Automated Text Selection for Raw Data Annotation

Sana Saeed
Department of Computer Science
Comstas University Islamabad
Abbottabad, Pakistan
sanasaeed40994@gmail.com

Ali Haider
Department of Computer Science
Comstas University Islamabad
Abbottabad, Pakistan
alihaider.ah1510@gmail.com

Kashif Bilal
Department of Computer Science
Comstas University Islamabad
Abbottabad, Pakistan
kashifbilal@cuiatd.edu.pk

Abstract*— The escalating volume of user-generated web content urges the need for advancements in automated text processing. The performance of state-of-the-art machine learning and deep learning techniques is significantly influenced by the quality of training data. However, real-world user-generated web content often contains excessive repetition. Such redundancy and data bloat have the potential to impede the performance of automated text analysis algorithms. Training datasets must mirror real-world complexity for optimal generalization, maintaining a balance between diversity and relevance, with minimal redundancy. Moreover, advanced techniques for natural language analysis are data-hungry; hence, substantial efforts are required to prepare a suitably large dataset. Large datasets are prepared by a group of team members, with each team member working on their own chunk of data, as it is not feasible for a single person to go through a huge amount of data alone. Therefore, automated text pruning can facilitate the selection of diverse text and enrich the meaningful vocabulary by reducing redundancy and eliminating unnecessary and irrelevant information. Hence, in this study, we propose text selection techniques based on text similarity for automated unique review selection from a large dataset. However, with large datasets, direct pairwise similarity comparison drastically increases the time complexity for similarity matching. Therefore, we adopted a cluster-based vectorized text pruning technique for similarity-based redundancy removal.

Keywords*— Automated text selection, raw text redundancy removal, text pruning, text similarity based text pruning.

I. INTRODUCTION

The exponential surge in user-generated textual content on the web is urges need for research advancements geared towards automated text processing [1], [2]. State-of-the-art text processing techniques utilize machine learning and deep learning algorithms for tasks such as data analysis, information extraction, and decision-making [3]–[5]. The performance of machine learning algorithms heavily relies on the quality of input data and data labels [6]. Moreover, cutting edge machine learning and deep learning algorithms exhibit data-hungry characteristics [6], [7]. However, realizing the benefits of modern deep learning techniques, substantial efforts are made in dataset preparation. The training datasets must mirror real-world complexities for optimal generalization, maintaining a balance between diversity and relevance, with minimal redundancy. However, real-world user-generated web content often contains excessive repetition that can bloat content with unnecessary data, thus hampering automated text analysis process. Contemporary deep learning algorithms learns the underlying patterns from text, however, redundancy compromises their efficiency and accuracy by adds noise, and diverting from valuable information and hindering accurate interpretation. Therefore addressing undesired data redundancy issue is crucial for optimal automated text analysis, delivering precise insights from user-generated web content.

During manual dataset preparation and data annotation process meticulous human efforts are need, that makes the whole process laborious and time-intensive. Therefore, in order to make it efficient and effective task is usually performed by a team of annotators and dataset is divided into the chunks of data and each annotator is assigned a chunk of data for annotations. However, in such scenario there is high probability that some chunks of information is annotated and fed to the model multiple times whereas, some important aspects are missed. Moreover, manually scrutinizing redundant information laborious and time-intensive task, particularly when addressing a multitude of concerns. However, selection of representative data items is crucial for machine learning algorithms to effectively capture the intricacies of the entire problem being addressed [8].

A good dataset must encompass a spectrum of variations, patterns, and complexities to facilitate the model's generalization to new, unseen instances [6]. For instance, when constructing a sentiment analysis model for restaurant reviews, the dataset should encompass positive, negative, and neutral sentiments from different types of restaurants and customer experiences. However, data for real word user generated content is usually scrapped randomly from the web resources and in order to assure that data consists of all scenarios and good representative of real world data complexities large amount of data is gathered for annotation. While manual inclusion of all necessary features is a laborious task that demands considerable human effort and time, automated text scrutiny can significantly reduce this burden [6]. Automated text selection processes can assist in choosing diverse data while mitigating information redundancy and promoting vocabulary diversification.

