A Hybrid Framework for Personalized Scheduling Using Natural Language Processing and Collaborative Filtering.

Hitesh Gehani[1], Shubhangi Rathkanthiwar[2]
[1]Research Scholar, Department of Electronics Engineering, Yeshwantrao Chavan College of Engineering, Nagpur, India.
[2]Professor, Department of Electronics Engineering, Yeshwantrao Chavan College of Engineering, Nagpur, India.

Abstract:

Introduction: In an era characterized by an abundance of digital information, the task of sifting through a vast array of events and appointments to find personalized and relevant scheduling solutions has become increasingly complex. In today's fast-paced, information-overloaded world, managing schedules efficiently remains a critical challenge. Traditional calendar systems often fail to accommodate personalized scheduling needs, leaving room for inefficiencies. This study aims to address this gap by developing an NLP-Enhanced Smart Calendar that leverages natural language processing (NLP) techniques for accurate, tailored scheduling. The objective of this study is to create a smart calendar system that integrates user preferences, historical scheduling patterns, and event categories to generate personalized scheduling recommendations using NLP and collaborative filtering algorithms. The proposed system integrates user preferences, historical scheduling patterns, and event categories to generate personalized scheduling recommendations. A combination of collaborative filtering and NLP algorithms continuously refines these suggestions. The system adapts dynamically to evolving user preferences and scheduling patterns.

Keywords: Natural Language Processing (NLP), Smart Calendar Systems, Event Scheduling, Named Entity Recognition (NER), Date and Time Parsing, User-friendly Calendar Apps, Voice Commands, Machine Learning, Automatic Scheduling, Human-Computer Interaction, Context- aware Event Management.

1. Introduction

In today's fast-paced digital age, the conventional methods of managing In an era marked by increasing reliance on digital tools, the need for intuitive and efficient scheduling solutions has never been more pressing. Traditional calendar applications often fall short in meeting the dynamic needs of users, prompting the development of innovative alternatives. This paper explores the creation of a groundbreaking solution: the NLP-Enhanced Smart Calendar. At the core of this project lies the implementation of cutting-edge Natural Language Processing (NLP) techniques, prominently featuring the BERT base uncased model. This sophisticated model serves as the cornerstone of the smart calendar, enabling it to interpret and respond to natural language commands with unparalleled accuracy and efficiency.

In addition to leveraging advanced NLP models, the development of the NLP-Enhanced Smart Calendar involves the utilization of modern web development technologies. The frontend of the application is built using React.js, a popular JavaScript library known for its flexibility and robustness. Complemented by Node.js for backend development and Streamlit for data visualization, the smart calendar offers a seamless and intuitive user experience across various platforms. Throughout this thesis, a comprehensive methodology is employed, encompassing requirement analysis, data collection, model development, integration, testing, and deployment. By adhering to this systematic approach, the project aims to deliver a sophisticated scheduling solution that meets the diverse needs of users while staying at the forefront of technological innovation. Integration of Natural language processing (NLP) into the Calendar, revolutionizing event scheduling and providing a user-friendly interface.

The NLP-Enhanced Smart Calendar project seeks to enhance traditional calendar apps by incorporating Natural Language Processing (NLP) capabilities. Current calendar apps often pose usability challenges and lack user-friendly features. This project aims to simplify interactions by allowing users to communicate with their calendars using everyday language. Instead of typing commands, users can simply speak naturally to schedule meetings, set reminders, and manage tasks. The objective is to streamline the process of scheduling events and organizing tasks, making it as effortless as having a conversation. By eliminating the need for complex menus and manual input, the NLP Smart Calendar aims to enhance user experience and productivity. By leveraging NLP technology, the project aims to make scheduling easier for individuals from various backgrounds, including professionals managing busy schedules, students balancing academic and extracurricular commitments, and individuals seeking better organization in their personal lives. The ultimate goal is to create calendar apps that are intuitive and user-friendly for all users, regardless of their level of technological expertise.

The development of a conversational text-based model marks a significant advancement in the realm of calendar management. This innovative model has been designed to understand natural language input and extract crucial information such as the name of the person, date, time, and venue from the text. By leveraging sophisticated techniques in Natural Language Processing (NLP), the model seamlessly interprets conversational text, enabling users to interact with their calendars in a more intuitive and efficient manner.

The workflow of the model begins with input in conversational text format, wherein users can express their scheduling needs in a natural and unrestricted manner. This text is then processed by the model,

which employs advanced algorithms to identify and extract pertinent details, including the name of the individual involved, the specific date and time of the event, and the venue where it is to take place. Once this information is successfully extracted, it is seamlessly integrated into the user's calendar, ensuring that important events and appointments are accurately marked and organized. The overarching goal of this model is to simplify the process of calendar management by eliminating the need for manual input and streamlining the scheduling process. By harnessing the power of NLP to understand conversational text and extract key information, this innovative solution empowers users to interact with their calendars effortlessly, ultimately enhancing productivity and user satisfaction.

