

AAIB Assignment 6

Acknowledgement: ChatGPT was used both in the generation of text and code

Task 1

How can businesses mitigate the impact of hallucinations in language models to ensure reliable and accurate use of AI in critical decision-making processes? When answering your question, link the justification or suggestions to the lecture 6 content. (Min 250 words, Max 1 A4 page)

In today's rapidly advancing technological landscape, businesses frequently employ artificial intelligence (AI) tools like ChatGPT to enhance decision-making processes. However, these AI models, particularly in natural language processing (NLP), sometimes generate 'hallucinations'—false or misleading outputs—which pose significant challenges, especially in critical decision-making scenarios. Understanding and mitigating these risks is essential for leveraging AI effectively without compromising decision quality.

Firstly, one effective strategy is enhanced input design, commonly referred to as prompt engineering. This involves crafting detailed, clear, and context-specific prompts that guide AI models to generate more accurate and relevant responses. Given AI models' literal interpretation of input, precise prompts can significantly reduce misunderstandings and erroneous outputs, thus enhancing output reliability.

Secondly, implementing layered validation mechanisms is crucial. This approach involves reviewing AI-generated outputs through additional automated systems or manual checks by trained staff. This step is vital because, despite their advanced capabilities, AI models do not fully grasp the nuances of natural language, making them prone to errors. External validation layers can thus act as a safeguard, catching and correcting errors before they impact decision-making.

Moreover, continuous monitoring and structured feedback play a pivotal role. Regular assessment of AI tool performance and collecting feedback can help identify recurrent patterns in errors, which can then be mitigated through refined prompts or adjustments in validation processes. This continuous improvement is crucial in adapting AI tools to better serve the dynamic needs of businesses.

Training end-users and decision-makers on the capabilities and limitations of AI models is also indispensable. Such educational initiatives ensure that users have realistic expectations and are better prepared to interpret AI-generated insights correctly.

Lastly, whenever possible, using structured data inputs rather than free-form text can significantly reduce ambiguities, guiding AI models towards more accurate data processing and minimizing the risk of errors.

By adopting these strategies, businesses can effectively mitigate the risks associated with AI hallucinations in critical decision-making. This ensures that AI tools are not just used, but are used wisely, enhancing decision-making processes without compromising on reliability and accuracy. These approaches, grounded in an understanding of NLP's capabilities and limitations, are essential for any business aiming to integrate AI into their core operational processes.

Task 2

Based on the notebook of lecture 6:

1. *Re-implement the sentiment analysis with the US Airline dataset using Tf-id vectors as input instead of the pure bag-of words approach.*
-

Figure 1 shows sentiment analysis using TF-IDF vectors as features and an SVM classifier. It begins by loading a dataset from a CSV file containing tweets, each labeled with sentiment. The data is restricted to necessary columns: 'tweet_id', 'airline_sentiment', and 'text'. A TF-IDF Vectorizer is initialized and fitted to convert the text data into a matrix of TF-IDF features. The dataset is then split into training and test sets with 20% of the data reserved for testing. An SVM with RBF kernel is trained on the TF-IDF vectors. The training process, however, warns of non-convergence, suggesting that the maximum iteration limit was insufficient to reach an optimal solution.

tf-idf for sentiment analysis (Task 2)

```
[13] # Load dataset
data = pd.read_csv('datasets/Tweets.csv')
data = data[['tweet_id', 'airline_sentiment', 'text']]

[14] # Initialize TF-IDF Vectorizer
tfidf_vectorizer = TfidfVectorizer()
tfidf_vectorizer.fit(data.text)
X_tfidf = tfidf_vectorizer.transform(data.text)

[15] # Split data into training and test sets
X_train_tfidf, X_test_tfidf, y_train, y_test = train_test_split(
    X_tfidf, data.airline_sentiment, test_size=0.2, random_state=0)

[16] # Train SVM with the TF-IDF vectors
clf_svm_tfidf = svm.SVC(max_iter=1000, gamma='scale', kernel='rbf', random_state=0)
clf_svm_tfidf.fit(X_train_tfidf, y_train)
```

/usr/local/lib/python3.10/dist-packages/sklearn/svm/_base.py:299: ConvergenceWarning: Solver terminated early (max_iter=1000). Consider increasing the number of iterations (max_iter) or the regularization parameter (C).
warnings.warn(

```
SVC
SVC(max_iter=1000, random_state=0)
```

Figure 1 Sentiment Analysis using tf-idf

-
2. *Compare the classification accuracy of both approaches.*

3. Explain in a few sentences what could be the reason for the different results.

The Accuracy and Confusion Matrix in Figure 2 show that TF-IDF outperforms Bag of Words.

