Language Model Development For Flight Booking

VItaly Horoshev

November 2023

Abstract

This project addresses the challenge of creating a language model tailored for handling flight booking-related queries. A specialized dataset was generated, consisting of varied user expressions categorized into three intents: booking, cancelling, and querying flights. A BERT model was fine-tuned on this dataset and integrated with a mock API functions to simulate a flight booking system. The model's ability to understand different user inputs and its interaction mechanism, including follow-up questions for unclear intents, are key highlights.

1. Introduction

The advancement of natural language processing (NLP) technologies has opened up possibilities for automated systems that can understand and respond to human language effectively. In this project, i leverage these advancements to create a language model specifically designed for handling flight booking queries. This includes not only recognizing the user's intent but also responding appropriately or seeking clarification when necessary.

1.1 Team

This project was a solo endeavor. I was responsible for all aspects, including dataset creation, model training, developing the interaction system, and integrating it with the mock API functions.

2. Related Work

In exploring the current landscape of NLP in flight booking and management systems, it's evident that most existing solutions have limitations. They primarily rely on basic keyword recognition or simple rule-based interactions, lacking the ability to understand more complex, natural language inputs. To understand the current state and the potential for advancements, let's look at several real-life examples:

Automated Online Customer Support in Airlines like American Airlines
These systems, such as the one used by American Airlines, generally provide automated responses to customer inquiries through chatbots. However, they often struggle when faced with complex or non-standard queries. For instance, a customer asking about flight changes due to unforeseen events like a weather crisis might receive a generic response or be redirected to a FAQ page, which

can be frustrating and unhelpful. This reflects a need for more context-aware and adaptable NLP systems in customer service.

Voice Recognition Systems in Airline Call Centers, as seen in Delta Airlines Delta Airlines, for example, employs voice-activated systems for customer service. However, these systems can be quite rigid, often requiring users to speak specific phrases to navigate the menu, which can lead to a frustrating experience. Users might find themselves stuck in loops, trying to get the system to understand their request. This highlights the potential for more sophisticated voice recognition technology that can interpret and respond to a wider range of spoken language.

Mobile Check-in Applications like Southwest Airlines' App Southwest Airlines offers a mobile app for services like check-ins and accessing flight information. While these apps are convenient, their interaction capabilities are limited to predefined user inputs and lack the ability to process free-form natural language. This suggests an area for growth where advanced NLP could be integrated to make these interactions more intuitive and conversational.

Online Travel Booking Platforms such as Expedia Platforms like Expedia allow users to book flights through a web interface but typically don't support conversational interactions. Users have to navigate through a set of predefined options, which can be less efficient and more time-consuming than a natural language conversation. Integrating NLP into such platforms could transform them into more interactive, conversational systems, streamlining the booking process.

These examples demonstrate a clear opportunity for NLP models, particularly advanced ones like BERT, to enhance the interaction between flight-related services and their users, providing more natural, responsive communication experiences.

3. Dataset

For this project, I had to create a unique dataset from scratch, as there wasn't anything available that exactly met my needs. The aim was to simulate the different ways people might talk about booking, canceling, or asking about flights. Here’s how I did it:

3.1 Base Phrases and Variations I started by brainstorming a list of straightforward phrases for each of the three categories: booking, canceling, and flight queries. These base phrases were simple, like "book a flight", "cancel my flight", and "show available flights". The real work was in adding variety to these phrases to reflect how people naturally speak or type. To do this, I:

- **Rephrased Sentences:** I changed the structure of sentences while keeping their original meaning. For instance, turning "I want to book a flight" into "Can I book a flight?"
- **Used Synonyms:** I switched out key words for their synonyms to create similar sentences with different wording, such as replacing "book" with "reserve".
- **Added Extra Words:** I included words that people often use in conversation, like "please", or transformed statements into questions to add more natural variation.

3.2 Ensuring Diversity The goal was to make sure the dataset covered a wide range of expressions without being biased towards any particular type of query. Balancing the dataset across the three categories was crucial to avoid any skew in the model training.

3.3 Reflection on Dataset Quality Interestingly, this manual approach to dataset creation turned out to be much more effective than my initial attempts at using automated NLP techniques for dataset generation. Those methods, while faster, tended to produce data that was either too repetitive or really poor in quality. The manual process, albeit more time-consuming, resulted in a dataset of significantly higher quality – more reflective of actual human conversations and inquiries about flights.