In the realm of real-world raw text content, such as user reviews, it is observed experimentally that the issue of text contains very high ratio of redundancy [9]. Data redundancy in machine learning and deep learning poses significant challenges that can undermine model performance and outcomes [10]. Redundant data, featuring similar instances, can lead to overfitting, causing models to memorize specifics instances instead of learning general patterns that result in poor performance on unseen data [10], [11]. Moreover, with inadequate redundancy in dataset, the model memorizes training data instead of learning underlying patterns that are triggered by redundant and highly similar data points within the training dataset, and results in biased training [11]. Hence, repetitive data may not contribute significantly to information gain and can also escalate the computational demands of the training process [11]. It is equally essential to establish a diverse dataset encompassing various scenarios, as diversity enables the model to adeptly handle an array of situations, enhancing its capacity for effective generalization. Simultaneously, the data should be informative, capturing crucial facts of the problem and underscoring features or

patterns pivotal to the model's decision-making process. Furthermore redundancy might offer stability, however, it requires meticulous handling to counteract associated drawbacks, to ensure the overall efficacy of machine learning and deep learning algorithms.

The data annotation processes employed in existing studies predominantly rely on human efforts. Moreover, large dataset[3], [12]–[16] are annotated by a team of annotators, therefore it is nearly impossible to remove all the redundant and duplicate information from the dataset, as all the team members are annotating the data separately and they do not have any information about the chunk of dataset that other team members are annotating. Moreover, direct pairwise document similarity matching is not computationally feasible for the large datasets. Hence, in response to these challenges, we present a novel solution through an automated text selection process, offering the following key contributions:

1. Introducing an automated mechanism for the elimination of redundant information and text selection from extensive datasets, utilizing text similarity as a guiding principle.
2. Implementing a highly efficient vectorized methodology to streamline the process of text pruning, thereby facilitating pairwise comparison of document similarity.
3. Conducting a comprehensive evaluation of our refined text dataset, assessing its content uniqueness and vocabulary diversity using a diverse range of evaluation metrics. This multifaceted evaluation approach ensures the effectiveness of our proposed solution.

The subsequent sections of this paper are structured as follows: In Section 2, we delve into the landscape of related research. Section 3 outlines the proposed methodology for text selection. The presentation of our experimental findings and evaluations can be found in Section 4. Lastly, in Section 5, we conclude by summarizing our key insights and discussing avenues for future work.

II. LITERATURE REVIEW

The predominant method for creating (Aspect Based Sentiment Analysis) ABSA datasets involves a structured annotation process [4], [5], [17]–[24]. Initially, one annotator (A) marks a portion of the data, and then another annotator (B) validates it while adhering to predefined annotation policies and rules. Subsequently, annotator A proceeds to label the remaining dataset, occasionally incorporating further instructions based on prior disagreements. In cases of uncertainty, collaboration between annotators A and B was sought. Discrepancies were resolved through the involvement of a third expert annotator or by employing majority voting. Interestingly, most datasets lack inter-annotator agreement scores due to this approach.. Different datasets, such as SemEval [3], [12]–[16], and MAMS [25], exhibit variations in aspect task annotations. Many datasets are expansions of existing ones, aimed at addressing missing aspects or introducing new subtasks. Challenges in annotation encompass delineating multiword aspect term boundaries, handling complex sentence structures, and clarifying neutral polarity ambiguities. English restaurant datasets like SemEval [3], [12]–[16], MAMS [25], TOWE, and ASTE originate from citysearch.com, while laptop data is sourced from Amazon.com reviews. Nonetheless, despite these efforts, none of the mentioned datasets have provided an automated mechanism for text selection and redundancy removal from the dataset consisting of real-world user generated-content.

