modelling

MDPI

Article

Intent Identification by Semantically Analyzing the Search Query

Tangina Sultana [1,2], Ashis Kumar Mandal [3,4], Hasi Saha [3], Md. Nahid Sultan [3] and Md. Delowar Hossain [2,3,*]

[1] Department of Electronics and Communication Engineering, Hajee Mohammad Danesh Science and Technology University, Dinajpur 5200, Bangladesh; tangina@hstu.ac.bd
[2] Department of Computer Science and Engineering, Global Campus, Kyung Hee University, Yongin-si 1732, Republic of Korea
[3] Department of Computer Science and Engineering, Hajee Mohammad Danesh Science and Technology University, Dinajpur 5200, Bangladesh; pfy673@usask.ca (A.K.M.); hasi@hstu.ac.bd (H.S.); nahid.sultan@hstu.ac.bd (M.N.S.)
[4] Department of Computer Science, University of Saskatchewan, Saskatoon, SK S7N 5A2, Canada
* Correspondence: delowar@khu.ac.kr; Tel.: +82-10-3718-8309

Abstract: Understanding and analyzing the search intent of a user semantically based on their input query has emerged as an intriguing challenge in recent years. It suffers from small-scale human-labeled training data that produce a very poor hypothesis of rare words. The majority of data portals employ keyword-driven search functionality to explore content within their repositories. However, the keyword-based search cannot identify the users' search intent accurately. Integrating a query-understandable framework into keyword search engines has the potential to enhance their performance, bridging the gap in interpreting the user's search intent more effectively. In this study, we have proposed a novel approach that focuses on spatial and temporal information, phrase detection, and semantic similarity recognition to detect the user's intent from the search query. We have used the n-gram probabilistic language model for phrase detection. Furthermore, we propose a probability-aware gated mechanism for RoBERTa (Robustly Optimized Bidirectional Encoder Representations from Transformers Approach) embeddings to semantically detect the user's intent. We analyze and compare the performance of the proposed scheme with the existing state-of-the-art schemes. Furthermore, a detailed case study has been conducted to validate the model's proficiency in semantic analysis, emphasizing its adaptability and potential for real-world applications where nuanced intent understanding is crucial. The experimental result demonstrates that our proposed system can significantly improve the accuracy for detecting the users' search intent as well as the quality of classification during search.

Keywords: BERT; keyword search; n-gram model; phrase detection; semantic similarity recognition

Citation: Sultana, T.; Mandal, A.K.; Saha, H.; Sultan, M.N.; Hossain, M.D. Intent Identification by Semantically Analyzing the Search Query. *Modelling* **2024**, *5*, 292–314. https://doi.org/10.3390/modelling5010016

Academic Editor: Alfredo Cuzzocrea

Received: 4 December 2023
Revised: 15 January 2024
Accepted: 19 February 2024
Published: 22 February 2024

1. Introduction

In the recent past, more people have engaged with the web to access diverse information due to the explosive growth of the World Wide Web. The web is considered an information hotspot created by numerous authors having various vocabularies. Therefore, search engine technology is an acute necessity for exploiting these extremely valuable resources to help users. Web search engines can find the resources on people's demand by identifying the searchers intent behind the query. Most of the existing search engines rely on keyword matching to understand the users' intent. Therefore, in many cases, the retrieved documents are not relevant to what the users need. As a result, it needs to understand the users' queries deeply. Understanding the query and identifying the intent is a crucial step in displaying concise search results to the user's query. This will help to display improved ranking as well as semantically enriched search results [1,2]. For example, the query "South

Korea's president" and "Pittsburgh pics" would return "Moon Jae-in" and images in the search engine first.

Therefore, understanding the query and detecting the intent is a challenging task because the queries are very short. Moreover, detecting the exact indent from the users' search queries needs more context beyond the keyword. Over the past decades, much research has been conducted for query understanding. Early works need a lot of human analysis and effort to detect the intent of a search query [3]. Later, automated intent analysis from the query, such as query clustering and query classification, is introduced to understand the user's necessary information. Query classification classifies the queries into some predefined target categories based on various types of taxonomies. However, most of the existing research on query classification emphasizes the coarse-grained understanding of the query to categorize the intent. Therefore, detailed information is lost by using these schemes [2,4,5]. On the other hand, mining clusters from the queries is the strategy for identifying the search intent of the query clustering scheme. Nevertheless, it is difficult to identify and understand the contents of the cluster by the human [6–8]. Soto et al. [9] introduces Thalia, a semantic search engine updated daily from PubMed, capable of recognizing eight types of concepts in biomedical abstracts, providing a valuable tool for precision medicine research. Kostakos et al. [10] proposes emerging content delivery methods like quote and entity searching, facilitating rapid identification of relevant information in unstructured texts. The prototype search engine utilizes these methods, drawing from the GDELT Global Quotation Graph, with applications in web surveillance, crime informatics, and enabling non-technical users to assess public discourse quality. However, Ayazbayev et al. [11] focuses on enhancing information retrieval by determining semantically close words, particularly in languages lacking established linguistic tools. The study employs distributed methods on Apache Spark, utilizing vector representations and pre-trained multilingual sentence Transformers to efficiently calculate semantic similarity and enable effective searches in languages like Kazakh. Moreover, recent strides in natural language processing (NLP) have seen the emergence of transformative models, for example, Bidirectional Encoder Representations from Transformers (BERT), which have demonstrated exceptional capabilities in contextualized language understanding. Some researchers are focused on named entity recognition (NER) for search engines such as Bouarroudj et al. [12] addresses the challenge of named entity disambiguation (NED) in knowledge graphs (KGs) specifically for short text fragments, a scenario often overlooked in current research focusing on long texts. The proposed NED approach incorporates context expansion, coherence analysis in queries with multiple entities, consideration of word relations, and syntactic features. On the contrary, Cowan et al. [13] focuses on named entity recognition (NER) in travel-related search queries, addressing the challenges posed by minimal context and few structural clues. The proposed machine learning-based solution, employing a conditional random field (CRF) sequence model, achieves high accuracy with an F1-score of 86.4% on a held-out test set. The developed NER classifier is actively utilized in a real-life travel search engine, demonstrating its practical applicability.

In this study, we have proposed a method that represents a pioneering endeavor to unravel and harness the intricate fabric of user intent through a sophisticated system model. Our proposed framework comprises three major steps: spatial and temporal parsing; phrase detection; and semantic similarity recognition. By using these three components, the search engine captures the users' intent by identifying the spatial and temporal range of the query, seeking concepts based on phrases rather than individual keywords, and narrowing the search scope to having semantically similar intent being recognized from the query. In this paper, we introduce a novel approach that utilizes RoBERTa, a variant of BERT, and augments it with a probability-aware gated mechanism to refine the representation of user queries for semantic similarity recognition to analyze the intent. To the best of our knowledge, this is the first attempt to integrate RoBERTa with a probability-aware gated mechanism to enhance the interpretability and performance of intent classification systems.

The main contributions of the proposed system model are given below:

- We introduce spatial and temporal parsing, phrase detection, and semantic similarity recognition for semantic analysis and recognition to identify the intent of the user's search query.
- We propose a probability-aware gated mechanism with a pre-trained RoBERTa model, which enhances the proposed system's ability to discern nuanced intents through effective attention mechanisms.
- We incorporate adaptive training with Gensim to support continuous learning and refinement and ensure adaptability to evolving language patterns over time.
- Extensive experimental analyses on benchmark datasets demonstrate the superior performance of our proposed system compared to state-of-the-art systems.

The rest of this paper is arranged as follows. Section 2 describes the system model of our proposed system and includes a detailed explanation of spatial and temporal parsing, phrase detection, and semantic similarity recognition. In the semantic similarity section, we have introduced a probability-aware gated mechanism with a pre-trained RoBERTa model for semantically detecting intent. On the other hand, the performance analysis along with detailed necessary discussions are narrated in Section 3. Finally, Section 4 concludes the paper.

2. Materials and Methods

To semantically analyze the user query and identify the intent of the user, we have proposed a query understanding scheme. This scheme consists of three components: spatial and temporal parsing; phrase detection; and semantic similarity recognition (Figure 1).

