appears to be slightly better than the results presented by Liang et al. (2021) when the modification level is 0. As reported in their Table 13, the LF-GRU is shown to be 75.2 ± 0.8, while the Unimodal (ℓ) is reported as 74.2 ± 0.5. This result is expected, as I used the same code as Liang et al. (2021) but with a wider hyperparameter search.

However, the test performance of other models, including MULT and the best unimodal models (v and a), is significantly worse compared to previous research. More specifically, the best unimodal (v) achievs an accuracy of approximately 58%, as shown in Fig. 3, and the MULT (ℓ, v, a) achieves an average accuracy of 76.4% [1]. The performance of MULT re-implemented by me is similar to LF, while the accuracy of MULT should be around 83.0 ± 0.1 as shown in Table 13 of Liang et al. (2021).

This reproducibility issue presents a major challenge for me in interpreting the experiment results. The MULT performs the best on the CMU-MOSI according to Liang et al. (2021), but when I re-implement it without any modifications to the code, its performance is similar to that of LF, which is one of the simplest fusion paradigms. Thus, it is crucial to resolve this issue to make the experimental results meaningful. I have tried several approaches to resolve the issue, including performing a hyperparameter search as depicted in Table 1, adopting the hyperparameter settings used in Liang et al. (2021), and experimenting with various random seeds, which has a significant impact on the test results in our experiment. Unfortunately, none of these efforts have been successful in achieving 80% accuracy for MULT, and the underlying cause of the problem remains uncertain.

A possible explanation for the problem is that I have only used the code provided by Liang et al. (2021)[2] and not the original paper (Tsai et al., 2019). The implementation of Liang et al. (2021) may have led to errors in the implementation of MULT that I have not yet discovered. Another weird observation is that when running unimodel (ℓ), most runs have an accuracy rate of 75%, while 2 in 10 runs have a significantly lower accuracy of 42.5% despite being trained with the same settings and using different random seeds.

[1] The runs results are named as MULT_LVA_RD_0 and can be find in https://wandb.ai/yuqinzhou/Free_1.2%20MOSI/table?workspace=user-

[2] The code can be found at https://github.com/pliang279/MultiBench.

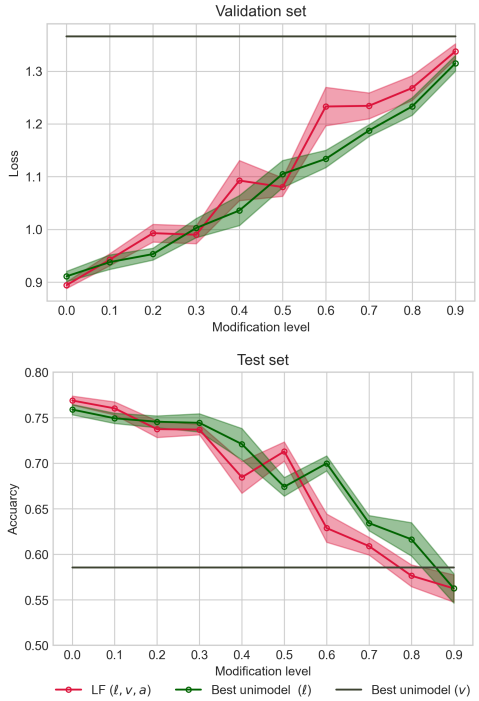


Figure 3: Model performance w.r.t modification level using randomly deletion of all words on the CMU-MOSI datasets . The solid line represents the mean over 10 experiments, the dark color area corresponds to ± 1 SEM. LF stands for Late fusion using GRU.