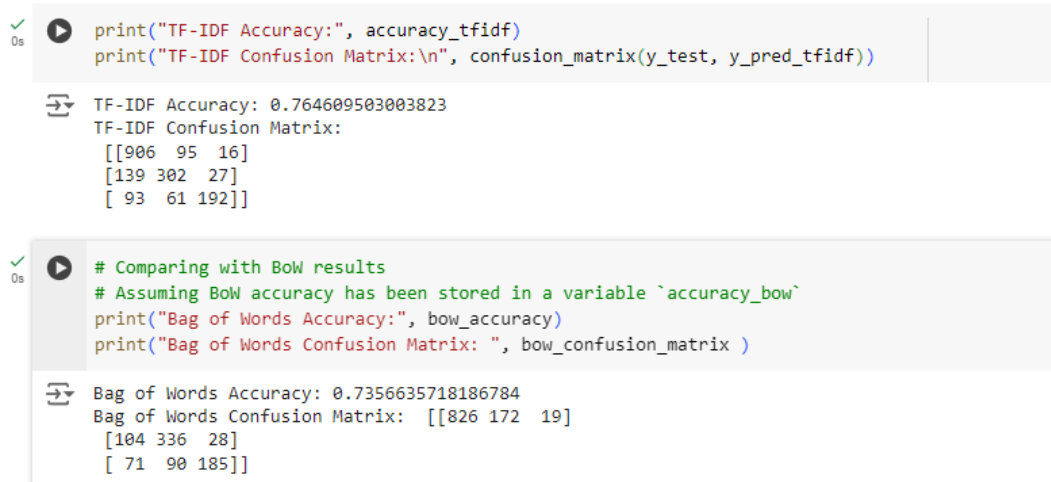


Figure 2 Accuracy and Confusion Matrix for BoW and TF-IDF

- TF-IDF: Achieved 76.46% accuracy.
- BoW: Achieved 73.57% accuracy.
- The confusion matrices for both methods reveal some insights into their classification behavior:
 - TF-IDF Confusion Matrix shows better handling of the main class (first class), with fewer misclassifications into the second class compared to BoW.
 - BoW Confusion Matrix demonstrates a tendency to more frequently misclassify instances of the main class into the second class, possibly contributing to its lower overall accuracy.

Word Weighting: TF-IDF weighs words not just on their occurrence in a single document but across all documents. It reduces the weight of terms that appear very frequently across the dataset (which might be less informative) and increases the weight of terms that are rare but could be more telling of sentiment. This results in a model that can better distinguish between the relevance of different terms for classification.

In contrast, BoW counts every occurrence equally without discriminating between the word's importance across different texts. It treats every term with the same importance, which can dilute the influence of truly indicative terms amidst commonly occurring but less informative words.

TF-IDF's ability to emphasize rare but potentially significant words could help in accurately classifying texts, especially those that might hinge on less common terms that are strongly indicative of sentiment. This can be particularly beneficial in distinguishing between positive and negative sentiments, where specific adjectives or adverbs could play a pivotal role.

TF-IDF might also offer better generalization over unseen data because it inherently accounts for the idiosyncrasies of language usage across different documents, avoiding overfitting to particular frequent terms that might not actually carry much sentiment-specific information.

Task 3

Select any business-related (ended) competition from kaggle1. Look at one of the top 3 winners of the competition and analyze their answers proposing a schematic framework of what you would like to investigate/test differently (e.g., a different model, an integration of models, a different DPT process, etc). Explain the context of the competition, the strongest points of the solution, and the main limitations. Besides the short text with this description your solutions must include a figure with a step-by-step plan of what you would like to experiment further in the solution to potentially make it better.












| # | △ | Team | Members | | Score | Entries | Last | Solution |
|---|------|---|---|---|----------|---------|------|---|
| 1 | ▲ 8 |  |  |  | 0.987824 | 331 | 5mo |  |
| 2 | ▲ 15 | Guanshuo Xu |  |  | 0.983412 | 74 | 5mo |  |
| 3 | ▲ 12 | nlp team |  |  | 0.974994 | 280 | 5mo |  |

Figure 3 LLM Detection Contest private leaderboard on Kaggle [1]

Team “” utilized a comprehensive strategy involving a diverse set of data sources and machine learning models. Here’s a brief breakdown of their approach:

Data Preparation:

Utilized a mix of ~160k examples, with ~40k human-written.

Enriched the training set with a variety of text sources, including synthetic datasets and publicly shared datasets.

Modeling Strategy:

Employed an ensemble of diverse approaches.

Focused on complexity and diversity in model training, including instruction tuning and contrastive decoding.

Applied a ghostbuster approach using different models like llama 7b and tiny llama 1.1B.

Experimented with training a Deberta-v3-small from scratch and using ranking loss with Deberta-v3-large models.

Performance:

Achieved high accuracy on private and public leaderboards, showcasing the effectiveness of their ensemble approach.

Strengths and limitations:

The strengths of the solution lie in its comprehensive use of diverse data and advanced modeling techniques, which likely contributed to its high performance. However, these same factors also introduce challenges such as high dependency on data quality and volume, potential scalability issues due to complex models, and the risk of overfitting to the specific nuances of the competition dataset. These limitations could affect the solution's effectiveness outside the controlled environment of a competition, especially in dynamic real-world applications where data and conditions can vary significantly.

| Aspect | Strengths | Limitations |
|--------------------|--|--|
| Data and Models | Utilized a diverse range of data sources and models, enhancing detection of AI-generated texts. | Dependency on large and diverse datasets, which may not always be available. |
| Methodology | Employed an ensemble approach, blending multiple models' outputs to improve accuracy. | Complex ensemble models can be resource-intensive and may not scale well in real-time scenarios. |
| Technological Edge | Advanced techniques like instruction tuning and contrastive decoding tailored models for the task. | Potential for model overfitting due to intense customization to the competition's dataset. |

Proposed Experimental Framework

To potentially enhance the solution, I propose the following schematic framework for further experimentation:

Integration of Transformer Variants:

Experiment with newer variants of transformers that are specifically designed for efficiency and performance, such as the GPT-Neo or GPT-J, which might offer better or comparable performance with reduced computational demand.

Cross-Model Validation:

Implement cross-validation techniques across different model architectures to ensure the robustness and generalizability of the models.

Incremental Training:

Employ incremental learning approaches to continually adapt the model to new forms of AI-generated texts as they evolve.

Automated Retraining Pipeline:

Develop an automated system to retrain models periodically with new data or pseudo-labeled data to keep the models updated with the latest text generation trends.

References:

- [1] “LLM - Detect AI Generated Text | Kaggle.” Accessed: Jun. 11, 2024. [Online]. Available: <https://www.kaggle.com/competitions/llm-detect-ai-generated-text/discussion/473295>