Auto Complete System

Simran Choudhary (2021205)
Saksham Lal(2020402)
Tanmay Parashar(MT23100)
Sarthak Pol(MT23082)

Abstract

In the field of Common Sense Inference, enhancing a machine's proficiency in generating coherent sentence completions has gained significant importance. This task poses a challenge for machines as it necessitates a deeper understanding beyond individual words, extending to grasp the underlying meaning and context of sentences. Our focus centers on the analysis of the HellaSwag dataset, a significant benchmark to discern patterns in various linguistic parameters. By statistically exploring this dataset, we aim to unravel the complexities of language context and propose effective approaches to improve a machine's ability to generate sensible and contextually appropriate sentence completions.

1 Introduction

In the digital age, the way we interact with machines has evolved rapidly, leading to an emphasis on user experience and efficiency. One of the manifestations of this evolution is the sentence auto-completion system. At its core, a sentence auto-completion system aims to predict and suggest the following word(s) a user is likely to type based on the preceding context. Such systems are ubiquitous, finding applications in search engines, messaging apps, email clients, and many other platforms where text input is essential.

The essence of an auto-completion system lies in its ability to "understand" and "predict" the user's intent. Historically, many of these systems were rule-based or utilized simple frequency counts. However, with the advent of machine learning, the paradigm shifted towards data-driven models that can learn from vast amounts of text data to make predictions.

This project dives into the development of a sentence auto-completion system using classical machine-learning techniques. We will explore the entire pipeline, from data preprocessing to model training and evaluation, emphasizing the importance of accuracy as our primary performance metric.

2 Dataset Summary

2.1 Dataset Characteristics

The training data consists of single-choice endings, which might simplify the training task. The validation data offers a more complex task with multiple choices for endings. The test data has multiple choices and will be crucial for evaluating the model's actual performance since it lacks labels.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

2.2 Nature of the Task

The primary objective is to select or validate an appropriate sentence ending based on the provided context sentences. The training dataset is more about validating if a given ending is appropriate, suggesting a binary classification approach. The validation and test datasets involve choosing the correct ending from multiple options.

2.3 train.csv

Contains 39,905 samples. Used for training the model. All samples are marked with the split type 'in-domain'. Each sample has only one choice for the ending.

2.4 Validation.csv

Contains samples with multiple choices for sentence completions. Each sample has a label indicating the correct ending. Used for validating the model's performance. Contains two types of splits: 'indomain' and 'zeroshot'.

2.5 test.csv

Contains samples with multiple choices for sentence completions. It does not have labels for the endings, as it is meant for testing the model. Contains the split type 'indomain'.

3 Previous Work and Literature survey

3.1 Introduction

HellaSwag dataset suggests a new path forward for NLP research, in which benchmarks co-evolve with the evolving state-of-the-art in an adversarial way, so as to present ever-harder challenges. SWAG is a dataset for commonsense NLI.[1] For each question, a model is given a context from a video caption and four ending choices for what might happen next. Only one choice is right – the actual next caption of the video.

3.2 Zellers et al. (2018)

This fresh challenge, which revolves around common sense-based natural language inference, appeared to be straightforward for humans, with an impressive 88% accuracy rate for humans. Zellers et al. (2018) introduced Adversarial Filtering .The key idea is to produce a dataset D which is adversarial for any arbitrary split of (Dtrain, Dtest).This requires a generator of negative candidates which we achieve by using a language model. The Incorrect answers were massively oversampled from a language model trained on in-domain data, and then selected using an ensemble of adversaries. The selection process happens iteratively, Last, humans validate the data to remove adversarial endings that seem realistic.

3.3 BERT (Devlin et al., 2018)

Now with introduction of this ,it near human-level performance was reached. However, BERT (Devlin et al., 2018) soon reached over 86%, almost human-level performance. Deep models such as BERT do not demonstrate robust commonsense reasonining ability by themselves. Instead, they operate more like rapid surface learners for a particular dataset. Their strong performance on SWAG is dependent on the finetuning process, wherein they largely learn to pick up on dataset-specific distributional biases.