2. Literature Review

Calendar data has become an important context source of user information due to the popularity of online calendar services such as Google Calendar and Outlook Calendar. According to a research study conducted by Promotional Products Association International in 2011, about 40\% of people referred to calendars on their computers, and about 22\% of people used their mobile calendars every day. As more people use online calendar services, more detailed user information is becoming available. Event scheduling is one of the most common applications that uses calendar data. Similar to Gervasio et al. and Berry et al., we define event scheduling as suggesting suitable time slots for calendar events given user preferences and calendar contexts. However, even with the development of communication technology, event scheduling is still time-consuming. According to Konolabs Inc., the average number of emails sent between people to set the time for a meeting is 5.7.1 At the same time, the market for digital assistants is growing fast. Gartner, Inc. stated that by 2019, at least 25\% of households will use digital assistants on mobiles or other devices as primary interfaces of connected home services. Thus, it is important for digital assistants to effectively schedule users’events[1].

The Transformer follows this overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder, shown in the left and right halves of Figure 1, respectively Encoder and Decoder Stacks.

Encoder: The encoder is composed of a stack of N = 6 identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position wise fully connected feed-forward network. We employ a residual connection around each of the two sub-layers, followed by layer normalization . That is, the output of each sub-layer is LayerNorm(x + Sublayer(x)), where Sublayer(x) is the function implemented by the sub-layer itself. To facilitate these residual connections, all sub-layers in the model, as well as the embedding layers, produce outputs of dimension dmodel = 512.

Decoder: The decoder is also composed of a stack of N = 6 identical layers. In addition to the two sub- layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. Similar to the encoder, we employ residual connections around each of the sub-layers, followed by layer normalization. We also modify the self-attention sub-layer in the decoder stack to prevent positions from attending to subsequent positions. This masking, combined with fact that the output embeddings are offset by one position, ensures that the predictions for position i can depend only on the known outputs at positions less than i [2].

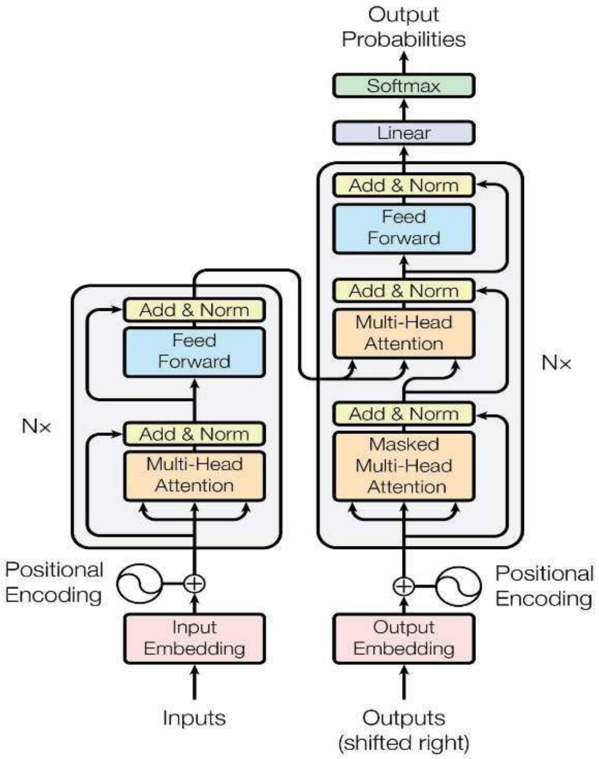


Figure 1. The Transformer - model architecture.

BERT: Framework: Pre-training and Fine-tuning
During pre-training, the model is trained on unlabeled data over different pre-training tasks. For fine- tuning, the BERT model is first initialized with the pre-trained parameters, and all of the parameters are fine-tuned using labeled data from the downstream tasks. Each downstream task has separate fine- tuned models, even though they are initialized with the same pre-trained parameters. The question- answering example in Figure 2 will serve as a running example for this section.

BERT's model architecture is a multi-layer bidirectional Transformer encoder based on the original implementation described in Vaswani et al. (2017) and released in the tensor2tensor library. Because the use of Transformers has become common and our implementation is almost identical to the original, we will omit an exhaustive background description of the model architecture and refer readers to Vaswani et al. (2017) as well as excellent guides such as *"The Annotated Transformer."*

In this work, we denote the number of layers (i.e., Transformer blocks) as L, the hidden size as H, and the number of self-attention heads as A. We primarily report results on two model sizes:

- **BERTBASE**
  L = 12, H = 768, A = 12, Total Parameters = 110M
- **BERTLARGE**:
  L = 24, H = 1024, A = 16, Total Parameters = 340M

BERTBASE was chosen to have the same model size as **OpenAI GPT** for comparison purposes. Critically, however, the **BERT** Transformer uses bidirectional self-attention, while the **GPT** Transformer uses constrained self-attention, where every token can only attend to context to its left.