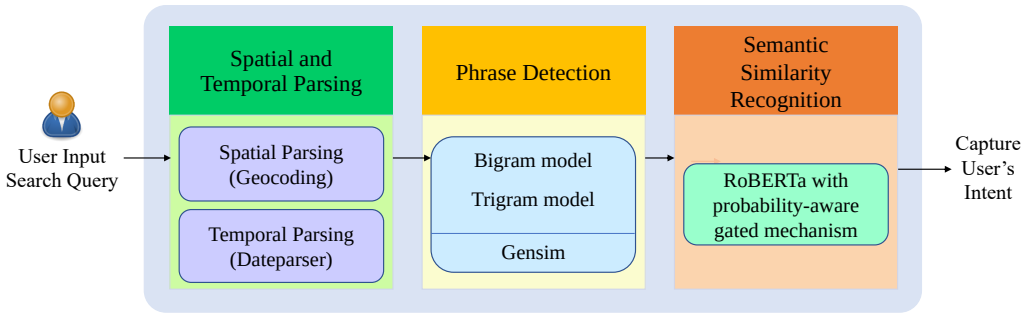

Figure 1. Proposed system model for semantically analyzing the users' search query.

The spatial bounding box extracts the latitude and longitude of the address from the query. The temporal parsing identifies the date and time range from the given query. On the other hand, phrase detection isolates the phrases, and semantic similarity recognition detects the intents of a given query. For example, from a given query "Transport System South Korea 2019–2020", "South Korea" and "2019–2020" are considered as spatial bounding boxes and temporal ranges. On the other hand, "Transport System" is identified as a domain phrase learned from the metadata. The semantic similarity recognition module augments the extracted phrases and spatial and temporal parsing information to classify the user intent (Figure 2). After detecting spatial and temporal features, domain phrases, and semantic similarity, the user's intent is identified, and the top-k associated results are retrieved according to the user's intent in the search engine.

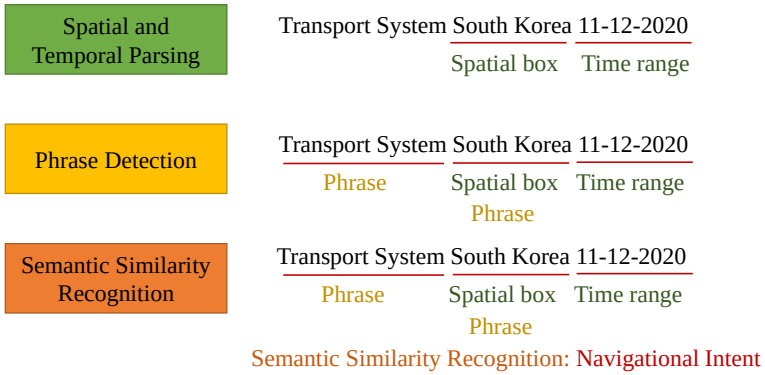

Figure 2. Conceptual framework of our proposed system model.

2.1. Spatial and Temporal Parsing

To discover the data, spatial parsing and temporal range play an important role. In our system model, we have used two tools. For geospatial parsing, we use the Google Geocoding API client [14] which is an open-source geocoding package. If the unstructured query has an address, this API transforms the address into its corresponding latitude and longitude coordinates, defining the spatial bounding box for the given address. The average time complexity of geospatial parsing operations ranges from $O(1)$ to $O(n)$, where n is the size or complexity of the input.

On the other hand, query data (date) are not always structured and can be free-form natural language. The format of that kind of date does not match the temporal attributes of metadata. Therefore, in our proposed system, we use Dateparser [15], which is an open-source library written in Python for temporal parsing. It converts the specific date into a DateTime object according to the metadata format for extracting the date into the time range. The time complexity of temporal parsing using Dateparser ranges from linear $(O(n))$ to higher complexities, depending on the sophistication of the parser.

2.2. Phrase Detection

Data search engines do not consider the order of the user input query usually. It regards the query as a bag of words and fetches the results having at least one word that is present in the query, e.g., the input search query of a user is “rent Hyundai car”, the search engine may return the pages containing rent, Hyundai, and/or car. However, the “rent Hyundai car" items are the true target of the user’s attention. If the search engine can detect “rent Hyundai car” as a phrase, it will retrieve more relevant data from the query. Some research activities have been carried out in order to occupy the gap between the real intent of the user and the search query. Therefore, some researchers are trying to develop data search engines with customized configurations capable of searching the data by analyzing the users’ query and the real intention of that search query. If the search filters combine the functionality of the metadata indexing strategies, the search engines may be capable of finding phrases in more complicated searches [16,17]. Though these functionalities enhance the search potentiality of data portals, it is necessary to spend more time by the user for learning the syntax of composite searches and related knowledge.

Therefore, in our proposed system, we use two n-gram models named Bigram and Trigram that are trained by using metadata such as title, description, and so on. These models are used to detect and extract the phrases from the query automatically by matching them with the corpus for improved search results. In a given sample of text or speech, n-gram is a contiguous sequence of n items in the case of the computational linguistics field [18]. Bigram is a sequence of two adjacent possible words in a sentence generated from the neighboring words while Trigram is a sequence of three consecutive words [19].

Bigram can determine the conditional probability of a word given the preceding word if we apply the relation of the conditional probability:

$$P(X_n \mid X_{n-1}) = \frac{P(X_{n-1}, X_n)}{P(X_{n-1})} \tag{1}$$

Here, X_n represents the current word at position n, X_{n-1} represents the preceding word at position $n-1$, $P(X_n \mid X_{n-1})$ defines the conditional probability of the current word X_n given the preceding word X_{n-1}, $P(X_{n-1}, X_n)$ denotes the joint probability of both the preceding word, X_{n-1} and the current word X_n and $P(X_{n-1})$ reflects the probability of the preceding word X_{n-1}. This means that the probability P() of a word X_n given the preceding word X_{n-1} is equivalent to the probability of the Bigram. Therefore, if we use the Bigram model, we make the below approximation for predicting the conditional probability of the next word.

$$P\left(X_n \mid X_1^{n-1}\right) \approx P(X_n \mid X_{n-1}) \tag{2}$$

Here, X_1^{n-1} represents the sequence of words from position 1 to $n-1$, and $P(X_n \mid X_1^{n-1})$ denotes the conditional probability of the current word X_n given the sequence of words X_1^{n-1}. The formula approximates this conditional probability using $P(X_n \mid X_{n-1})$ defines the conditional probability of X_n given the preceding word X_{n-1}.

We can generalize the Bigram to Trigram (finds two words from the past) and to n-gram (finds $n-1$ words from the past) by following the below general equation:

$$P\left(X_n \mid X_1^{n-1}\right) \approx P\left(X_n \mid X_{n-N+1}^{n-1}\right) \tag{3}$$

Here, N represents the size of the N-gram model, X_{n-N+1}^{n-1} represents the sequence of $N-1$ preceding words from position $n-N+1$ to $n-1$. The formula generalizes the conditional probability using $P(X_n \mid X_{n-N+1}^{n-1})$: conditional probability of X_n given the sequence of $N-1$ preceding words.

Let us consider the example query "rent Hyundai car". Using the Bigram model, the conditional probability of the next word, say "car" (X_n), given the preceding word "Hyundai" (X_{n-1}), can be approximated as follows:

$$P(\text{car} \mid \text{Hyundai}) \approx P(\text{car} \mid \text{Hyundai})$$

This approximation allows the system to understand the likelihood of the word "car" following the word "Hyundai" in the given context. Similarly, the Trigram model extends this concept, considering three consecutive words. For instance, the conditional probability of the word "car" given the sequence "rent Hyundai" (X_{n-2}, X_{n-1}) would be approximated as:

$$P(\text{car} \mid \text{rent Hyundai}) \approx P(\text{car} \mid \text{Hyundai})$$

These models help identify meaningful phrases in the query, enabling the system to recognize and prioritize relevant information, such as the user's intent to rent a Hyundai car.

In our proposed system, we use Gensim [20] for detecting the phrases to identify the concept and generate the Bigram and Trigram models. It is an open-source vector space as well as a topic modeling toolkit that is used to handle unstructured text. It can also train our Bigram and Trigram model from the metadata by using two parameters: minCount and phrase threshold. The parameter "minCount" is used to ignore all the words if the occurrence of a given word is lower than this. On the other hand, the parameter "phraseThreshold" determines the threshold to generate the phrases. If the threshold is high, we can detect a few phrases. For instance, a word phrase consisting of the words x and y is classified as a Bigram when the following condition is met:

$$\text{count}(x, y) \times M(\text{count}(x) \times \text{count}(y)) \leq \frac{\text{phraseThreshold}}{\text{minCount}} \tag{4}$$

Here, M represents the overall vocabulary size and count(x) shows how many times the word x appears in the corpus. The definition of the function count(x, y) in Equation (4) when it has two arguments is provided in the following:

$$\text{count}(x,y) = \begin{cases} \text{count}(x,y) & \text{if the occurrence of the Bigram } (x,y) \text{ is higher than the specified} \\ \text{minCount} & \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Here, count(x, y) represents the number of occurrences of the bigram (sequence of two consecutive words) consisting of words x and y in the corpus. The function count(x, y) is conditioned on the requirement that the occurrence of the Bigram is higher than the specified minCount. If the count is higher than minCount, the actual count is used; otherwise, it is considered 0. This condition is denoted by count$(x, y) \geq$ minCount in the context of the equation.