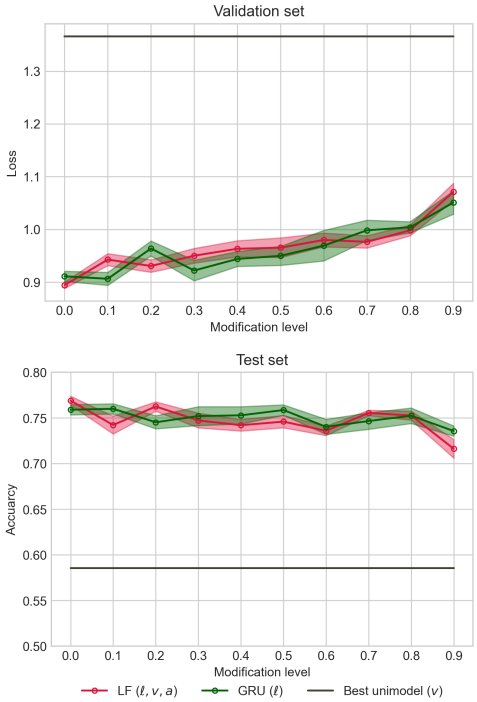


Figure 4: Model performance w.r.t modification level using stop words deletion on the CMU-MOSI datasets . The solid line represents the mean over 10 experiments, the dark color area corresponds to ± 1 SEM. LF stands for Late fusion using GRU.

3.3.2 Improper experiments design

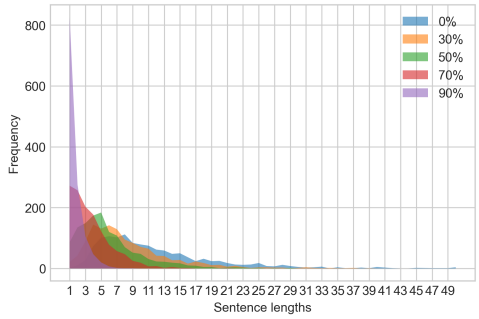
In accordance with the methodology outlined in Section 2.1.1, I have carried out experiments involving the random deletion of all words or stop words as demonstrated in Figures 3 and 4.

As illustrated in Fig. 4, the test accuracy does not decrease with increasing modification level, unlike in Figure 3. This can be attributed to the fact that stop words make up only a small part of each sentence, and even when almost all stop words are removed (e.g., modification level = 0.9), many instances can still preserve their syntactic or pragmatic structure. By comparing Fig. 5a and Fig. 5b, we can observe a significant difference in sentence length when randomly deleting all words versus deleting only stop words.

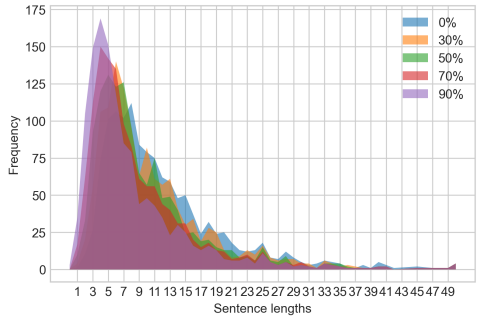
Additionally, the training set for CMU-MOSI comprises 1284 instances, while the number of aligned sentences is 982. "Aligned" refers to sentences with the same number of words and word embeddings after pre-processing, which is crucial as the processed data lack text information. To delete stop words, I have to randomly pick words from the raw data (containing text information) and delete the corresponding word embeddings in the processed data. As a result, I am only able to remove stop words from these aligned sentences, which constitute 76.4% of the total.

It is important to note that our research objective is to explore the impact of modifying instances on model performance through both empirical and theoretical research. However, the experiment design for stop words deletion leads to fluctuating test performance, making the results less interpretable. I believe there may not be dominant underlying theoretical properties behind this outcome. On the other hand, the experiment involving the random deletion of all words exhibits a similar pattern in the change of test performance w.r.t the modification level. This could be a critical point when considering long-range correlations as discussed in 2.2.2.

In the subsequent sections, I will outline my efforts in theoretical analysis, including the reasons for not conducting experiments with techniques from information retrieval.



(a) Random deletion of all words



(b) Stop words deletion

Figure 5: Distribution of sentences lengths w.r modifying the CMU-MOSI dataset. Values in the legends refers to the modification level.