We study this question by introducing Hella Swag, a new benchmark for commonsense NLI. We use Adversarial Filtering (AF), a data collection paradigm in which a series of discriminators is used to select a challenging set of generated wrong answers. AF is surprisingly effective towards this goal: the resulting dataset of 70k problems is easy for humans (95.6% accuracy), yet challenging for machines (50%).

Even though BERT was never exposed to randomly shuffled text during pretraining, it easily adapts to this setting, which suggests that BERT is largely performing lexical reasoning over each (context, answer) pair.

In all cases, BERT performance begins high (70-90%), but there are enough generations for Adversarial Filtering to lower the final accuracy considerably. While the one-sentence case converges to slightly higher than random – 35% when it converges – the two and three sentence cases are higher, at 40% and 50% respectively. Given more context, it becomes easier to classify an ending as machine- or human written.

4 Exploratory Data Analysis

To understand the text length distribution within our dataset, we applied a function, count_characters, to various textual columns and the following findings were recorded:

4.1 Character Count Analysis in Text Columns

The columns ctx_a, ctx_b, ctx, and endings were evaluated for their character count.

On average, ctx_a has a character count of 214.182, while ctx_b averages 3.313. Similarly, columns ctx and endings have average counts of 217.865 and 560.834, respectively. Visual representations, such as histograms, showcased varied distributions of text lengths across these columns.

Implications: The varied lengths indicate potential normalization or truncation steps for model training. Columns with shorter lengths may require additional scrutiny to ensure they carry significant informational content.

Based on this analysis, we'll explore text normalization techniques and investigate columns with notably shorter or longer text lengths.

4.2 Word Count Analysis in Text Columns

To further evaluate the textual content within our dataset, we conducted a word count analysis on the columns ctx_a, ctx_b, ctx, and endings.

The ctx_a column has an average word count of 38.21, whereas ctx_b averages 0.77. The columns ctx and endings contain an average of 38.99 and 97.98 words, respectively.

Implications: Understanding word counts can help in refining tokenization strategies during model training. Columns with significantly lower word counts may need to be examined to ensure they provide enough context or information.

Figure 1 below represents a word cloud of the most occurring words in these columns.(Larger words are more frequent) The visualization thus serves not only as an effective summary of the dataset's content but also as a tool for identifying key areas for further analysis and exploration in our study.

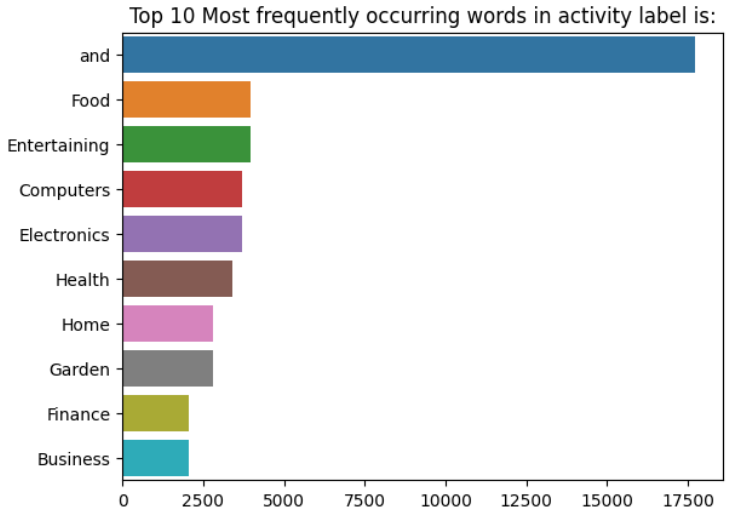
Figure 1: WORD CLOUD.

4.3 Term Frequency Analysis

To better understand the text data in specific columns, we conducted basic pre-processing. This involved creating new columns to store lists of words for subsequent analysis or processing. The next step was to identify the 10 most common words in each corpus. A dedicated function was utilized to generate and display bar plots, highlighting the frequency of these top words.