III. METHODOLOGY

In our proposed methodology, initial we apply data preprocessing to eliminate redundant and unnecessary information from the dataset. Additionally, to filter out documents exhibiting high similarity with one another we measured document similarity using cosine similarity, measured on the TF-IDF embedding. Cosine similarity performs pairwise comparisons for every document that lead to a factorial increase (n!) of pair wise comparisons.

To mitigate this challenge, we adopt a divide and conquer strategy. Initially, we apply k- means on BERT document level embedding to group similar documents together and calculated the cosine similarity matrix for each cluster. Further we adopted a vectorized approach to prune text efficiently, that removes documents exceeding specified threshold values of similarity with some other document. The entire process of text selection and redundant information pruning is elucidated in Figure 1 for a clear visual representation.

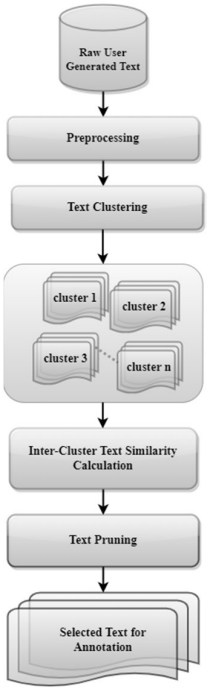

Fig. 1 Process flow diagram of similarity based automated text redundancy removal.

A. Preprocessing

Text preprocessing within the realm of natural language processing (NLP) serves as a crucial step aimed at augmenting data quality and eliminating extraneous details from textual content. Furthermore, it plays a pivotal role in diminishing data dimensionality, thereby enhancing algorithmic efficiency and elevating the caliber of extracted features. This preparatory phase encompasses an array of techniques and operations that are applied to raw text data, orchestrating its cleansing, normalization, and transformation

into a format conducive to subsequent analysis, modeling, and feature extraction.

The primary objective behind text preprocessing is to elevate data quality, minimize noise, and amplify the efficiency and effectiveness of ensuing NLP tasks. After meticulous data analysis, we systematically executed a series of text preprocessing actions. These actions were meticulously selected based on the specific nature of the task, the intrinsic attributes of the text, and the desired outcomes.

1) Lowercasing

Converting all text to lowercase ensures uniformity and consistency within the textual content. Lowercase transformation entails converting all alphabetic characters to their lowercase counterparts, regardless of their initial case. Lowercasing ensures that all words, regardless of their original formatting, are treated in a standardized manner. This uniformity aids in simplifying subsequent analyses, processing and helps to mitigate discrepancies that may arise due to variations in letter case. By unifying the text by lowercasing, the potential for inconsistencies is minimized, contributing to a more coherent and streamlined text processing pipeline.

1) Removing HTML Content, Hyperlinks, and Mentions

The procedure entails the removal of HTML (Hyper Text Markup Language) content, hyperlinks, and mentions from the text. This step is undertaken to eliminate extraneous web-related elements and user references that might not contribute to the core textual content. This preprocessing ensures that the subsequent analysis focuses solely on the intrinsic textual information, enhancing the quality and relevance of the processed text.

2) Removing Punctuation

Removing punctuation, including periods, commas, and question marks, is employed to eliminate non-essential characters that often lack substantial meaning when considered independently. This preprocessing step aids in reducing noise and streamlining the text for more effective analysis and feature extraction.

3) Handling Contractions

Handling contractions involves expanding them to their full forms, for instance, "don't" to "do not". This process ensures uniform token representation, aiding in accurate analysis and subsequent natural language processing tasks. By treating contractions and their expanded forms consistently, potential variations in meaning are minimized, leading to more reliable results.

4) Removing Stop Words

Stop words refer to commonly used words in a language that lack substantial individual meaning within the context of natural language processing and analysis. The process of eliminating stop words involves excluding non-informative terms such as "and", "the", and "is". Removing stop words form the text enhances the significance of content-bearing terms, refining the text for more meaningful natural language processing tasks. Each language possesses its own collection of stop words; for English, a standardized stop word list comprising 170 words is available in the NLTK library. In our approach, we utilized NLTK to perform stop word removal from the text.