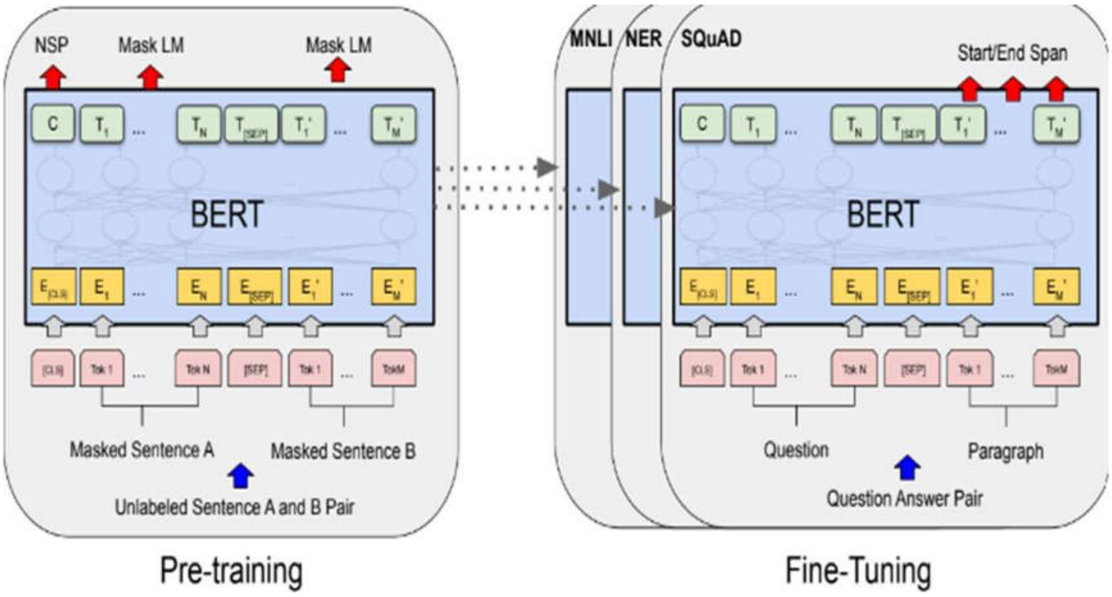


Figure 2. Overall pre-training and fine-tuning procedures for BERT.

Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).

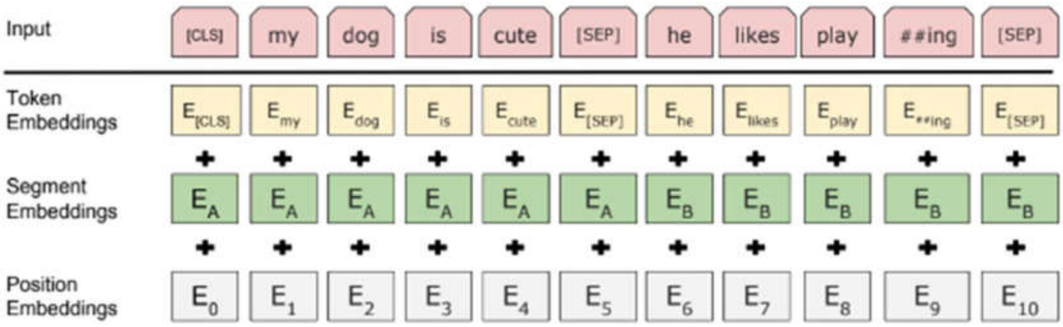


Figure 3. BERT input representation.

3. Methodology

The model focuses on leveraging advanced technologies like TensorFlow, Python, and Hugging Face for Natural Language Processing tasks. Central to the work is BERT, a transformer-based model, pre-trained on large datasets for feature extraction, and fine-tuned for specific tasks. The CoNLL 2003 dataset was used for named entity recognition, with BERT and SpaCy integrated for efficient information extraction.

Model Used: BERT base (uncased)

The Bidirectional Encoder Representations from Transformers (BERT) model, used in this research, is a pre-trained transformer-based model that excels at understanding complex natural language. BERT was pre-trained on a vast corpus of English text using a self-supervised learning approach, meaning it learned from raw, unlabeled data without human intervention. The pre-training involved two key objectives:

A. Masked Language Modeling (MLM): In this task, BERT randomly masks 15\% of the words in a sentence and then predicts the masked words based on the context. This approach differs from traditional models like recurrent neural networks (RNNs) and autoregressive models (e.g., GPT) by enabling BERT to consider the context from both the left and right of a given word, resulting in a bidirectional representation.

B. Next Sentence Prediction (NSP): During pre-training, BERT is fed pairs of sentences. In some cases, these sentences are consecutive in the original text, while in others, they are not. The model learns to predict whether the two sentences follow each other in the text. This task helps BERT capture relationships between sentences, enhancing its performance in tasks involving sentence-level understanding.