The outcome of this analysis is the generation of four distinct bar plots, each showcasing the 10 most frequently occurring words in a specific corpus. These visualizations provide valuable insights into the predominant words within each context, aiding in the interpretation of activity labels.

Refer to Figure 1 for a visual representation that offers a rough idea of the distribution of activity labels. These visual aids contribute to a clearer understanding of the language patterns present in the respective corpora, facilitating further insights into the nature of the dataset.



4.4 N-Grams Analysis

In our initial analysis to uncover meaningful patterns within the language context of the HellaSwag dataset, we employed n-gram analysis as a foundational method. Leveraging the CountVectorizer module with a specified n-gram range, we examined both bigrams (two-word phrases) and trigrams (three-word phrases) present in the 'activity_label' column of our dataset.Key inferences from our analysis include:

4.4.1 Bigrams

Domain-specific Vocabulary For domain-specific datasets, such as HellaSwag, frequent bigrams highlight key terms and vocabulary specific to the domain.

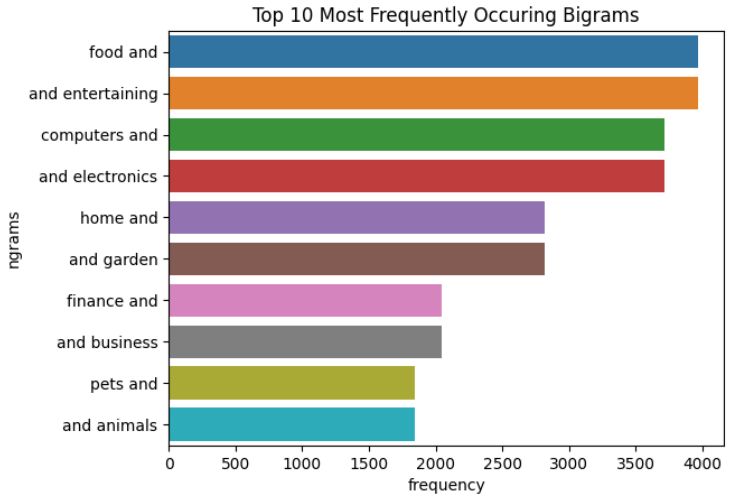
Sentence Structure Analyzing the order and frequency of bigrams provides insights into common sentence structures, a crucial aspect for developing a model adept at coherent sentence completion.

Sequential Patterns Trigrams offer a more detailed exploration of sequences of three words, providing insights into specific language constructs or narrative structures beyond what bigrams reveal.

4.4.2 Inferences

1. Data Quality Assurance: Notably, the top 10 most frequent bigrams and trigrams consist of meaningful English phrases, ensuring that the dataset is free from major noise arising from incoherent or incorrect sentences.

2. Impact on Context: A substantial presence of common stop words, such as 'and,' in bigrams/trigrams prompts further investigation into its impact on context and meaning within sentences.



4.4.3 Deductions from "And"-dominated N-Grams:

Bigrams:

1.Ubiquitous Connector: The consistent inclusion of "and" in top bigrams indicates a prevalent use of this conjunction, suggesting frequent connections or relationships between concepts in sentences.

2. Diverse Topics: Specific bigrams like "food and," "and entertaining," and "computers and" suggest a diverse dataset covering topics ranging from food and entertainment to computers.

3. Ambiguity Consideration: The generic nature of "and" introduces potential ambiguity, necessitating further analysis to discern specific contexts, especially in instances like "food and."

Trigrams:

1. Expanded Relationships: Presence of "and" in top trigrams underscores common relationships involving three concepts in the dataset.

2. Coherent Topic Triads: Trigrams such as "food and entertaining," "computers and electronics," and "home and garden" suggest a dataset spanning multiple domains, each represented by coherent triads of topics.