5) Lemmatization

Lemmatization involves converting words to their base or root forms, considering the linguistic context of word. This process aids in standardizing vocabulary, enhancing the accuracy of word analysis, and ensuring a deeper understanding of word meaning within the given context.

6) Removing Duplicate Entries

Eliminating duplicate entries is a fundamental procedure in data preprocessing that yields significant benefits for the analysis phase. Duplicates have the potential to distort analytical outcomes, as they artificially amplify the occurrence of specific data points, leading to skewed insights. However, through the identification and subsequent removal of duplicates, the reliability of dataset is augmented, resulting in a clearer representation of the underlying information. Moreover, this action contributes to the optimization of computational processes, leading to resource conservation and an overall enhancement in the efficiency of data analysis.

B. Text Embedding

Text embedding are numerical representations of textual data designed to capture semantic and contextual relationships between words, phrases, sentences, or even entire documents. They convert the complex and sparse nature of natural language into dense and continuous vectors in a high-dimensional space. These embedding play a fundamental role in many natural language processing (NLP) tasks, as they enable machine learning algorithms to process and understand textual information more effectively.

1) TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) [26] is numerical representation used in information retrieval and text mining to evaluate the importance of a term (word) within a document relative to a collection of documents (corpus). TF-IDF takes into account both the frequency of a term in a specific document (local importance) and the rarity of the term across the entire collection of documents (global importance).

Term Frequency (TF) measures the frequency ($f_{t,d}$) of a term (t) within a specific document (d). It's calculated by dividing the number of times a term appears in the document appears ($f_{t,d}$) by the total number of terms in the document as in Eq. (1). It is a local measure of importance for a term.

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t'}} \qquad (1)$$

Inverse Document Frequency (IDF) measures how rare a term is across the entire corpus. It's calculated by taking the logarithm of the total number of documents, N divided by the number of documents containing the term, $\{d \in D: t \in d\}$, as in Eq. (2). It's a global measure of importance.

$$idf(t,D) = log\frac{N}{|\{d \in D: t \in d\}|} \qquad (2)$$

TF-IDF score of a term in a specific document is obtained by multiplying its TF value by its IDF value as in Eq. (3). This score represents the importance of the term in the context of that document.

$$TFIDF = tf(t,d) \times df(t,D) \qquad (3)$$

The TF-IDF score for each term in each document creates a numerical representation of the documents. In this representation, terms that are frequent in a particular document but rare in the entire corpus will receive high TF-

IDF scores, indicating their importance to that document's content.

2) BERT Document Embedding

BERT [27] (Bidirectional Encoder Representations from Transformers) is a state- of-the-art pre-trained transformer based language model that has revolutionized many natural language processing (NLP) tasks by providing contextually rich word level and document level embedding . Unlike traditional models that either do not account for the context e.g TF-IDF or contextual models like word2vec, that process text from left to right or right to left, BERT introduces bidirectional learning, allowing it to consider the entire context of a word by processing both directions in a sentence. This contextual representation enables BERT to capture intricate relationships between words and produce highly contextually and semantically rich embedding.

C. K-Means Clustering

K-means clustering is a widely used unsupervised machine learning technique that aims to partition a dataset into distinct groups or clusters based on similarity patterns among data points. For text clustering text is converted into numeric data points, we used BERT-sentence level embedding to convert text into numeric representative. The algorithm iteratively assigns each data point to the nearest cluster center and recalculates cluster centers as the mean of the points assigned to that cluster. This process continues either, until convergence or until it meets a specific criteria. The clustering process results in clusters that minimize the sum of squared distances between data points and their assigned cluster centers. K-means is a simple yet effective method for identifying inherent structures within data, enabling insights into grouping patterns and allowing for the categorization of data points with shared characteristics.