3.4 Final Dataset After this process, I ended up with a dataset of 1,000 entries, each labeled according to its intended category. This number struck a balance between having enough examples for effective model learning and keeping the dataset to a size I could manage and review manually.

4. Model Description

In this project, I leveraged the BERT (Bidirectional Encoder Representations from Transformers) model, a powerful tool in natural language processing. Here's a breakdown of the technical details involved in adapting and training this model for the flight booking task:

4.1 BERT Model Setup

- **Model Selection:** I selected the 'bert-base-uncased' variant of BERT for its balance between size and performance. This version of BERT, pre-trained on a large corpus of English data, provides a deep understanding of language context.
- **Tokenizer:** The BERT tokenizer was used to preprocess the text. This tokenizer breaks down the input text into tokens that the model can understand, including converting words to lower case, as the model is an 'uncased' version.

4.2 Data Preprocessing and Tokenization

- **Tokenization Process:** Each phrase in the dataset was tokenized to convert it into a format suitable for BERT. This involved encoding the phrases into tokens, adding special tokens (like **[CLS]** and **[SEP]**), truncating sentences to a maximum length, and padding shorter sentences.
- **Attention Masks:** Along with tokenized input, attention masks were created to let the model differentiate between the actual content and the padding.

4.3 Model Configuration and Training

- **Input Layers:** The input to the model consists of the tokenized sentences and their corresponding attention masks.
- **Output Layer Modification:** BERT's pre-trained model was customized with an additional output layer for classification, tailored to categorize inputs into three classes - booking, canceling, and querying flights.
- **Training Parameters:** I used TensorFlow with Keras functionalities for training. The training involved several parameters:
- **Optimizer:** Adam optimizer with a learning rate of 2e-5.
- **Loss Function:** Categorical cross-entropy, as this is a multi-class classification problem.
- **Batch Size and Epochs:** A batch size of 32 and 3 training epochs, balancing the time cost and model performance.
- **Performance Monitoring:** During training, I monitored the loss and accuracy metrics to track the model's learning progress. The training was executed on Google Colab with GPU support, significantly speeding up the training process.

4.4 Handling Ambiguity with Confidence Threshold

- **Confidence Assessment:** The model predicts a probability distribution across the three classes. I implemented a confidence threshold to determine if the model's prediction was reliable. If the highest probability was below this threshold, the system triggered a follow-up question.

4.5 Integration with a Mock API

- The trained model was then integrated into a basic Python application with mock API functions representing flight booking, cancellation, and inquiry actions. This step was crucial to simulate a real-world application.

5. Experiments

5.1 Model Training

The experimental phase of this project was crucial in ensuring the effectiveness of the BERT model for the specific task of understanding flight-related queries. Here's a detailed look at the experimental setup, the training process, and the results obtained:

5.1 Model Training and Validation The process of training and validating the model involved several steps:

- **Dataset Split:** I divided the dataset into training and validation sets, using an 80-20 split. This approach ensured that the model was trained on a majority of the data while reserving a portion for validation to assess its performance on unseen data.
- **Model Setup:** Using TensorFlow, the BERT model was set up with the additional classification layer. This layer was crucial for adapting the pre-trained BERT model to our specific three-class classification task.
- **Training Loop:** The training involved feeding batches of tokenized text data and corresponding labels to the model. I used gradient descent for optimization, adjusting the model's weights based on the computed loss from each batch.
- **Monitoring Metrics:** During training, I kept a close eye on accuracy and loss metrics. The training process showed rapid improvements in accuracy, indicating that the model was effectively learning from the training data.

5.2 Handling Overfitting Concerns Initially, there was a concern about potential overfitting, as the model quickly reached very high accuracy levels. To address this:

- **Regularization Techniques:** Although not initially included, I was prepared to implement dropout or weight decay as regularization methods if overfitting became apparent in the validation phase.
- **Dataset Quality:** Upon further analysis, the high accuracy was attributed to the quality and consistency of the dataset, along side with high capabilities of BERT model, rather than overfitting. The manual creation process of the dataset ensured that it was representative and comprehensive, aiding in effective model training.

5.3 Validation and Results In the validation phase, the model maintained high accuracy, indicating good generalization to new data:

- **Accuracy Measurement:** The model achieved close to 100% accuracy on the validation set, mirroring its performance on the training set.
- **Interpretation of Results:** This high level of accuracy on both sets suggested that the model was well-tuned to the dataset and could reliably classify user queries into the correct categories.