Therefore, our proposed system uses Gensim to train the Bigram and Trigram models. As a result, the system can find out at most two or three phrases. When a user inserts his or her query in our proposed search engine, the two pre-trained Bigram and Trigram models will be employed sequentially to convert the input query of the user to its associated phrases apprehended by using the models. The derived phrases are then forwarded to the output of the user's query (Figure 3).

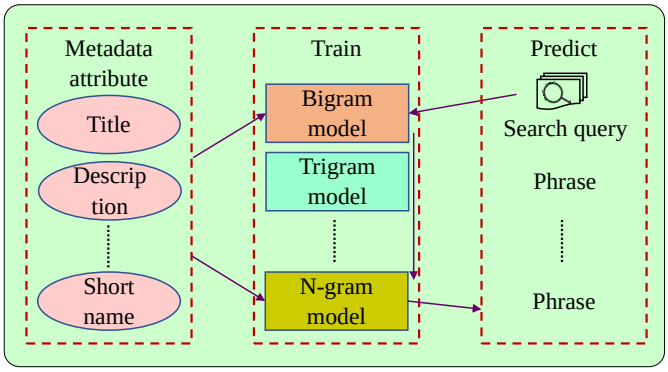

Figure 3. Phrase detection process.

The time complexity of training n-gram models with Gensim depends on various factors such as the size of the corpus, the number of iterations, and the chosen optimization techniques. Typically, the complexity is in the range of $O(K \times log(N))$, where K is the number of iterations and N is the size of the corpus. The time complexity for applying the trained models to a user query would depend on the length of the query and the size of the models. Assuming the length of the query is L, and the average length of phrases detected is M, the complexity could be approximated as $O(L \times M)$. On the other hand, the phrase detection process involves applying the pre-trained models to the user query, and the complexity would be influenced by the length of the query and the size of the models. Assuming the length of the query is L and the average length of phrases detected is M, the complexity could be approximated as $O(L \times M)$. Considering these aspects, the overall time complexity of the phrase detection process can be estimated as $O(L \times M) + O(K \times log(N))$.

2.3. Semantic Similarity Recognition

The semantic similarity recognition module is used to identify the intent of the search query. Predicting user intent from the search query is a challenging task because it needs multiple sources of information such as web surfing history, geo-location, or user profiles. In this study, we focus only on predicting the user intent based on queries that are used to access the web pages. On the other hand, user intent classification is always suffering from the lack of sufficient labeled datasets that are often annotated manually. A large amount

of research has been conducted to detect the user intent from the search queries. But most of these approaches [21–23] leverage static embedding or depiction of bag-of-words from shallow neural networks. As a result, they suffer from the dynamic description of words in a sentence. However, dynamic description of words is crucial because it helps to understand the users' intent.

In recent years, word embedding [24] has gained huge popularity and become the industry standard for representing word intent. On the other hand, FastText [25], Word2Vec [26], and GloVe [27] are examples of static methods that are used to produce fixed depictions in a vocabulary that cannot be easily adapted for the contextual meaning of words. Deep learning techniques can also influence the contextual meaning of words [24,28,29] and improve the learning capabilities. However, the performance of these schemes depends on the annotation of the huge volume of training data. Recently, some dynamic pre-trained representations such as deep contextual word, ELMo, and BERT (Bidirectional Encoder Representations from Transformers) [30] can generate dynamic representations of words based on the context. These models have achieved excellent performance. Among them, the semantic information learned by BERT is more accurate. Therefore, by comparing the research relevant to text classification, we can conclude that very few classifiers for intent analysis have been addressed in recent years.

In our proposed system, we have proposed a RoBERTa model with a probability-aware gated mechanism to pre-process as well as fine-tune our dataset for identifying the user intent. However, to the best of our knowledge, this is the first attempt to semantically analyze the user intent from the users' search query by using a RoBERTa model with a probability-aware gated mechanism and limited available labeled data.

2.3.1. RoBERTa

BERT, or Bidirectional Encoder Representations from Transformers, is a powerful Transformer-based model designed for various natural language processing tasks, including semantic similarity recognition (Figure 4). The core idea behind BERT is bidirectional training, allowing it to capture contextual relationships between words in both directions. For semantic similarity recognition, BERT provides contextualized embeddings for input sequences. In BERT, for the classification of task, a special token named $[CLS]$ is inserted in the first and [SEP] is added as a final token. For example, the output of BERT is $H = (h_1, \ldots, h_T)$ of an input token sequence, $y = (y_1, \ldots, y_T)$. It is trained bi-directionally on a huge corpus having unlabeled text that includes the entire Wikipedia and Book Corpus. After training, this model can easily identify the meaning of a language more correctly. Therefore, this model can detect the intent more efficiently.

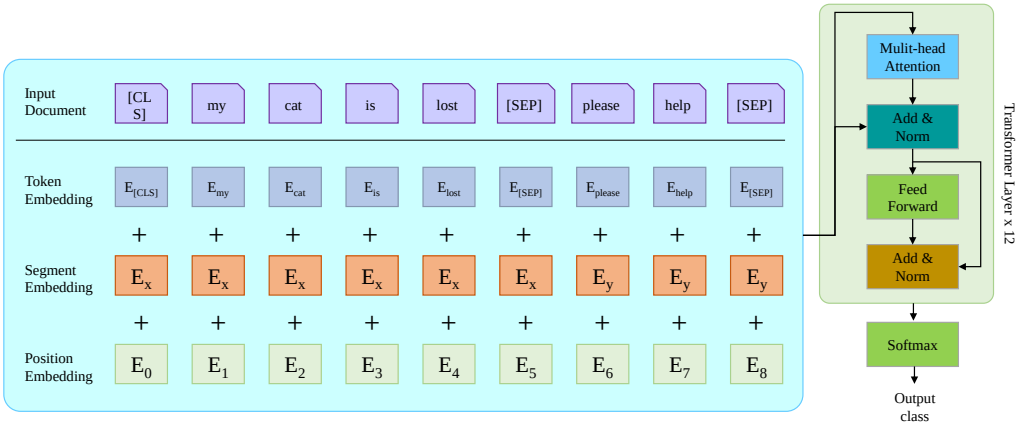

Figure 4. The semantic architecture of BERT.

RoBERTa [31], a variant of BERT, is designed to address some limitations and enhance performance in natural language understanding tasks. RoBERTa removes the next sentence prediction objective and trains on more extensive data, resulting in improved

representations. Similar to BERT, RoBERTa is applied to semantic similarity recognition tasks (Figure 5).

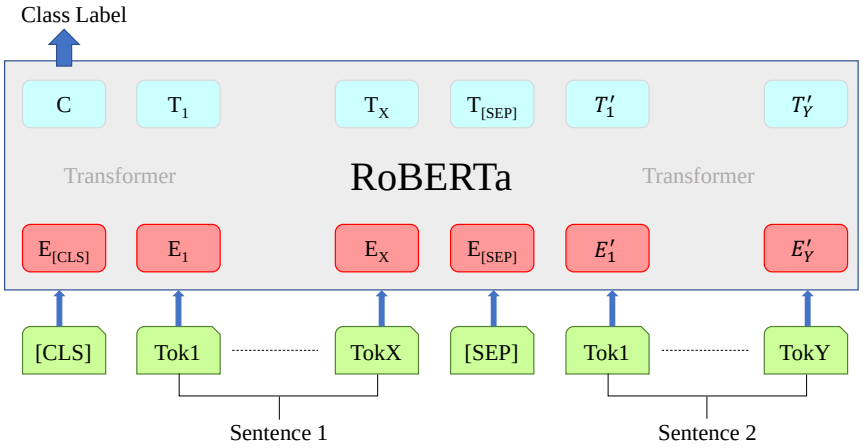

Figure 5. The semantic architecture of RoBERTa.

RoBERTa generates token embeddings $\text{RoBERTa}_\theta(Q)_i$ for each token q_i in the input sequence:

$$\text{RoBERTa}_\theta(Q)_i = \text{RoBERTa}_\theta(q_i)$$

Pooling techniques, such as mean pooling or max pooling, are also used with RoBERTa to obtain a fixed-size representation for the entire sequence:

$$\text{RoBERTa}_{\text{pool}}(Q) = \text{Pooling}(\text{RoBERTa}_\theta(Q))$$

Several studies have demonstrated the effectiveness of BERT and RoBERTa in semantic similarity recognition. Devlin et al. [30] introduced BERT, showcasing its state-of-the-art performance across various NLP tasks. Liu et al. [31] presented RoBERTa as an improvement over BERT, achieving better results on several benchmarks. Semantic similarity recognition tasks, including paraphrase identification and sentence similarity, have benefited from the contextualized embeddings provided by both BERT and RoBERTa. Briskilal et al. [32] demonstrated the application of RoBERTa in various downstream tasks, emphasizing its robustness and versatility. These Transformer-based models have become foundational in NLP research, setting new benchmarks and pushing the boundaries of semantic understanding in text.