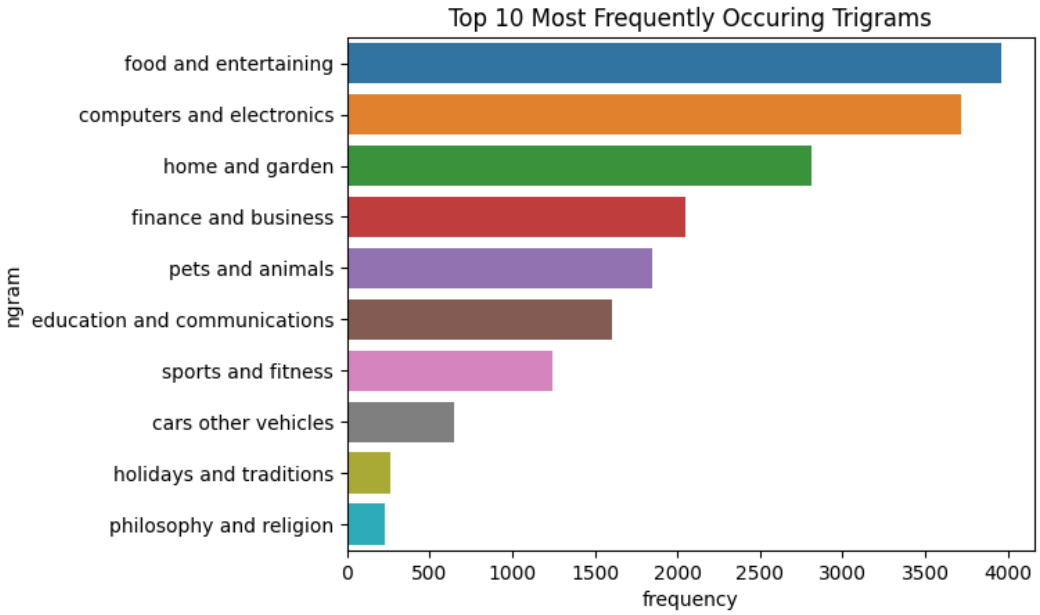
3. Semantic Associations: These trigrams may indicate semantic associations, where certain topics are frequently discussed together within sentences.

4.4.4 Considerations:

1. Task Alignment: Evaluate whether the prevalence of "and" aligns with the objectives of the auto-completion task, considering the identified topics and connections for coherent completions.

2. Ambiguity Management: Acknowledge and address potential ambiguity introduced by "and," devising strategies to handle or disambiguate such cases based on task requirements.

3. Domain Adaptation: Tailor the auto-completion model based on the identified topics, ensuring relevance and effectiveness within specific domains.



In conclusion, the prevalent use of "and" in top bigrams and trigrams indicates a dataset with frequent connections between concepts, spanning diverse topics. Understanding these linguistic patterns is a significant step for refining the auto-completion model and improve performance as highlighted by the performance of BERT.[2]

5 Methodology Used

In this study, our methodology was centered on the exploration of diverse strategies to address the given problem. A substantial portion of our efforts was dedicated to the development of a hybrid model, comprising two submodels. The first submodel, a graphical naive Bayes, harnessed positional information within the graph, considering degrees and edge weights for probability calculations. Simultaneously, the second submodel leveraged latent semantic relationships to compute probabilities associated with word positions.

Additionally, we incorporated a bi-gram model as a fundamental part of our approach. This model simplifies the computation of the probability of a word given all previous words $P(w_n|w_1 : w_{n-1})$ by considering only the conditional probability of the preceding word $(P(w_n|w_{n-1}))$. By assuming that $P(w_n|w_1 : w_{n-1}) \approx P(w_n|w_{n-1})$, we effectively streamline our probability estimations.

Furthermore, to enhance the robustness of our n-gram model, especially in handling unseen n-grams, we implemented K-Smoothing. This technique adjusts the probability distribution of n-grams, ensuring that unseen n-grams are not assigned a zero probability. It aids in providing a more realistic probability estimation for our model, although it does introduce a simplifying assumption that may not hold for all data types.