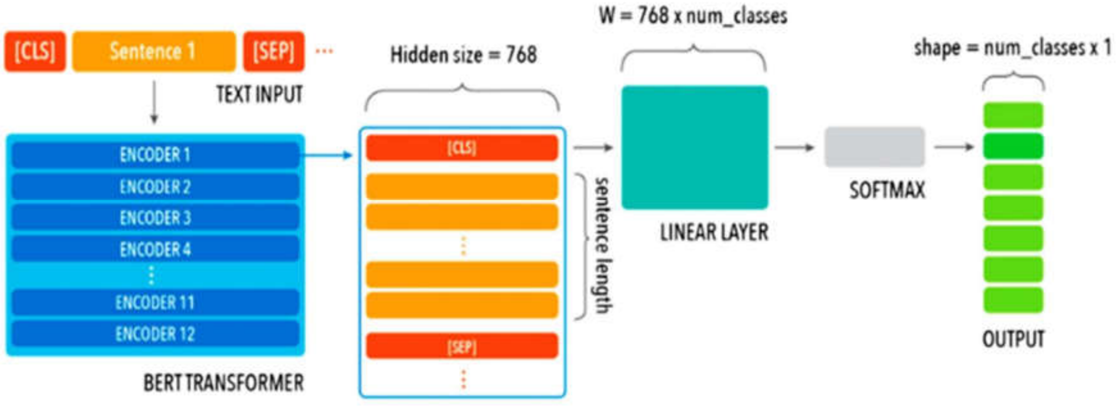
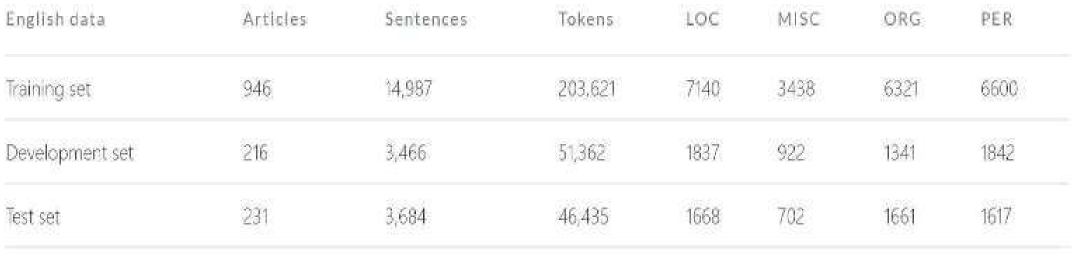


Figure 4. BERT base uncased model architecture.

Dataset

The CoNLL 2003 dataset is a benchmark dataset designed for named entity recognition (NER) tasks. The dataset was featured in Neural Architectures for Named Entity Recognition by Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, Chris Dyer April 2016. CoNLL-2003 is a named entity recognition dataset released as a part of CoNLL-2003 shared task: language- independent named entity recognition. The data consists of eight files covering two languages: English and German. For each of the languages there is a training file, a development file, a test file and a large file with unannotated data.



English data	Articles	Sentences	Tokens	LOC	MISC	ORG	PER
Training set	946	14,987	203,621	7140	3438	6321	6600
Development set	216	3,466	51,362	1837	922	1341	1842
Test set	231	3,684	46,435	1668	702	1661	1617

Figure 5. Dataset (CoNLL 2003) Features.

The project begins with a thorough requirement analysis phase. This involves gathering user feedback through interviews or surveys to understand their needs and expectations regarding event scheduling. Key features such as event creation, time and date parsing, and venue management are identified. Additionally, the feasibility of integrating BERT and SpaCy for information extraction is assessed during this phase to ensure that the chosen technologies align with project goals.

Following the requirement analysis, the technical feasibility assessment is conducted. This involves evaluating available technologies and frameworks for natural language processing (NLP) and calendar management. Specifically, the capabilities of BERT and SpaCy are examined to determine their suitability for extracting information such as person's name, time, date, and venue from user input.

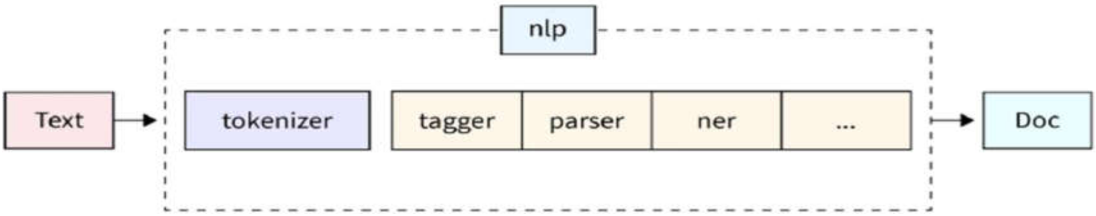


Figure 6. Representation of spaCy pipeline.