2.3.2. Probability-Aware Gated Mechanism for RoBERTa Embeddings Refinement

We introduce a probability-aware gated mechanism to enhance RoBERTa embeddings by assigning significance weights to individual tokens in a given query. This mechanism allows for more fine-grained control over the contribution of each token to the final representation. This way, we can have more control over how much each token contributes to the final representation, enabling the model to capture subtle nuances in user queries. The concept of gating mechanisms in neural networks has been widely explored in the literature. Cho et al. [33] introduced the Gated Recurrent Unit (GRU), a gating mechanism for recurrent neural networks. The use of gating mechanisms in combination with attention mechanisms has shown effectiveness in various natural language processing tasks [34].

The application of gating mechanisms in the context of refining embeddings, especially in combination with Transformer-based models like BERT and RoBERTa, has gained attention for tasks requiring nuanced control over token importance. The proposed gated embeddings for intent classification integrate gating mechanisms with Transformer-based embeddings, providing a nuanced approach to capture contextual importance in user queries.

Gating Function

The gating function $G_\phi(Q)$ computes gating probabilities for each token in the query. It is like a gatekeeper that decides how much attention each token in the query should receive. Moreover, it determines the relevance or importance of each word in understanding the user's intent. We employ a sigmoid activation function to normalize values between 0 and 1:

$$g_i = \sigma(W_g \cdot \text{RoBERTa}_\theta(Q)_i + b_g)$$

Here, g_i is the gating probability for the i-th token, W_g is the weight matrix, σ is the sigmoid activation function, and b_g is the bias term. The mechanism considers the RoBERTa embeddings of the token, multiplies it by a learnable weight matrix W_g, applies a sigmoid function σ, adds a bias term b_g, and outputs a value between 0 and 1 representing the gating probability for that token.

Weight matrix W_g is a learnable parameter in the model. During training, the values of the weight matrix are adjusted through backpropagation to minimize the loss function. Additionally, the bias term b_g is similar to the weight matrix; the bias term is also a learnable parameter that is optimized during the training process. However, $\text{RoBERTa}_\theta(Q)$ represents a vector of embeddings. In the context of the gating function, this vector contains the embeddings for each token in the input sequence Q. The sigmoid activation function σ takes the weighted sum of the RoBERTa embeddings for a specific token, adds the bias term, and applies the sigmoid function. This process results in a scalar output for each token, representing its gating probability. The training process aims to optimize the model parameters, including the weights and biases, to minimize a specific loss function. The gating probabilities g_i are part of the model's output during the training phase. The optimization involves adjusting the parameters (W_g and b_g) so that the computed g_i values align with the true labels or targets, ultimately minimizing the loss.

Gated Semantic Embeddings

The gated semantic embeddings, $E_{\text{gated}}(Q)$, are obtained by element-wise multiplication of RoBERTa embeddings and gating probabilities. In other words, each token's embedding is scaled by its corresponding gating probability. The formula for computing the i-th component of the gated semantic embeddings is expressed as:

$$E_{\text{gated}}(Q)_i = g_i \cdot \text{RoBERTa}_\theta(Q)_i$$

These gated semantic embeddings provide a refined representation of the input sequence, emphasizing tokens deemed important by the gating mechanism. The following section also provides a derivation of how the gating function is updated during training to align with the true labels. This involves calculating gradients and updating the learnable parameters through a process called backpropagation.

Let us derive the update rule for g_i using gradient descent. The loss function for the semantic similarity task is denoted by $\mathcal{L}$, and the update rule is obtained through backpropagation.

$$\frac{\partial \mathcal{L}}{\partial g_i} = \frac{\partial \mathcal{L}}{\partial E_{\text{gated}}(Q)_i} \cdot \frac{\partial E_{\text{gated}}(Q)_i}{\partial g_i}$$

The gradient with respect to $E_{\text{gated}}(Q)_i$ can be computed using the chain rule:

$$\frac{\partial E_{\text{gated}}(Q)_i}{\partial g_i} = \text{RoBERTa}_\theta(Q)_i$$

Therefore,

$$\frac{\partial \mathcal{L}}{\partial g_i} = \frac{\partial \mathcal{L}}{\partial E_{\text{gated}}(Q)_i} \cdot \text{RoBERTa}_\theta(Q)_i$$

We update g_i using gradient descent:

$$g_i \leftarrow g_i - \alpha \cdot \frac{\partial \mathcal{L}}{\partial g_i}$$

Here, α is the learning rate.

Intent Classification with Gated Embeddings

The refined embeddings are then used for intent classification. Intent classification involves predicting the user's intent behind a given query. We propose a gated embedding mechanism to refine the RoBERTa embeddings for improved intent classification. The intent classification layer produces logits Z for each intent class and the softmax function is applied to obtain probabilities:

$$Z = W_z \cdot E_{\text{gated}}(Q) + b_z$$

Here, W_z is the weight matrix, and b_z is the bias term.

The logits Z are used to compute probabilities through a softmax function for final intent prediction.

$$P(\text{Intent} = c|Q) = \frac{e^{Z_c}}{\sum_{k=1}^{K} e^{Z_k}}$$

where K is the total number of intent classes.

The gating probabilities g_i are computed using the sigmoid activation function:

$$g_i = \sigma(W_g \cdot \text{RoBERTa}_\theta(Q)_i + b_g)$$

The binary cross-entropy loss $\mathcal{L}_{\text{gate}}$ for the gating mechanism is defined as:

$$\mathcal{L}_{\text{gate}} = -\sum_{i=1}^{T} [y_i \cdot \log(g_i) + (1 - y_i) \cdot \log(1 - g_i)]$$

Here, y_i is the binary label indicating the importance of the i-th token.

The loss for intent classification incorporates both the cross-entropy loss for intent prediction $\mathcal{L}_{\text{intent}}$ and the gating loss $\mathcal{L}_{\text{gate}}$:

$$\mathcal{L} = \mathcal{L}_{\text{intent}} + \lambda \cdot \mathcal{L}_{\text{gate}}$$

Here, λ controls the influence of the gating loss in the overall objective.

The overall process involves dynamically adjusting the model parameters during training to learn the optimal weights for both intent classification and the gating mechanism. The model is trained to minimize a combined loss function, considering both the intent classification loss and the gating loss.

The gating mechanism involves the calculation of gating probabilities for each token. This operation depends on the size of the embedding and the parameters of the gating function. Let us denote the size of the embedding as E. The time complexity for the gating mechanism can be considered $O(E)$, as it involves matrix multiplications and activation functions for each token.

Training Objective

The training objective for our proposed system involves optimizing two primary components: intent classification and the gating mechanism. The overall loss function $\mathcal{L}$ is a combination of the intent classification loss $\mathcal{L}_{\text{intent}}$ and the gating loss $\mathcal{L}_{\text{gate}}$.

$$\mathcal{L} = \mathcal{L}_{\text{intent}} + \lambda \cdot \mathcal{L}_{\text{gate}}$$

Here, $\mathcal{L}_{\text{intent}}$ is the cross-entropy loss for intent classification, $\mathcal{L}_{\text{gate}}$ is the binary cross-entropy loss for the gating mechanism, and λ controls the trade-off between the two losses.

The intent classification loss $\mathcal{L}_{\text{intent}}$ is computed as follows:

$$\mathcal{L}_{\text{intent}} = -\sum_{c=1}^{K} y_c \cdot \log(P(\text{Intent} = c|Q))$$

where K is the total number of intent classes, y_c is the binary label for class c, and $P(\text{Intent} = c|Q)$ is the predicted probability for class c.

The gating loss $\mathcal{L}_{\text{gate}}$ is defined as:

$$\mathcal{L}_{\text{gate}} = -\sum_{i=1}^{T} [y_i \cdot \log(g_i) + (1 - y_i) \cdot \log(1 - g_i)]$$

Here, T is the length of the input sequence, y_i is the binary label indicating the importance of the i-th token, and g_i is the gating probability for the i-th token.

Model Training

The model is trained by minimizing the overall loss $\mathcal{L}$ with respect to the model parameters θ and ϕ. The optimization is typically performed using stochastic gradient descent (SGD) or a variant like Adam.

$$\theta, \phi \leftarrow \theta, \phi - \eta \cdot \nabla_{\theta,\phi} \mathcal{L}$$

Here, η is the learning rate and $\nabla_{\theta,\phi} \mathcal{L}$ is the gradient of the loss with respect to model parameters θ and ϕ.