D. Cosine Similarity

Cosine similarity [28] metric quantifies the similarity between two non-zero vectors by examining the cosine of the angle between them in a multi-dimensional space. Rather than focusing solely on the magnitudes of the vectors, cosine similarity concentrates on their orientations, making it especially helpful when dealing with high-dimensional data. Cosine similarity stands as a pivotal metric within Natural Language Processing, allowing the assessment of text similarity between two documents regardless of their sizes. This method involves representing words as vectors and text documents as n-dimensional vectors within a vector space we uses TF-IDF as described in previous section. By projecting these vectors in the multi-dimensional space, the cosine similarity metric mathematically gauges the angle between them. Mathematically, the calculation of Cosine similarity compares the dot product of two non-zero vectors, A and B, and divides it by the product of their Euclidean norms (magnitudes), $||A||$ an $||B||$, respectively, as in Eq. (4). When applied to the realm of text similarity, A and B represent the vectorized forms of two text documents.

$$Cosine\ Similarity = \frac{\|A\| \cdot \|B\|}{A \cdot B} \qquad (4)$$

Numerical values of cosine similarity for text similarity ranges between 0 and 1, wherein a score of 1 signifies two vectors with identical orientations. Conversely, a score closer to 0 indicates diminished similarity between the two documents.

E. Vectorized Pruning Redundent Information

We have utilized vectorized pruning as a strategy to manage the intricacies inherent in pairwise document similarity comparisons. This approach involves the computation of a similarity matrix through cosine 1 similarity calculations for a given document list, DOC. With "m" representing the total document count, this operation yields a similarity matrix (S) of dimensions m×m. Subsequently, we binary-coded the similarity matrix by applying a threshold similarity of 0.8. As a result, we obtained a new matrix B in which values exceeding 0.8 were replaced with, while others were set to 0. This binary differentiation allowed us to distinguish between similar and dissimilar matrices. However, there is repeating comparison of two same documents in the matrixes since we are finding similarity of list of document with itself. To mitigate this, we opted to retain only the upper half of the matrix, situated above the diagonal, while the diagonal and lower diagonal elements were reset to 0. This action yielded a refined matrix D comprised solely of single comparisons. Subsequently, we extracted a list of indices L_i from matrix D where the matrix elements possessed a value of 1, signifying similar documents. This curated list, L_i provides the positions of similar documents that necessitate removal from the original document list DOC.

IV. Results and Evaluations

A. Dataset and Experimental Setup

For evaluating our technique we scrapped 659,507 restaurant reviews form the internet resources. However, upon subjecting the raw text to preprocessing and eliminating duplicate entries, our dataset was refined to encompass 88,469 distinct reviews. Notably, our examination revealed a substantial presence of redundancy, with 571,037 sentences, constituting approximately 86.6% of the dataset, being duplicates.

In order to enhance the manageability of the data, we employed a clustering approach to group reviews. Our strategy ensured that each cluster contained, on average, fewer than 1,000 reviews. Through this clustering process, we effectively organized the data into 89 distinct clusters.

To further refine the dataset and eliminate redundant content, we employed a cosine similarity-based review pruning mechanism with a threshold set at 0.8. This refinement process led to the removal of an additional 1,163 reviews that exhibited redundant information. As a result, our final dataset is composed of 88,469 reviews that stand as unique representations within their contextual contexts.

B. Evaluation Metrices

The goal of our proposed review selection approach is to eliminate redundant information while simultaneously enriching the dataset's diversity. This is achieved by guaranteeing the distinctiveness of information within each individual training example. The act of discarding redundancy is pivotal, as it mitigates various challenges encompassing over fitting, bias, heightened model intricacy, reduced efficiency, extended training duration, and potential misrepresentations in assessments.