Once the technical feasibility is established, the design phase commences. Wireframes and mockups are developed to visualize the user interface, taking into consideration the integration of BERT and SpaCy for information extraction. Algorithms for text parsing using SpaCy are designed to extract date and time information from user input, while BERT models are utilized to extract person's name and venue details.

In the implementation phase, the front-end interface is developed using suitable web or mobile frameworks. BERT and SpaCy are integrated into the backend system to process user input and extract relevant event details. The application's features for event scheduling, editing, and deletion are implemented, leveraging the extracted information to manage events effectively.

Testing is conducted at various stages of development to ensure the robustness and accuracy of the application. Unit testing verifies the functionality of individual components, while integration testing validates the interaction between different modules. User acceptance testing gathers feedback on usability and identifies any issues that need to be addressed before deployment. Upon successful testing, the application is deployed on suitable platforms such as web servers or app stores. Compatibility with different devices and browsers is ensured, and documentation and user guides are provided to facilitate adoption.

Below in Figure 7 is the working of the Transformer encode: BERT, which stands for Bidirectional Encoder Representations from Transformers, is a powerful tool in Natural Language Processing (NLP). It's essentially a deep learning model that excels at understanding the complexities of human language.

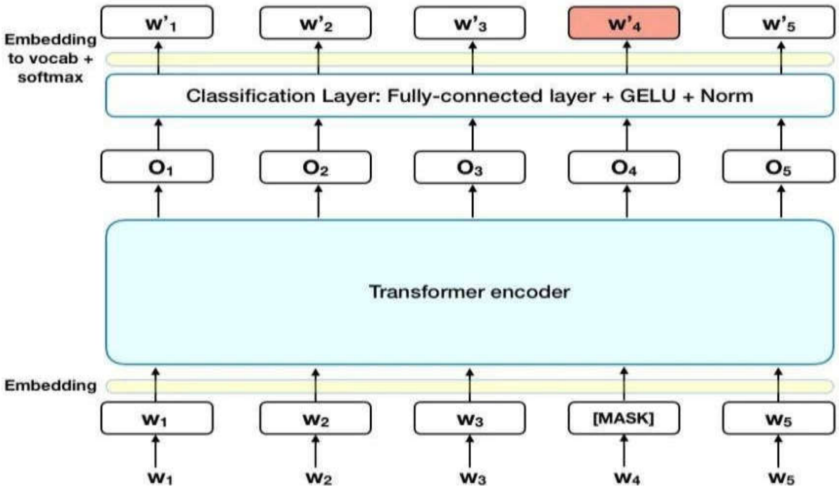

w'₁
w'₂
w'₃
w'₄
w'₅
Embedding to vocab + softmax
Classification Layer: Fully-connected layer + GELU + Norm
O₁
O₂
O₃
O₄
O₅
Transformer encoder
Embedding
w₁
w₂
w₃
[MASK]
w₅
w₁
w₂
w₃
w₄
w₅

Figure 7. Transformer encoder diagram.

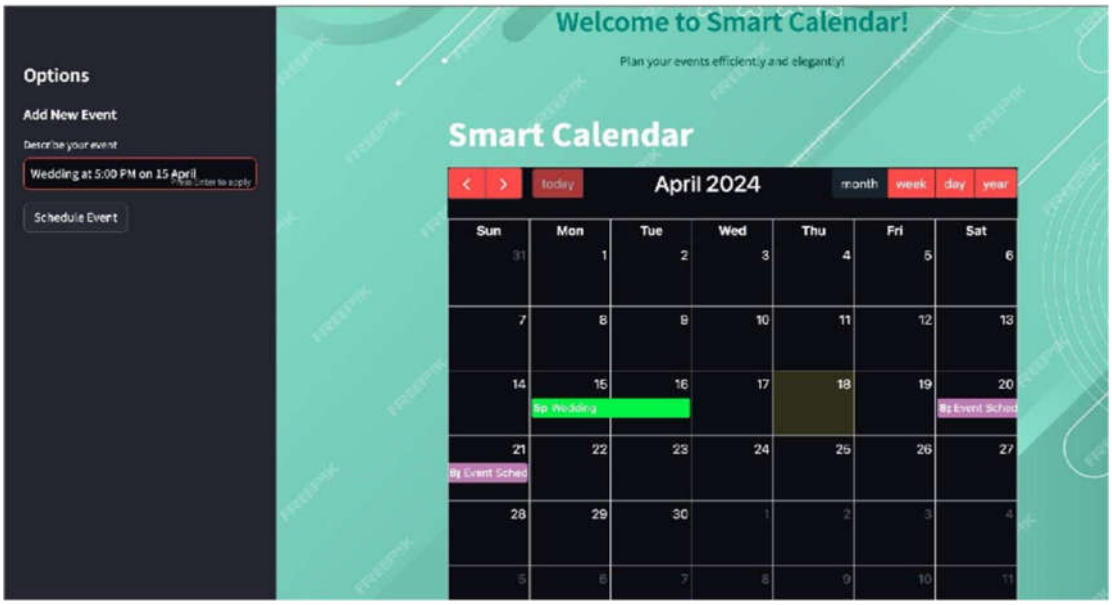
Workflow: BERT works through a structured process that begins with input preparation, where text is transformed into smaller units known as tokens. These tokens are then converted into numerical representations, enabling the model to process them effectively. The underlying architecture of BERT is the Transformer, which uses an attention mechanism to analyze relationships between words in a sentence, allowing the model to understand the context of a word from both directions in the sentence, making it bidirectional. Unlike traditional NLP models, BERT undergoes a two-step training process. Initially, it is pre-trained on a massive corpus of unlabeled text, allowing it to capture the general structure and nuances of language. Following pre-training, BERT is fine-tuned for specific tasks, such as sentiment analysis or named entity recognition, by adding a task-specific layer and training it with labeled data. This two-step process enables BERT to adapt and perform with high accuracy on a variety of natural language processing tasks.