4 Failure on theoretical analysis

4.1 Short-length instances

The main reason I can't currently use long-term memory to analyze the results or employ information retrieval techniques is due to the properties of the CMU-MOSI datasets. As shown in Figure 5b, the most common sentence length in CMU-MOSI is around 8 (when the modification level is 0). However, 8-length sentences are too short to provide a valid distribution for information retrieval techniques. For example, Eq. 1 calculates P_X by considering the word frequency of a sentence, while in information retrieval, we often calculate the frequency of the top 5 or 10 documents, which could have 100,000 words. Comparing word distribution in top 5 documents to the whole dataset would be more convincing than comparing the word distribution in a short sentence to the whole dataset. Similarly, Eq. 2 necessitates a lengthy text, whereas most instance length in CMU-MOSI is too short to account for long-range connections.

Therefore, at the end of the experiments, I realize that we need to consider a dataset with longer training examples, such as those with at least 500 words each. Otherwise, we won't be able to utilize the tools in information retrieval or the long-range correlation analysis.

4.2 Goodness of fit

The second fact I have realized is that although natural language obeys Zipf's laws, the word distribution of a natural language dataset may not necessarily conform to Zipf's law. As discussed in Sec. 2.2.1, we want to determine at which modification level the word distribution in CMU-MOSI no longer follows Zipf's law, which may provide insight into analyzing the change in the test performance of models. However, I find that even when the modification level is 0, CMU-MOSI does not follow Zipf's law.

Formally, Zipf's law predicts the normalized frequency $f(k; s, N)$ of the word with rank k among the top N frequent words, as given by the formula [3]:

$$f(k; s, N) = \frac{k^{-s}}{H_{N,s}} \tag{4}$$

where s is the Zipf's law coefficient that characterizes the distribution, and $H_{N,s}$ is the N-th generalized harmonic number. Using Eq. 4 and maximum likelihood, I calculate the optimal Zipf's law coefficient for the CMU-MOSI dataset.

The next step is to use a goodness-of-fit test, such as the Kolmogorov-Smirnov test (Massey Jr, 1951), to compare the CMU-MOSI data to the estimated Zipf's law distribution. However, as shown in Fig. 6, a visual comparison of the distribution of the top 500 frequent words with the estimated Zipf's law distribution indicates that the word distribution does not follow Zipf's law, as this is evident from the low frequency of the top 10 words.

Therefore, since the unmodified CMU-MOSI dataset does not follow Zipf's law, it is not possible to identify a threshold modification level at which the dataset stops following Zipf's law

5 Conclusion and Future work

In this project, we conducted experiments to modify the datasets using different strategies and attempted to identify a theoretical framework to explain the influence of the modification levels. However, the experimental evaluation of the some models like MULT encountered reproducibility issues,

[3]The formula is given by https://en.wikipedia.org/wiki/Zipf's_law.

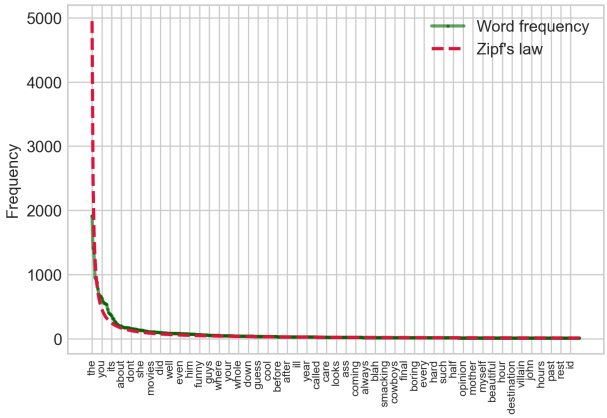


Figure 6: The distribution of the top 500 frequent words in the CMU-MOSI dataset v.s. the estimated maximum likelihood Zipf's law distribution with a coefficient of $s = 0.99$. The x-axis displays every 10th word among the top 500 frequent words.

and the experiment design of stop words deletion needs to be revised to make the results more interpretable.