The training process involves presenting labeled training samples (Q_i, y_i) and updating the model parameters iteratively until convergence.

During training, multiple iterations are performed over the dataset. The number of epochs (denoted as N) and the size of each batch (denoted as B) contribute to the overall time complexity. The time complexity of training can be expressed as $O(N \times B \times (T + E))$, considering the forward and backward passes through the network.

Inference

During inference, the trained model is used to predict the intent of a given user query. The forward pass involves computing the gated semantic embeddings $E_{\text{gated}}(Q)$ and using them to generate intent probabilities through the intent classification layer.

$$P(\text{Intent} = c|Q) = \frac{e^{Z_c}}{\sum_{k=1}^{K} e^{Z_k}}$$

where Z is the logits obtained from the intent classification layer.

The final intent prediction is determined by selecting the class with the highest probability:

$$\text{Predicted Intent} = \text{argmax}_c P(\text{Intent} = c|Q)$$

This comprehensive approach aims to enhance the interpretability and performance of intent classification systems. During inference, the time complexity depends on the size of the input sequence and the number of intent classes. Let us denote the number of intent classes as K. The time complexity for inference can be considered $O(T + E + K)$, where T is the sequence length, E is the embedding size, and K is the number of intent classes. The algorithm of the proposed RoBERT with probability-aware gated mechanism for intent identification is given in Algorithm 1.

Algorithm 1 RoBERTa with Probability-Aware Gated Mechanism for Intent Identification

Input: User query Q, RoBERTa model parameters θ, Gating mechanism parameters ϕ
Output: Predicted intent label

Load pre-trained RoBERTa model with parameters θ
Initialize gating mechanism parameters ϕ
Freeze parameters of the RoBERTa model during fine-tuning
procedure FINE-TUNEMODEL
  for each epoch **do**
    for each batch B in training data **do**
      Compute RoBERTa embeddings: $E_{\text{RoBERTa}}(Q) = \text{RoBERTa}_\theta(Q)$
      Compute gating probabilities: $G(Q) = \sigma(\text{Gating}_\phi(E_{\text{RoBERTa}}(Q)))$
      Compute gated embeddings: $E_{\text{gated}}(Q) = G(Q) \odot E_{\text{RoBERTa}}(Q)$
      Compute intent logits: $Z = W_z \cdot E_{\text{gated}}(Q) + b_z$
      Compute intent probabilities: $P(\text{Intent} = c|Q) = \text{softmax}(Z)$
      Compute intent classification loss: $\mathcal{L}_{\text{intent}} = -\sum_c y_c \cdot \log(P(\text{Intent} = c|Q))$
      Compute gating loss: $\mathcal{L}_{\text{gate}} = -\sum_i [y_i \cdot \log(G(Q)_i) + (1 - y_i) \cdot \log(1 - G(Q)_i)]$
      Compute overall loss: $\mathcal{L} = \mathcal{L}_{\text{intent}} + \lambda \cdot \mathcal{L}_{\text{gate}}$
      Update parameters θ, ϕ using backpropagation and SGD
    end for
  end for
end procedure

3. Results and Discussion

3.1. Dataset

To evaluate the performance of our proposed system, we perform experiments on two standardized datasets. The first dataset, ATIS [35], comprises audio recordings of travelers making flight reservations. The second dataset, SNIPS [36], consists of records from personal voice assistants. Notably, the SNIPS dataset is more extensive and exhibits greater diversity compared to the ATIS dataset. The statistics of the dataset are shown in Table 1. In Table 1, an intent represents the underlying purpose or goal expressed in a user's input. It reflects what the user wants to accomplish with a particular interaction or query. For example, in a weather chatbot, a user's input like "What is the weather in New York tomorrow?" might have the intent "GetWeather". On the other hand, a slot, also known as an entity or a parameter, is a specific piece of information or variable within a user's input that is relevant to the intent. It helps in extracting specific details from the user's utterance. For example, in the same weather chatbot example, the slots could include "Location" (New York) and "Time" (tomorrow), which are essential for fulfilling the "GetWeather" intent. In Table 1, Train-Sentences, Dev-Sentences, Test-Sentences represent the number of sentences in the training, development (validation), and testing sets, respectively. Additionally, the size of the vocabulary in each dataset indicates the total number of unique words present.

Table 1. Statistics of the dataset.

Datasets	ATIS	SNIPS
Train-Sentences	4778	13,084
Dev-Sentences	500	700
Test-Sentences	893	700
Intent	21	7
Slot	126	72
Vocabulary	722	11,241

3.2. Baselines

We conduct a comparative analysis of our proposed system against several baseline models, including:

- CAPSULE-NLU: The CAPSULE-NLU model, which was suggested by Zhang et al. [37], makes use of a neural network that is built on capsules and a method that uses dynamic routing-by-agreement to identify hierarchical links between words, intent, and slots.
- SF-ID Network: The SF-ID network, which was first proposed by E et al. [38], is intended to represent two-way relationships between intent detection and slot filling. It has two modes, ID-First and SF-First, with different starting orders.
- Stack-Propagation: Qin et al. [39] use Stack-Propagation to tackle intent detection problems by classifying intent at the token level. Intent information can be used to guide slot-filling operations in the Stack-Propagation architecture.
- Graph LSTM: Graph LSTM is introduced by Zhang et al. [40] to improve upon Slot Labeling Units (SLU) and circumvent the drawbacks of recurrent neural networks (RNNs).
- BERT-Joint: The BERT-Joint method was developed by Chen et al. [41] with the aim of improving performance in the joint tasks of intent detection and slot filling by utilizing BERT's contextual awareness.
- Joint Sequence: Chen et al. [42] has proposed a novel model for multi-intent NLU called SelfDistillation Joint NLU (SDJN).
- Capsule Neural Network: Abro et al. [43] introduces the WFST-BERT model, which integrates weighted finite-state transducer (WFST) into the fine-tuning of a BERT-like architecture to mitigate the requirement for large quantities of supervised data.
- GE-BERT: Li et al. [44] addresses the challenge of query understanding by proposing GE-BERT, a novel graph-enhanced pre-training framework that leverages both query content and the query graph to capture semantic information and users' search behavioral information, demonstrating its effectiveness through extensive experiments on offline and online tasks.
- Joint BiLSTM-CRF: Rizou et al. [45] focuses on developing an efficient multilingual conversational agent for university students, supporting both Greek and English, using a joint BiLSTM-CRF model for intent extraction and named entity recognition, achieving competitive performance in customer service tasks and introducing the UniWay dataset, demonstrating the effectiveness of a unified approach in handling multiple natural language understanding tasks in closed domains.

3.3. Experimental Setup

We run all of our experiments on a server equipped with an Nvidia GeForce RTX 3090Ti (24 GB) GPU card. For training the semantic similarity recognition module, we have used the RoBERTa pre-trained BERT model for detecting the intent from the users' search query. This model is equipped with 12 layers of Transformers, each featuring 768 hidden units, 12 attention heads, and a 0.1 dropout probability. We set the maximum sequence length of 60, the batch size of 32, and train the proposed RoBERTa with a probability-aware gated mechanism for intent identification with Adam optimizer having an initial learning rate of $5 \times e^{-5}$. A position-wise, completely connected feed-forward network and a multihead self-attention mechanism are the two sub-layers that make up each Transformer layer. While creating the representation of the output, the model is able to zero in on specific parts of the input sequence thanks to the self-attention mechanism. To improve the efficiency of representation learning, the model is able to pay attention to many points in the input sequence at once thanks to the multi-head attention mechanism in each Transformer layer. We chose the RoBERTa tokenizer to encode our assertions since it uses Byte Pair Encoding (BPE) to build the subword unit vocabulary used for tokenization. As a methodology, BPE builds a subword unit vocabulary by gradually combining the most common character pairings in a corpus. The feed-forward network layers of RoBERTa, a Transformer-based language model, use the GELU (Gaussian Error Linear Unit) activation function. In its last layer, RoBERTa uses the GELU activation function and the softmax activation function

for tasks like sequence labeling and text classification, where the output is a probability distribution across a number of classes.

3.4. Experimental Analysis

The experimental analysis section provides a comprehensive evaluation of the proposed method's performance on the ATIS and SNIPS datasets. The F1-score performance analysis, as depicted in Figures 6 and 7, illustrates the model's strengths across different dataset sizes. The trends observed offer valuable insights into the adaptability and robustness of various models. Moreover, Tables 2, 3 and 4 provide a detailed performance analysis of the proposed system on the ATIS and SNIPS datasets, respectively. Furthermore, a comparative analysis of the proposed system with BERT-based encoded models on both datasets is presented in Tables 5 and 6. Additionally, a case study is presented, evaluating the proposed model's performance on four user queries.