The benefits of BERT are significant and multifaceted. Firstly, BERT excels in contextual understanding by analyzing words bidirectionally, allowing it to grasp the meaning of a word based on its surrounding context. This bidirectional approach gives BERT a considerable advantage over traditional models that process words sequentially. Secondly, versatility is a key strength of BERT, as it can be fine-tuned for a wide range of NLP tasks, making it adaptable for various applications such as sentiment analysis, named entity recognition, and more. Lastly, BERT has demonstrated state-of- the-art performance across numerous NLP benchmarks, consistently outperforming other models and establishing itself as a reliable choice for complex natural language understanding tasks.

Figure 1 illustrates a block diagram of a multi-head attention system, a crucial component of the Transformer architecture used in BERT. This neural network architecture allows the model to focus on different parts of the input data simultaneously through a mechanism known as attention. The diagram includes several layers, each serving a specific purpose. The Encoder Input Embedding layer converts the input text into numerical representations, which are then processed by the Positional Encoding layer to add information about the order of words in the sentence. The Multi-Head Attention mechanism enables the model to attend to different parts of the input sentence in parallel, with each "head" focusing on different aspects. After attention, the Add and Norm layers combine and normalize the outputs, followed by a Feed Forward layer, which further processes the results. The final Output Embedding converts the processed data back into a numerical form suitable for various tasks, and Softmax calculates the probabilities for possible outcomes, such as predicting the next word in a sentence or determining the sentiment of a text.

4. Results and Analysis

User Interface: The NLP-Enhanced Smart Calendar boasts a user-friendly interface designed to streamline event planning. As depicted in the image, users can effortlessly create new events by simply describing them in natural language. The calendar's sophisticated NLP capabilities automatically extract and populate the date and time fields, eliminating the need for manual input. For instance, when the user types "Wedding at 5:00 PM on 15 April," the calendar promptly identifies the key details and fills in the corresponding fields. With a single click of the "Schedule Event" button, the newly defined event is seamlessly integrated into the user's calendar. This intuitive approach significantly enhances the efficiency and convenience of event management, allowing users to effortlessly organize their schedules.



Options

Add New Event

Describe your event

Wedding at 5:00 PM on 15 April

Welcome to Smart Calendar!

Plan your events efficiently and elegantly!

Smart Calendar

April 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15 5p Wedding	16	17	18	19	20 8p Event Sched
21 8p Event Sched	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11

Figure 8. User Interface.

Text Classification:

A. Entity Recognition:

Entity recognition involves identifying and classifying named entities within text. In the context of the NLP-Enhanced Smart Calendar, this process is utilized to extract crucial information from natural language event descriptions, such as individuals, locations, and dates. The accompanying image illustrates the output of the entity recognition module for the sentence "I have to attend Tom's wedding at hotel Tulip Imperial." As depicted, the model successfully recognizes "Tom" as a person (PER) and "Hotel Tulip Imperial" as a location (LOC). This information can then be used to automatically populate the corresponding fields in the calendar's user interface.

B. Date and Time extraction:

The NLP pipeline is a core component of the NLP-Enhanced Smart Calendar. It is responsible for processing natural language text and extracting relevant information for scheduling events. The image provided below shows the code for the NLP pipeline. As you can see, the pipeline takes a sentence as input and performs the following tasks: Tokenization: Breaks the sentence down into individual words (tokens). Named Entity Recognition (NER): Identifies and classifies named entities in the sentence, such as dates and times. Intent Classification: Determines the user's intent from the sentence (e.g., schedule a meeting, cancel an event). The output of the pipeline is a list of dictionaries, where each dictionary contains information about the extracted entities and the user's intent. This information is then used to populate the calendar user interface and schedule events.