To thoroughly gauge the diversity and distinctiveness of the text, we harnessed a range of evaluation metrics. These encompass the Type-Token Ratio (TTR), Hapax Legomena

count, and the Herfindahl-Hirschman Index (HHI). Through the utilization of these metrics, we aim to comprehensively evaluate the extent to which our process succeeds in enhancing dataset variety and information uniqueness.

1) Type-Token Ratio (TTR)

Type-Token Ratio (TTR) is a linguistic measure used to analyze the lexical diversity or richness of a text. It assess the diversity of vocabulary in a given text. TTR is calculated by dividing the number of unique words (types) in a text by the total number of words (tokens) in the text and then multiplying by 100 to get a percentage as in Eq. (5). A higher TTR indicates a more diverse vocabulary, while a lower TTR suggests more repetitive use of words.

$$TTR = \frac{Total\ Number\ of\ \ Unique\ Words}{Total\ Number\ of\ Words} \times 100 \qquad (5)$$

2) Hapax Legomena:

Hapax Legomena" is a linguistic term referring to words that appear only once in a specific context, corpus, or dataset, representing rare instances without repetition. This concept sheds light on the uniqueness and scarcity of certain words within a given text. Identifying hapax legomena in text analysis provides insights into vocabulary richness and diversity, allowing assessment of language uniqueness. Higher counts of hapax legomena suggest specialized content, while lower counts indicate more common language. Its applications span literary analysis, historical linguistics, and natural language processing, enhancing understanding of word distribution, linguistic trends, and language structure. Recognizing these rare words provides deeper insights into a text's distinct characteristics and vocabulary.

3) Herfindahl-Hirschman Index (HHI):

Originally used in economics, HHI can be adapted to measure word diversity in the text. It computes the sum of the squares of the proportions of different words in the text as n Eq. (6), where *n* represents the total number of distinct words in the text. f_i is the frequency of the i^{th} distinct word in the text and *N* stands for the total number of words in the text.

$$HHI = \sum_{i=1}^{n} \left({}^{f_i}/_{N} \right)^2 \qquad (6)$$

A higher HHI score indicates more concentrated word usage, while a lower score suggests greater word diversity.

4) Meaningful Words Ratio:

This measure focuses on the proportion of content-bearing words (nouns, verbs, adjectives, etc.) in a text, excluding function words like "the," "and," "is," etc. Mathematically, Meaningful Words Ratio (MWR) involves dividing the count of content-bearing words (CCBW) by the total number of words in the text after excluding function words, for instance, stop words ($CWEFW$), as in Eq. (7).

$$MWR = \frac{CCBW}{CWEFW} \qquad (7)$$

This formula quantifies the ratio of meaningful words to the overall text length, providing insights into the proportion of words that contribute to the content's semantic richness

5) Entropy:

Entropy is a measure of uncertainty or randomness in a text's vocabulary distribution. A higher entropy suggests higher diversity. A dataset with *n* number of total of distinct words in the text. and $P(xi)$ is the probability of the i^{th} distinct word occurring in the text, entropy can be calculated as in Eq. 8

$$Entropy = -\sum_{i=1}^{n} (P(x_i) \times \log_2(P(x_i))) \qquad (8)$$

A higher value of entropy in the context of text analysis indicates a greater diversity and randomness in the distribution of words within the text. In other words, when the entropy value is higher, it suggests that the words used in the text are more evenly distributed and not dominated by a small set of frequently occurring words.

C. Experimental Results

Initially, our dataset consists of 659,507 restaurant reviews. However, after filtering the text using our proposed text selection method, we are left with 88,469 unique entries. To measure the uniqueness and diversity of the text, we use several evaluation metrics, the results of which are elaborated in Table 1.

Initially, the count of single-occurring words, known as Hapax Legomena, for the raw data is 912. After the redundancy pruning process, this count increases significantly to 15,873. This increase in the Hapax Legomena score indicates the successful removal of redundancy. Additionally, the Type-Token Ratio, which measures lexical diversity, significantly increases from 0.35% to 1.7%.