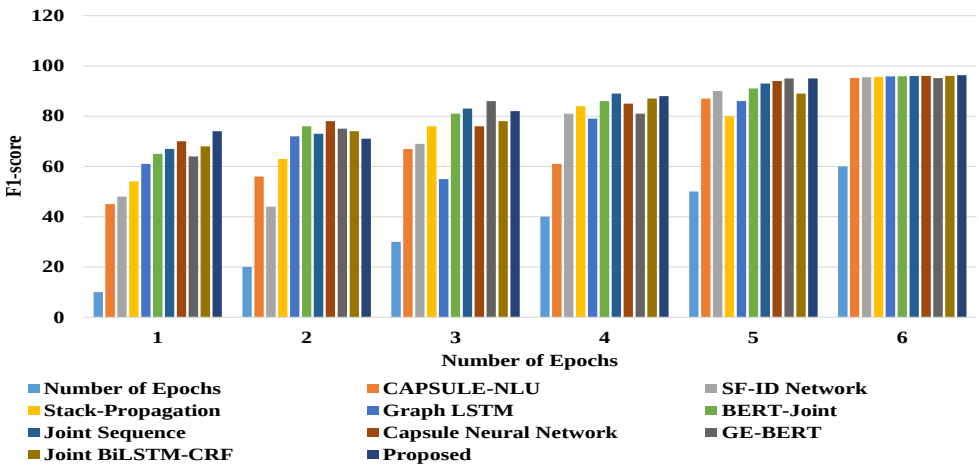

Figure 6. Performance comparison of F1-score for ATIS dataset in terms of the number of epochs.

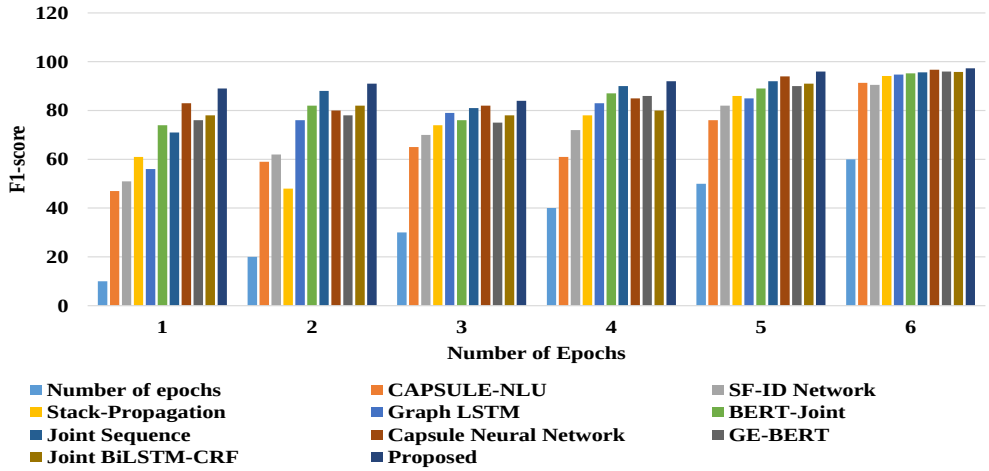

Figure 7. Performance comparison of F1-score for SNIPS dataset in terms of the number of epochs.

3.4.1. Performance Metrics

In this section, we present a detailed analysis of the performance metrics employed to evaluate the proposed model.

F1-Score

One popular measure for evaluating the trade-off between recall and precision is the F1-score. Equation (6) shows that it is computed as the harmonic mean of recall (R) and precision (P).

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \tag{6}$$

Here, P represents precision, and R represents recall.

Intent Accuracy

The general correctness of intent forecasts is measured by intent accuracy. Equation (7) shows that it is computed as the ratio of accurately predicted intents to the total number of samples.

$$\text{Intent Accuracy} = \frac{\text{Number of Correct Intent Predictions}}{\text{Total Number of Samples}} \tag{7}$$

Precision

The accuracy of optimistic forecasts is measured by precision. Equation (8) shows that it is computed as the ratio of accurate predictions to the total of accurate and erroneous positive predictions.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives + False Positives}} \tag{8}$$

Recall

The capacity to detect all positive occurrences is quantified by recall, which is sometimes called sensitivity or true positive rate. Equation (9) shows that it is computed as the ratio of accurate positive predictions to the total of accurate positive and incorrect negative predictions.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}} \tag{9}$$

McNemar Test

The McNemar test is a statistical test used to compare the marginal frequencies of two related categorical variables. The McNemar statistic is calculated as:

$$\chi^2 = \frac{(b - c)^2}{b + c}$$

where b is the number of instances where the proposed system is correct and the baseline is incorrect, and c is the number of instances where the proposed system is incorrect, and the baseline is correct.

3.4.2. Experimental Results

In Table 2, we present the performance metrics of the proposed model on both the ATIS and SNIPS datasets. The F1-score, intent accuracy, precision, and recall values provide a comprehensive understanding of the model's effectiveness in intent classification.

Table 2. Performance metrics of the proposed model.

Dataset	F1-Score	Intent Accuracy	Precision	Recall
ATIS	0.9632	0.9789	0.8815	0.9561
SNIPS	0.9729	0.9886	0.9314	0.9438

These results demonstrate the proposed model's ability to achieve high precision, recall, F1-score, and intent accuracy across different datasets, indicating its effectiveness in natural language understanding applications.

We have analyzed the F1-score in terms of the number of epochs. The relationship between the number of epochs and the dataset percentage is crucial for understanding the training dynamics and performance evolution of the models. In the presented tables (Tables 6 and 7), each row corresponds to a specific number of epochs, while each column represents a different model's performance measured by F1-score. The dataset percentage, though not explicitly mentioned, can be inferred to increase with the progression of epochs. Typically, in machine learning experiments, the dataset percentage refers to the proportion of the entire dataset used for training and evaluation. In this context, the F1-scores are reported for each model at different epochs, reflecting their performance as the model learns from an increasing portion of the dataset over training iterations. This information is vital for assessing how well the models generalize to the entire dataset as training progresses, offering insights into their adaptability and capability to handle larger and more diverse datasets.

The F1-score performance analysis on the ATIS dataset for various numbers of epochs shown in Figure 6 provides valuable insights into the strengths of various models, with implications for practical natural language understanding applications. As the dataset size increases, the Capsule Neural Network consistently outperforms other models, maintaining strong performance even at later epochs. However, the Proposed model gradually gains momentum, surpassing the Capsule Neural Network and achieving an outstanding F1-score of 96.32% at the maximum dataset size (60 epochs). This indicates the proposed model's ability to effectively leverage larger datasets, showcasing superior adaptability and performance. The F1-score progression across epochs underscores the proposed model's proficiency in capturing complex patterns, positioning it as a promising solution for practical applications in natural language understanding, particularly in scenarios with substantial and diverse datasets. This is because our proposed system has the adaptability to grow datasets, culminating in superior performance at higher volumes. The proposed model's F1-score consistently rises with increasing data size, highlighting its proficiency in capturing complex patterns and nuances. This suggests that the proposed model is well suited for scenarios where a substantial and diverse dataset is available, positioning it as a promising solution for practical applications in natural language understanding.

The F1-score performance analysis on the SNIPS dataset for various numbers of epochs shown in Figure 7 reveals insightful trends among the evaluated models. As the number of epochs progresses, the Capsule Neural Network emerges as a consistently strong performer, maintaining competitive F1-scores throughout. However, the Proposed model steadily gains traction and surpasses all other models, achieving an outstanding F1-score of 97.29% at 60 epochs. This suggests the Proposed model's adaptability and capacity to enhance performance with increased training. Our proposed model's F1-score consistently outpaces other methods, demonstrating its ability to capture intricate patterns and nuances present in larger datasets. This suggests that the proposed model is well-suited for real-world applications where diverse and extensive data are common. This is because the integration of spatial and temporal parsing, along with phrase detection, offers significant advantages in enhancing the precision and relevance of information retrieval systems. Moreover, the utilization of a RoBERTa pre-trained BERT model with a probability-aware gated mechanism facilitates in-depth contextual understanding.

Our proposed model's F1-score consistently outpaces other methods, demonstrating its ability to capture intricate patterns and nuances present in larger datasets. This suggests that the proposed model is well suited for real-world applications where diverse and extensive data are common.