- **Reflections on Model Performance:** The results were promising, showing that the fine-tuned BERT model, coupled with a carefully crafted dataset, was highly effective for this specific application. It demonstrated the model's capability to handle various phrasings and expressions in flight booking contexts.

6. Results

The results from training and validating the BERT model were quite insightful. Here are the key observations:

- **High Accuracy:** The model consistently achieved near-perfect accuracy on both the training and validation datasets. This exceptional performance is indicative of the model's ability to accurately understand and categorize various flight-related queries.
- **Handling Ambiguity:** One of the notable results was the model's capability to handle ambiguous inputs. The implementation of a confidence threshold for follow-up questions proved effective. In cases where the model was less certain, it successfully prompted for more information, closely mimicking a natural human interaction.
- **Realistic Simulations with Mock API:** The integration with the mock API functions worked seamlessly. The model's predictions triggered the appropriate API responses, providing realistic simulations of booking, cancelling, or querying about flights.
- **Reflection on User Queries:** Testing with a variety of user queries, including those not present in the training dataset, showed that the model could generalize well. This ability is crucial for real-world applications where user inputs can be highly unpredictable.

7. Conclusion and Future Work

7.1 Conclusion This project demonstrates the potential of fine-tuned BERT models in specialized domains like flight booking services. The success of the model in understanding a range of user queries with high accuracy is a promising step toward more intelligent and responsive AI systems in customer service.

- **Effective Dataset Creation:** The manual approach to dataset creation, though time-consuming, was crucial in achieving high-quality training data that contributed significantly to the model's performance.
- **Model's Versatility:** The ability of the BERT model to adapt to this specific task underscores its versatility and effectiveness in handling complex language understanding tasks.

7.2 Future Work Looking ahead, the project could be enhanced in several ways:

- **Advanced Interaction Capabilities:** Developing the model to manage more nuanced conversations, like understanding context over multiple messages or handling interruptions and corrections by users.
- **Customization for Other Domains:** Adapting the approach for different domains or services, such as hotel bookings or customer service in other industries, to test the model's adaptability.
- **User Experience Optimization:** Improving the model's ability to provide more personalized responses based on user history or preferences, enhancing the overall user experience.
- **Voice Integration:** Exploring the integration of voice recognition technology, allowing users to interact with the system through spoken language.
- **Real-Time Learning and Adaptation:** Implementing machine learning techniques that allow the model to learn and adapt from ongoing user interactions, enhancing its performance and accuracy over time.

In conclusion, this project lays the groundwork for more advanced and user-friendly AI-driven systems in the airline industry and possibly other customer service domains.

References

- **American Airlines Chatbot**

URL: https://www.aa.com/i18n/customer-service/contact-american/american-customer-service.jsp

- **Delta Airlines Voice Recognition System**

URL: https://www.delta.com/

- **Southwest Airlines Mobile App**

URL: https://www.southwest.com/html/air/products/mobile.html

- **Expedia Travel Booking**

URL: https://www.expedia.com/

- **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding by Jacob Devlin et al.**

Original paper introducing BERT.
URL: https://arxiv.org/abs/1810.04805

- **Transformers Library by Hugging Face**

Documentation and tutorials for using the Transformers library.
URL: https://huggingface.co/transformers/

- **TensorFlow Documentation**

Official TensorFlow documentation for model implementation.
URL: https://www.tensorflow.org/

- **Google Colaboratory**

Information about using Google Colab for machine learning projects.
URL: https://colab.research.google.com/

- **Natural Language Processing with Python by Steven Bird, Ewan Klein, and Edward Loper**

A comprehensive guide to NLP basics, useful for understanding dataset creation and processing.
Book URL: https://www.nltk.org/book/

- **Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville**

For understanding the fundamentals of deep learning.
Book URL: https://www.deeplearningbook.org/

- **Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems by Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, and Harshit Surana**

Practical insights into applying NLP in real-world scenarios.
Book URL: https://www.oreilly.com/library/view/practical-natural-language/9781492054047/

- **Evaluating BERT for Text Classification and Question Answering by Yi Tay et al.**

For insights on evaluating BERT in different tasks.
URL: https://arxiv.org/abs/1904.03329

- **Pandas Documentation**

For data manipulation and analysis during dataset preparation.
URL: https://pandas.pydata.org/

- **Scikit-Learn Documentation**

Useful for any additional machine learning tasks and data processing.
URL: https://scikit-learn.org/stable/