Furthermore, the value of Entropy, which reflects the diversity and randomness in word distribution based on the probability of word occurrence, increases from 16.22 to 16.41. The Meaningful Word Ratio, indicative of the proportion of words contributing to semantic content, increases from its initial value 0.6 to 1, signifying a positive trend.

The Herfindahl-Hirschman Index (HHI), assessing word diversity based on word frequency, decreases from 0.0086 to 0.0026. This decline in HHI suggests a positive trend, as a smaller HHI value signifies greater word diversity.

In summary, all evaluation metrics consistently demonstrate a trend towards increased text diversity and uniqueness in the text selected after undergoing the text pruning processing.

Table I. Post text-pruning evaluation results.

Evaluation Metrics	Raw Dataset	Selected Dataset
Number of Reviews	659,507	88,469
Type-Token Ratio (TTR)	0.35%	1.7%
Hapax Legomena	921	15873
Herfindahl-Hirschman Index	0.0086	0.0026
Meaningful Words Ratio	0.6	1
Entropy	16.22	16.41

V. Conclusion and Future Work

This paper addressed the intricate challenges posed by redundancy and similarity within real world user generated data, proposing an innovative automated text selection approach. By removing duplicate entries and leveraging vectorized methods for efficient pruning, the study successfully enhanced the quality and diversity of the dataset while curbing computational complexity. The results demonstrated that the proposed technique significantly refined the dataset, reducing redundancy and promoting richer content. The evaluation metrics, including Type-Token Ratio, Hapax Legomena count, and Herfindahl-Hirschman Index, indicated substantial improvements in vocabulary diversity and information uniqueness.

In future work, we plan to assess the performance of our task using advanced deep learning and transformer-based

similarity metrics. Additionally, we intend to evaluate the performance across domains beyond restaurant reviews.

REFERENCES

[1] J. Wang and Y.-L. Liu, "Deep learning-based social media mining for user experience analysis: A case study of smart home products," *Technol. Soc.*, vol. 73, p. 102220, 2023, doi: https://doi.org/10.1016/j.techsoc.2023.102220.

[2] Z. Wang, W.-J. Huang, and B. Liu-Lastres, "Impact of user-generated travel posts on travel decisions: A comparative study on Weibo and Xiaohongshu," *Ann. Tour. Res. Empir. Insights*, vol. 3, no. 2, p. 100064, 2022, doi: https://doi.org/10.1016/j.annale.2022.100064.

[3] N. Hossain, J. Krumm, M. Gamon, H. Kautz, and M. Corporation, "SemEval-2020 Task 7: Assessing Humor in Edited News Headlines," no. 2019, 2020.

[4] P. Nandwani and R. Verma, "A review on sentiment analysis and emotion detection from text," *Soc. Netw. Anal. Min.*, vol. 11, no. 1, pp. 1–19, 2021, doi: 10.1007/s13278-021-00776-6.

[5] S. Sazzed, "A Hybrid Approach of Opinion Mining and Comparative Linguistic Analysis of Restaurant Reviews," *Int. Conf. Recent Adv. Nat. Lang. Process. RANLP*, pp. 1281–1288, 2021, doi: 10.26615/978-954-452-072-4_144.

[6] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: a big data-ai integration perspective," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 4, pp. 1328–1347, 2019.

[7] H. Li, "Deep learning for natural language processing: advantages and challenges," *Natl. Sci. Rev.*, vol. 5, no. 1, pp. 24–26, 2017, doi: 10.1093/nsr/nwx110.

[8] A. Ratner, C. De Sa, S. Wu, D. Selsam, and C. Ré, "Data programming: Creating large training sets, quickly," *Adv. Neural Inf. Process. Syst.*, pp. 3574–3582, 2016.

[9] A. Timoshenko and J. R. Hauser, "Identifying customer needs from user-generated content," *Mark. Sci.*, vol. 38, no. 1, pp. 1–20, 2019.