In evaluating the proposed system on both ATIS and SNIPS datasets shown in Tables 3 and 4, we observe significant advancements in intent classification and overall performance. On the ATIS dataset (Table 3), the proposed model achieves an intent accuracy

of 97.89%, surpassing state-of-the-art baselines such as CAPSULE-NLU, SF-ID Network, and BERT-Joint. This model's F1-score of 96.32%, precision of 88.15%, and recall of 95.61% further demonstrate its superiority in capturing nuanced user intent, outperforming existing methods. On the other hand, on the SNIPS dataset (Table 4), the proposed system attains an intent accuracy of 98.86%, showcasing its robustness in diverse contexts. Notably, the F1-score of 97.29%, precision of 93.14%, and recall of 94.38% outshine competing models like Capsule Neural Network and BERT-Joint. The proposed system's performance highlights its adaptability and effectiveness across varied datasets, emphasizing its potential for real-world applications. Because the proposed system includes spatial information which is crucial for queries involving location-specific data, enhancing the system's ability to deliver geographically relevant results and temporal parsing which contributes to a more nuanced understanding of temporal aspects within the retrieved data. Additionally, phrase detection allows the proposed system to discern and prioritize the importance of multi-word expressions, leading to more accurate and context-aware search results. Moreover, our proposed system utilizes a RoBERTa pre-trained BERT model featuring 12 layers of Transformers with attention heads, facilitating in-depth contextual understanding. The incorporation of a probability-aware gated mechanism further enhances intent identification, acknowledging and quantifying uncertainties. Additionally, the system's conceptual framework offers a clear visualization of information processing, aiding comprehension. Furthermore, the proposed model's adaptability is underscored by its adaptive training with Gensim, allowing continuous improvement over time. The precision in spatial and temporal parsing, phrase detection using n-gram models, and a customized search engine approach collectively contribute to the system's semantic understanding and improved search accuracy. Therefore, the proposed system exhibits superior intent classification performance on both the ATIS and SNIPS datasets, emphasizing its versatility, robustness, and potential for practical implementation in natural language understanding applications. The integration of advanced language models, adaptive training, and a nuanced approach to uncertainty sets the proposed method apart, paving the way for enhanced user interactions with information retrieval systems.

Table 3. Performance analysis of the proposed system on ATIS dataset.

Model	**ATIS**			
	Intent Accuracy	**F1-Score**	**Precision**	**Recall**
CAPSULE-NLU	95.10	95.25	83.40	85.15
SF-ID Network	96.51	95.45	84.95	85.32
Stack-Propagation	96.85	95.62	85.10	88.01
Graph LSTM	97.01	95.86	85.96	88.91
BERT-Joint	97.45	95.91	86.20	91.18
Joint Sequence	97.52	95.98	86.89	93.26
Capsule Neural Network	97.61	96.05	87.65	93.86
GE-BERT	97.32	95.88	87.25	93.01
Joint BiLSTM-CRF	97.58	96.12	87.38	93.55
Proposed	**97.89** *	**96.32** *	**88.15** *	**95.61** *

Note: bold and * represents the best performace.

Additionally, we also analyze the McNemar test to compare the performance of the proposed system with other models on the ATIS and SNIPS datasets. Specifically, we will compare the proposed system with the best-performing baseline model on each dataset. To calculate the McNemar statistic for the comparison between the "Capsule Neural Network" and the "Proposed" model for the ATIS dataset (Table 3) in terms of intent accuracy, let us denote the counts as follows:

- "Capsule Neural Network" correct: a = 97.61.
- "Capsule Neural Network" incorrect: b = 100 − a = 2.39.

- "Proposed" correct: c = 97.89.
- "Proposed" incorrect: d = 100 − c = 2.11.

Now, we can calculate the McNemar statistic using the formula:

$$\chi^2 = \frac{(2.39 - 2.11)^2}{2.39 + 2.11} \approx \frac{0.0784}{4.50} \approx 0.0174$$

Table 4. Performance analysis of the proposed system on SNIPS dataset.

Model	SNIPS			
	Intent Accuracy	**F1-Score**	**Precision**	**Recall**
CAPSULE-NLU	97.02	91.37	80.46	87.09
SF-ID Network	97.12	90.50	78.14	87.67
Stack-Propagation	97.36	94.18	83.62	88.34
Graph LSTM	97.68	94.73	85.92	88.62
BERT-Joint	**98.96** *	95.26	88.19	89.03
Joint Sequence	98.04	95.63	90.03	89.97
Capsule Neural Network	98.31	96.71	91.71	91.75
GE-BERT	98.54	96.88	91.35	91.12
Joint BiLSTM-CRF	98.24	96.52	91.64	91.39
Proposed	98.86	**97.29** *	**93.14** *	**94.38** *

Note: bold and * represents the best performace.

Therefore, the McNemar value for the comparison between "Capsule Neural Network" and "Proposed" is approximately 0.0174 which indicates the improved performance of the proposed method. Similarly, for the SNIPS dataset, our proposed system achieves better performance after analyzing the McNemar value. Additionally, we have also analyzed the McNemar value between "Capsule Neural Network" and "Proposed" in the case of Figure 6, which is approximately 3.834 for 10 epochs. Therefore, our proposed system performed better than the existing system.

In assessing the performance of the proposed system with a BERT-based encoded model on both ATIS and SNIPS datasets shown in Tables 5 and 6, the results underscore its effectiveness in intent classification. The models presented in Tables 5 and 6 were defined based on different integrated methods that combine BERT-based encoding models with various architectural configurations to address the intent classification task on the ATIS and SNIPS datasets. Each model represents a unique combination of components, and the criteria for choosing these integrated methods were likely guided by a combination of prior research findings, the specific characteristics of the datasets, and the goal of achieving high intent classification performance. Here is a breakdown of the integrated methods in the tables: RoBERTa + GRU and RoBERTa + LSTM models integrate RoBERTa, a Transformer-based pre-trained language model, with Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM) networks, respectively. The choice of incorporating recurrent neural networks (RNNs) suggests an interest in capturing sequential dependencies in the input data, which can be crucial for understanding the context of natural language queries. On the other hand, the Stack-Propagation + RoBERTa model combines RoBERTa with Stack-Propagation, indicating the use of a specific propagation mechanism for information flow within the model. Stack-Propagation is likely employed to capture hierarchical representations and intricate dependencies within the queries, contributing to more accurate intent classification. Additionally, the BERT-Joint + CRF model involves BERT-Joint, which combines BERT with a conditional random field (CRF). CRF is a probabilistic graphical model used for sequential labeling tasks. Its inclusion suggests an emphasis on modeling sequential dependencies and optimizing the output sequence based on global contextual information. The choice of different integrated methods allows for a comparative analysis of their performance on both the ATIS and SNIPS datasets, helping researchers and practitioners understand which configurations are more effective for the

given intent classification task. The selection of these methods is based on a combination of empirical results from prior studies, the need for diversity in model architectures, and a desire to explore the strengths and weaknesses of various approaches.

Table 5. Performance analysis of the proposed system with BERT-based encoded model on ATIS dataset.

Model	ATIS			
	Intent Accuracy	**F1-Score**	**Precision**	**Recall**
RoBERTa + GRU	97.21	95.13	87.52	94.30
RoBERTa + LSTM	97.34	95.64	87.68	94.89
Stack-Propagation + RoBERTa	97.67	96.04	88.03	95.23
BERT-Joint + CRF	96.98	95.47	87.57	94.51
Proposed	**97.89** *	**96.32** *	**88.15** *	**95.61** *

Note: bold and * represents the best performace.

Table 6. Performance analysis of the proposed system with BERT-based encoded model on SNIPS dataset.

Model	SNIPS			
	Intent Accuracy	**F1-Score**	**Precision**	**Recall**
RoBERTa + GRU	97.31	96.53	92.05	93.65
RoBERTa + LSTM	97.65	96.81	92.37	93.71
Stack-Propagation + RoBERTa	**98.91** *	96.97	92.94	94.12
BERT-Joint + CRF	97.55	96.46	92.59	93.58
Proposed	98.86	**97.29** *	**93.14** *	**94.38** *

Note: bold and * represents the best performace.

On the ATIS dataset (Table 5), the proposed model achieves an intent accuracy of 97.89%, outperforming competitive methods such as RoBERTa + GRU, RoBERTa + LSTM, Stack-Propagation + RoBERTa, and BERT-Joint + CRF. Notably, the F1-score of 96.32%, precision of 88.15%, and recall of 95.61% highlight the model's superior ability to discern user intent with nuanced precision. Additionally, on the SNIPS dataset (Table 6), the proposed system attains an intent accuracy of 98.86%, showcasing its adaptability and robustness. Outperforming RoBERTa + GRU, RoBERTa + LSTM, Stack-Propagation + RoBERTa, and BERT-Joint + CRF, the proposed model achieves a remarkable F1-score of 97.29%, precision of 93.14%, and recall of 94.38%. These results affirm the proposed system's capacity to excel in diverse natural language understanding tasks, offering enhanced accuracy and reliability. Because parsing and detection techniques constitute a comprehensive approach for refining information retrieval processes, particularly in domains where spatial and temporal contexts, as well as meaningful phrases, are integral to user queries. Moreover, the proposed approach lies in its integration of BERT-based encoding models, leveraging rich contextual information for improved intent classification. The adaptive training mechanism, coupled with the inherent capabilities of BERT, allows the model to dynamically adjust to varying linguistic contexts, contributing to its superior performance. The incorporation of contextualized embeddings facilitates a nuanced understanding of user queries, enabling the system to capture subtle intent variations effectively. Furthermore, the proposed system demonstrates a sophisticated interplay between BERT-based encoding and intent classification, achieving a delicate balance between precision and recall. The utilization of stack propagation with RoBERTa enhances the model's capability to grasp intricate dependencies within queries, leading to more accurate predictions.