Theoretical analysis also encountered two problems. Firstly, if we want to analyze the results from the perspective of long-range correlation, choosing a multimodal dataset seems less important than choosing a dataset with longer sentence lengths. Secondly, I had the wrong assumption that a natural language dataset should follow Zipf's law when, in fact, it may not. Thus, we may not be able to analyze the results from this perspective.

Considering future plans, given the current research question, the most important step is to find a suitable dataset with longer sentence lengths. Meanwhile, traditional statistical methods like goodness-of-fit tests may not be suitable for this project, and a benchmark dataset might be more appropriate. For example, once we have a suitable benchmark, we can train our models on modified data and then run the model on the benchmark to see the extent to which the long-range correlation can be captured by models. This plan shall be more intepretable and suitable for a deep learning paper.

References

Eduardo G Altmann, Giampaolo Cristadoro, and Mirko Degli Esposti. 2012. On the origin of long-range correlations in texts. *Proceedings of the National Academy of Sciences*, 109(29):11582–11587.

Olivier Bachem, Mario Lucic, and Andreas Krause. 2017. Practical coreset constructions for machine learning. *arXiv preprint arXiv:1703.06476*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155.

Ondrej Bohdal, Yongxin Yang, and Timothy Hospedales. 2020. Flexible dataset distillation: Learn labels instead of images. *arXiv preprint arXiv:2006.08572*.

Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. 2019. Towards multimodal sarcasm detection (an _obviously_ perfect paper). *arXiv preprint arXiv:1906.01815*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Werner Ebeling and Alexander Neiman. 1995. Long-range correlations between letters and sentences in texts. *Physica A: Statistical Mechanics and its Applications*, 215(3):233–241.

Werner Ebeling and Thorsten Pöschel. 1994. Entropy and long-range correlations in literary english. *Europhysics Letters*, 26(4):241.

Jack Goetz and Ambuj Tewari. 2020. Federated learning via synthetic data. *arXiv preprint arXiv:2008.04489*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. 2022. Dataset condensation via efficient synthetic-data parameterization. In *International Conference on Machine Learning*, pages 11102–11118. PMLR.

Paul Pu Liang, Yiwei Lyu, Xiang Fan, Zetian Wu, Yun Cheng, Jason Wu, Leslie Yufan Chen, Peter Wu, Michelle A Lee, Yuke Zhu, et al. 2021. Multibench: Multiscale benchmarks for multimodal representation learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Frank J Massey Jr. 1951. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78.

Thierry Mora and William Bialek. 2011. Are biological systems poised at criticality? *Journal of Statistical Physics*, 144:268–302.

Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. 2021. Dataset distillation with infinitely wide convolutional networks. *Advances in Neural Information Processing Systems*, 34:5186–5198.

J Arturo Olvera-López, J Ariel Carrasco-Ochoa, J Francisco Martínez-Trinidad, and Josef Kittler. 2010. A review of instance selection methods. *Artificial Intelligence Review*, 34:133–143.

Steven T Piantadosi. 2014. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21:1112–1130.

Vassilis Plachouras, Ben He, and Iadh Ounis. 2004. University of glasgow at TREC 2004: Experiments in web, robust, and terabyte tracks with terrier. In *Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004, Gaithersburg, Maryland, USA, November 16-19, 2004*, volume 500-261 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).

Frank Rosenblatt. 1957. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory.

Noveen Sachdeva, Mehak Preet Dhaliwal, Carole-Jean Wu, and Julian McAuley. 2022. Infinite recommendation networks: A data-centric approach. *arXiv preprint arXiv:2206.02626*.

Kumiko Tanaka-Ishii. Statistical universals of language.

Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2019. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2019, page 6558. NIH Public Access.

Ivor W Tsang, James T Kwok, Pak-Ming Cheung, and Nello Cristianini. 2005. Core vector machines: Fast svm training on very large data sets. *Journal of Machine Learning Research*, 6(4).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. 2018. Dataset distillation. *arXiv preprint arXiv:1811.10959*.

Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. 2016. Mosi: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. *arXiv preprint arXiv:1606.06259*.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. 2020. Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929*.