[10] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Glob. Transitions Proc.*, vol. 3, no. 1, pp. 91–99, 2022, doi: https://doi.org/10.1016/j.gltp.2022.04.020.

[11] M. Al-Rawi, Y. Al-Zuqary, F. B. Saghezchi, J. Yang, J. Bastos, and J. Rodriguez, "Data redundancy may lead to unreliable intrusion detection systems," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2017, pp. 1897–1902.

[12] R. Van Der Goot, "Effectiveness of Intermediate Training on an Uncurated Collection of," pp. 230–245, 2023.

[13] J. Barnes *et al.*, "SemEval 2022 Task 10 : Structured Sentiment Analysis," pp. 1280–1295, 2022.

[14] S. M. Mohammad and F. Bravo-marquez, "SemEval-2018 Task 1 : Affect in Tweets," pp. 1–17, 2018.

[15] S. Kiritchenko and S. M. Mohammad, "SemEval-2016 Task 7 : Determining Sentiment Intensity of English and Arabic Phrases," pp. 42–51, 2016.

[16] E. Agirre, D. Cer, M. Diab, and A. Gonzalez-agirre, "SemEval-2012 Task 6 : A Pilot on Semantic Textual Similarity," no. 3, pp. 385–393, 2012.

[17] M. Pontiki *et al.*, "SemEval-2016 Task 5 : Aspect Based Sentiment Analysis," pp. 19–30, 2016.

[18] J. Zhou, J. Zhao, J. X. Huang, Q. V. Hu, and L. He, "MASAD: A large-scale dataset for multimodal aspect-based sentiment analysis," *Neurocomputing*, vol. 455, pp. 47–58, 2021, doi: https://doi.org/10.1016/j.neucom.2021.05.040.

[19] N. Akoury, S. Wang, J. Whiting, S. Hood, N. Peng, and M. Iyyer, "STORIUM: A dataset and evaluation platform for machine-in-the-loop story generation," *EMNLP 2020 - 2020 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, pp. 6470–6484, 2020, doi: 10.18653/v1/2020.emnlp-main.525.

[20] S. U. S. Chebolu, F. Dernoncourt, N. Lipka, and T. Solorio, "Survey of Aspect-based Sentiment Analysis Datasets," 2022, [Online]. Available: http://arxiv.org/abs/2204.05232

[21] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "SemEval-2015 Task 12: Aspect Based Sentiment Analysis," *SemEval 2015 - 9th Int. Work. Semant. Eval. co-located with 2015 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. NAACL-HLT 2015 - Proc.*, pp. 486–495, 2015, doi: 10.18653/v1/s15-2082.

[22] N. Asghar, "Yelp Dataset Challenge: Review Rating Prediction," 2016, [Online]. Available: http://arxiv.org/abs/1605.05362

[23] A. Krishna, V. Akhilesh, A. Aich, and C. Hegde, "Sentiment Analysis of Restaurant Reviews Using Machine Learning Techniques," in *Emerging Research in Electronics, Computer Science and Technology*, V. Sridhar, M. C. Padma, and K. A. R. Rao, Eds., Singapore: Springer Singapore, 2019, pp. 687–696.

[24] M. Liakata, G. Bouchard, and S. Riedel, "SentiHood: Targeted Aspect Based Sentiment Analysis Dataset for Urban Neighbourhoods," 2015.

[25] Q. Jiang, L. Chen, R. Xu, X. Ao, and M. Yang, "A Challenge Dataset and Effective Models for Aspect-Based Sentiment Analysis," pp. 6280–6285, 2019.

[26] C. Liu, Y. Sheng, Z. Wei, and Y.-Q. Yang, "Research of text classification based on improved TF-IDF algorithm," in *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)*, 2018, pp. 218–222.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv Prepr. arXiv1810.04805*, 2018.

[28] D. Gunawan, C. A. Sembiring, and M. A. Budiman, "The implementation of cosine similarity to calculate text relevance between two documents," in *Journal of physics: conference series*, 2018, p. 12120.