C. Accuracy of the Fine-tuned BERT Model:

The performance of the fine-tuned BERT model was evaluated over several epochs, and the results are illustrated in Figure 3. The graph depicts the accuracy achieved by the model as training progressed, demonstrating a consistent improvement in performance. Epochs vs. Accuracy: The x-axis represents the number of epochs, while the y-axis denotes the accuracy percentage. As shown, the model's accuracy increased from 72% at epoch 1 to an impressive 90% by epoch 5. This upward trend indicates that the model effectively learned from the training data, refining its understanding and improving its predictions with each subsequent epoch. Implications of Accuracy: The significant enhancement in accuracy suggests that fine-tuning the BERT model is effective for the specific tasks associated with the NLP-enhanced smart calendar. The increase in accuracy correlates with the model's ability to better understand and interpret user input, which is crucial for extracting relevant event information in a natural language format.This analysis highlights the efficacy of leveraging state-of-the-art models like BERT in natural language processing applications, particularly in improving user experience in scheduling tasks through enhanced accuracy in event recognition and context understanding.

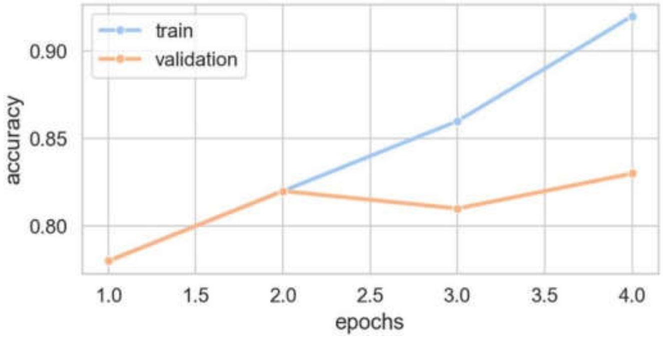


Figure 9. Accuracy of the Fine-tuned BERT Model Over Epochs. The graph illustrates the increase in accuracy from 72% to 90% over five epochs, indicating effective learning and adaptation of the model to the training data.

D. Performance Metrics Analysis:

To evaluate the performance of the fine-tuned BERT model, we employed density plots to visualize the distribution of key performance metrics: accuracy and F1 score. These metrics provide valuable insights into the model's effectiveness in correctly classifying instances and its ability to balance precision and recall.

Density Plot Interpretation**:** The density plot presented in Figure X illustrates the distribution of the accuracy and F1 score across multiple evaluations of the model. The x-axis represents the score values, while the y-axis indicates the density of occurrences within those score ranges.

1. Accuracy Distribution: The accuracy distribution, depicted in coral, demonstrates that the model achieved an average accuracy around 0.85, with a relatively narrow spread, indicating consistent performance across different validation sets. The peak of the density curve suggests that most of the evaluations yielded accuracy values clustered around this mean, signifying the model's reliability in predicting the correct outcomes.
2. F1 Score Distribution: The F1 score distribution, shown in royal blue, centers around an average of 0.80. The slight spread observed suggests variability in the model's ability to balance precision and recall. A prominent peak in the F1 score density reflects that a significant portion of the evaluations achieved competitive results, reinforcing the model's robustness in handling class imbalances.

Comparative Insights: The overlapping regions of the density curves indicate a positive correlation between accuracy and F1 score, suggesting that improvements in one metric tend to coincide with enhancements in the other. This observation is crucial for understanding the model's overall performance and its suitability for practical applications. Overall, the density plot provides a comprehensive view of the model's performance metrics, allowing for an effective assessment of its predictive capabilities. Future work will focus on refining the model further to improve these metrics and evaluate its performance on unseen data.

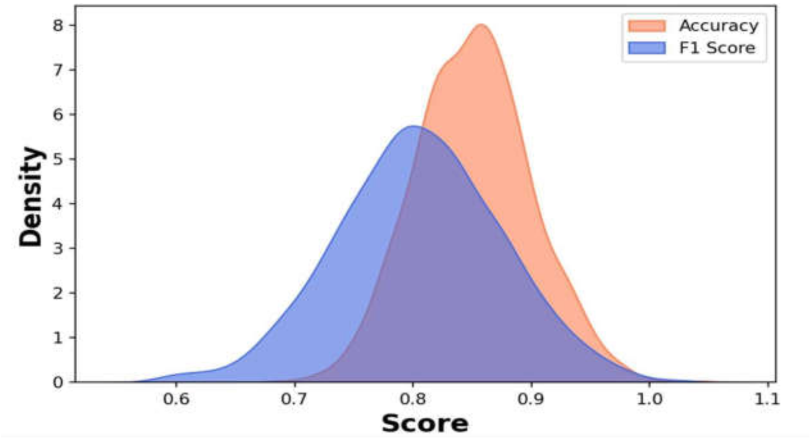


Figure 10. Density Plot of Model Performance Mertrics.