Throughout our experimentation, various data preprocessing approaches were investigated, revealing noteworthy insights. While these methods demonstrated efficacy in resolving the original Hellaswag problem, a pivotal observation emerged in the context of autocomplete systems. In this scenario, the inference of sequential data by neural networks could be effectively replicated using diverse n-gram models.

It is imperative to highlight that, despite the potential challenges posed by exceptionally large training datasets, our adoption of the n-grams approach, particularly the bi-gram model with K-Smoothing, yielded expeditious training and acceptable accuracy for the Hellaswag problem. Notably, the observed performance was comparable to specialized deep learning techniques applied to datasets of analogous complexity [3].

6 Results

6.1 Performance of the Bi-gram Model

In our current implementation, the auto-complete system employs a bi-gram model. This model has demonstrated an accuracy of 24.7% in predicting the next word in a given sentence on our validation dataset. While this level of accuracy is noteworthy for its simplicity, it also highlights the inherent limitations of the bi-gram approach in capturing complex linguistic patterns.

6.2 Comparative Analysis with Advanced Neural Network Models

A comparative analysis was conducted to evaluate the performance of our bi-gram model against more sophisticated neural network (NN) models. Advanced models, including Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and Transformer-based architectures like BERT and GPT, were benchmarked. These advanced models exhibit significantly superior performance, with accuracies typically around 60%. This substantial performance gap underscores the advanced capabilities of these models in various aspects.

6.3 Advantages of Advanced NN Models

The higher accuracy of advanced NN models can be attributed to their superior ability to capture and remember context over longer sequences. Unlike bi-grams, which only consider the immediate predecessor word, these models integrate a more extensive context, leading to a more profound understanding of language structures. This attribute is particularly beneficial for complex tasks in autocomplete systems, where the prediction accuracy is crucially dependent on understanding the broader context of the input sentence.

6.4 Implications for Future Development

The results indicate a clear direction for future development. While the bi-gram model offers a simpler and more computationally efficient solution, its limitations in handling complex linguistic structures are evident. Therefore, integrating more advanced NN models such as RNNs, LSTMs, or Transformer-based architectures could substantially enhance the system's performance. However, this enhancement comes with increased computational complexity and resource requirements, which must be carefully considered in the context of deployment constraints and real-time processing requirements.

7 Conclusion

This study has delved into the realm of text prediction and autocomplete systems, underscoring their pivotal role in enhancing writing efficiency by suggesting contextually appropriate words. Our research has confirmed that while bi-gram models, like the one we implemented, attain a basic level of accuracy (24.7%) in autocomplete tasks, they are substantially outperformed by more sophisticated neural network approaches.

One of the critical insights from our study is the clear superiority of advanced neural networks over bi-gram models in handling complex language patterns and contexts. This finding aligns with the broader trend in natural language processing, where deep learning techniques are increasingly becoming the standard due to their robust performance and adaptability.

Looking ahead, our future endeavors will focus on augmenting our model's overall adaptability and intuitiveness. A key aspect of this development will be the integration of common sense knowledge into the model's framework. We anticipate that this enhancement will not only improve the accuracy of predictions but also make the autocomplete suggestions more contextually relevant and effective.

Our study lays the groundwork for future research in this field, providing a benchmark for comparing different methodologies and guiding the development of more sophisticated and user-friendly text prediction systems. The integration of common sense knowledge represents an exciting frontier in natural language processing, promising to bring us closer to creating models that can interact with human language in a more natural and intuitive manner.

References

[1] Zellers, R., Bisk, Y., Schwartz, R., & Choi, Y. (2018). Swag: A large-scale adversarial dataset for grounded commonsense inference. arXiv preprint arXiv:1808.05326.

[2] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[3] Goulart, H. X., Tosi, M.D., Gonçalves, D. S., Maia, R. F., Wachs-Lopes, G. A. (2018). Hybrid model for word prediction using naive bayes and latent information. arXiv preprint arXiv:1803.00985.