5. Future Scope

The proposed approach for enhancing the calendar functionality paves the way for several promising areas of development and research:

Multilingual Support: Expanding the calendar's language capabilities to support multiple languages will allow it to cater to a diverse global audience. This could involve translating the user interface, event details, and notifications into different languages. Implementing natural language processing (NLP) techniques can enable users to input events and retrieve information in their preferred language, providing a seamless user experience across linguistic boundaries.

Voice Commands: Integrating voice recognition technology will enable users to interact with the calendar using voice commands, offering a hands-free, efficient way to manage their schedules. Users will be able to create, edit, and delete events, set reminders, and receive updates through spoken commands, improving accessibility and convenience, particularly for those with mobility impairments. Integration with Other Apps: Connecting the calendar with email, productivity tools, and social networks will allow for comprehensive event management across various platforms. By synchronizing events with other applications, users can receive notifications, share events, and coordinate schedules seamlessly, enhancing overall productivity.

Customized Experience: Leveraging machine learning algorithms, the calendar can learn from users' scheduling patterns and preferences to offer personalized event suggestions and reminders. As the calendar adapts to individual habits, it can provide tailored recommendations for optimal scheduling, helping users manage their time more effectively.

Compatibility with Wearable Devices: Extending the calendar's accessibility to smartwatches and other wearable devices will provide users with real-time updates and notifications on their wrists. Users can quickly glance at their schedules, set reminders, and receive alerts on the go, enhancing convenience and connectivity.

6. Conclusion

The integration of Natural Language Processing (NLP) into smart calendar systems represents a significant advancement in personal productivity and efficiency. Throughout this thesis, we have explored the development and implementation of an NLP-enhanced smart calendar, aiming to streamline the process of event scheduling and management. By harnessing the power of NLP algorithms, our smart calendar has demonstrated the capability to understand and interpret natural language inputs, thereby enabling users to interact with their calendars in a more intuitive and seamless manner.

One of the key findings of this research is the effectiveness of NLP in improving user experience and reducing cognitive load associated with calendar usage. By allowing users to input event details in natural language, our smart calendar eliminates the need for manual data entry and simplifies the scheduling process. Moreover, the integration of NLP facilitates more intelligent features such as context-aware event suggestions and automatic scheduling, enhancing the overall functionality of the calendar system.

Furthermore, the success of our NLP-enhanced smart calendar opens up numerous avenues for future research and development. One promising direction is the refinement of NLP algorithms to achieve even greater accuracy and understanding of natural language inputs. Additionally, exploring the integration of advanced machine learning techniques could enable the smart calendar to adapt and personalize its recommendations based on user preferences and historical data.

Another exciting area for future exploration is the incorporation of multimodal inputs, allowing users to interact with the calendar using not only text but also speech and gestures. This would further enhance the accessibility and usability of the smart calendar, catering to a wider range of users with varying preferences and needs. Additionally, exploring the potential integration of NLP with emerging technologies such as augmented reality (AR) and virtual assistants could redefine the way we interact with and perceive calendar systems in the future.

In conclusion, the development of an NLP-enhanced smart calendar represents a significant step forward in harnessing the power of natural language understanding to enhance productivity and efficiency. Through this thesis, we have demonstrated the feasibility and effectiveness of such a system while also identifying several avenues for future research and development. By continuing to innovate and refine NLP technologies, we can unlock new possibilities for smart calendar systems, ultimately empowering users to better manage their time and resources in an increasingly complex world.

Author Statements:

- Ethical approval: The conducted research is not related to either human or animal use.
- Conflict of interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- Acknowledgement: The authors declare that they have nobody or no-company to acknowledge.
- Author contributions: The authors declare that they have equal right on this paper.
- Funding information: The authors declare that there is no funding to be acknowledged.
- Data availability statement: The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Donghyeon Kim, Jinhyuk Lee, Donghee Choi, Jaehoon Choi, Jaewoo Kang, "Learning User Preferences and Understanding Calendar Contexts for Event Scheduling", arXiv:1809.01316v3 [cs.LG] 18 Jul 2020.

[2] Ashish Vaswan, Noam Shazeer, Jakob Uszkoreit, Niki Parmar, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, "Attention Is All You Need," arXiv:1706.03762v7 [cs.CL] 2 Aug 2023.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv:1810.04805v2 [cs.CL] 24 May 2019.

[4] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, Chris Dyer, Carnegie Mellon University, "Neural Architectures for Named Entity Recognition," arXiv:1603.01360v3 [cs.CL] 7 Apr 2016.

[5] Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc Le. 2018. Semi- supervised sequence modelling with cross-view training. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1914-1925.

[6] Jeremy Howard and Sebastian Ruder 2018. Universal language model fine-tuning for text classification. In ACL. Association for Computational Linguistics.

[7] Erik F Tjong Kim Sang and Fien De Meulder 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In CoNLL.

[8] Zhongqiang Huang and Mary Harper. Self-training PCFG grammars with latent annotations across languages. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 832–841. ACL, August 2009.