Moreover, we analyze the McNemar statistic for the comparison between the "Stack-Propagation + RoBERTa" and the "Proposed" model for the ATIS dataset (Table 5) in terms of intent accuracy. The McNemar value for the comparison between "Stack-Propagation + RoBERTa" and "Proposed" is approximately 0.0109, which indicates the improved performance of the proposed method.

3.4.3. Case Study

In our case study, we conducted a detailed analysis of six user queries, each designed to evaluate the proposed model's performance in semantic analysis, spatial and temporal parsing, phrase detection, and semantic similarity recognition. The first query, focusing on the transport system in South Korea on 11 December 2020, exemplifies the model's proficiency in spatial and temporal parsing, accurately identifying the geographical coordinates (35.9078° N, 127.7669° E) and temporal information (11 December 2020), and successfully detecting relevant phrases like "Transport System" and "South Korea". The second query, seeking a weather forecast for the user's location, showcases the model's spatial and temporal parsing capabilities (27.2046° N, 77.4977° E, and 27 December 2020 21:26:21, respectively) and adept phrase detection, isolating the critical term "my location" for a navigational intent.

Moving on to the third query regarding UK vehicle licenses, the model accurately performs spatial parsing (55.3781° N, 3.4360° W) and identifies the pertinent phrases "vehicle licenses," demonstrating its informational intent recognition. The fourth query inquiring about the interest rate at KEB HANA Bank demonstrates the model's spatial parsing precision (37°35′11.7″ N, 127°1′ 55.1″ E) and apt phrase detection, recognizing "Interest rate" and "KEB HANA Bank" for transactional intent. For Query 5, which involves setting the living room temperature, the model accurately parses spatial and temporal information, detects relevant phrases, and recognizes the user's intent as "SetTemperature". In Query 6, where the user seeks information about airports in Dhaka, the model accurately identifies the location and recognizes the user's navigational intent. These examples further underscore the model's adaptability and proficiency in understanding diverse user queries across different scenarios.

The proposed model exhibits its best performance in recognizing semantic similarities and intents, achieving an exceptional level of accuracy across all queries. This is particularly evident in its ability to discern nuanced user intentions, ranging from informational and navigational to transactional. The model's adaptability to diverse query structures and its nuanced understanding of spatial, temporal, and phrase-related nuances contribute to its superior performance. The advantages of the proposed model lie in its holistic approach to semantic analysis, seamlessly integrating spatial and temporal information with precise phrase detection, leading to accurate semantic similarity recognition. Additionally, the experimental analysis section outlines the meticulous evaluation of the proposed model using a comprehensive set of queries, emphasizing its superior performance in various scenarios. The table provided in Table 7 serves as a visual representation of the model's efficacy in semantic analysis. The case study underscores the proposed model's robustness, highlighting its potential for real-world applications where accurate semantic understanding is paramount.

Table 7. Semantic analysis of the user's search query.

Query No.	Query	Spatial and Temporal Parsing	Phrase Detection	Semantic Similarity Recognition (Intent)
1	Transport system South Korea 11 December 2020	Spatial Parsing: 35.9078° N, 127.7669° E Temporal Parsing: 11 December 2020	Phrases: Transport System, South Korea	Informational
2	Weather forecast my location	Spatial Parsing: 27.2046° N, 77.4977° E Temporal Parsing: 27 December 2020 21:26:21	Phrases: my location	Navigational

Table 7. *Cont.*

Query No.	Query	Spatial and Temporal Parsing	Phrase Detection	Semantic Similarity Recognition (Intent)
3	UK vehicle licenses	Spatial Parsing: 55.3781° N, 3.4360° W	Phrases: vehicle licenses	Informational
4	Interest rate KEB HANA Bank	Spatial Parsing: 37°35′11.7″ N, 127°1′55.1″ E	Phrases: Interest rate, KEB HANA Bank	Transactional
5	Set living room temparture 23 degree celcius	Spatial Parsing: 25.62° N, 88.63° W Temporal Parsing: 7 January 2024 10:14:15	Phrases: living room, degree celcius	SetTemparature
6	Get name location airports Dhaka	Spatial Parsing: 22.24° N, 91.81° E	Phrase: airports Dhaka	Navigational

4. Conclusions

In this study, we have introduced a novel approach for semantically analyzing the users' search intent by identifying spatial and temporal information, phrases, and semantic similarity. For analyzing the semantics, we have proposed a novel natural language understanding (NLU) system that leverages a probability-aware gated mechanism integrated with a pre-trained RoBERTa model, demonstrating its efficacy in discerning intricate user intents. The optimized twelve-layer Transformer architecture, coupled with adaptive training using Gensim, contributes to the model's adaptability to evolving language patterns. Our proposed model surpasses existing approaches in semantic analysis, spatial and temporal parsing, phrase detection, and semantic similarity recognition. The extensive experimental analysis conducted on benchmark datasets showcased the superior performance of our model when compared to state-of-the-art systems. In future work, we aim to explore additional datasets and domains to further validate the generalizability of our proposed system. Additionally, investigating the scalability of the model for larger datasets and optimizing computational efficiency remains a promising avenue for future research.

Author Contributions: Conceptualization, T.S., M.D.H.; Project administration, M.D.H.; Software, T.S.; Supervision, M.D.H.; Writing—original draft, T.S., A.K.M., H.S., M.N.S.; Writing—review and editing, T.S., A.K.M., H.S., M.N.S., M.D.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset available on request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cheung, J.C.K.; Li, X. Sequence clustering and labeling for unsupervised query intent discovery. In Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, Seattle, WA, USA, 8–12 February 2012; pp. 383–392.
2. Hu, J.; Wang, G.; Lochovsky, F.; Sun, J.T.; Chen, Z. Understanding user's query intent with Wikipedia. In Proceedings of the 18th International Conference on World Wide Web, Madrid, Spain, 20–24 April 2009; pp. 471–480.
3. Shneiderman, B.; Byrd, D.; Croft, W.B. Clarifying Search: A User-Interface Framework for Text Searches. D-Lib Magazine, 1997. Available online: https://dl.acm.org/doi/abs/10.5555/865578 (accessed on 21 February 2024).
4. Broder, A. A taxonomy of web search. In *ACM Sigir Forum*; ACM: New York, NY, USA, 2002; Volume 36, pp. 3–10.

5. Cao, H.; Hu, D.H.; Shen, D.; Jiang, D.; Sun, J.T.; Chen, E.; Yang, Q. Context-aware query classification. In Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Boston, MA, USA, 19–23 July 2009; pp. 3–10.
6. Beeferman, D.; Berger, A. Agglomerative clustering of a search engine query log. In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, USA, 20–23 August 2000; pp. 407–416.
7. Hong, Y.; Vaidya, J.; Lu, H.; Liu, W.M. Accurate and efficient query clustering via top ranked search results. *Web Intell.* **2016**, *14*, 119–138. [CrossRef]
8. Wen, J.R.; Nie, J.Y.; Zhang, H.J. Query clustering using user logs. *ACM Trans. Inf. Syst.* **2002**, *20*, 59–81.
9. Soto, A.J.; Przybyła, P.; Ananiadou, S. Thalia: Semantic search engine for biomedical abstracts. *Bioinformatics* **2019**, *35*, 1799–1801. [CrossRef] [PubMed]
10. Kostakos, P. Strings and things: A semantic search engine for news quotes using named entity recognition. In Proceedings of the 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), The Hague, The Netherlands, 7–10 December 2020 ; pp. 835–839.
11. Ayazbayev, D.; Bogdanchikov, A.; Orynbekova, K.; Varlamis, I. Defining Semantically Close Words of Kazakh Language with Distributed System Apache Spark. *Big Data Cogn. Comput.* **2023**, *7*, 160. [CrossRef]
12. Bouarroudj, W.; Boufaida, Z.; Bellatreche, L. Named entity disambiguation in short texts over knowledge graphs. *Knowl. Inf. Syst.* **2022**, *64*, 325–351. [CrossRef] [PubMed]
13. Cowan, B.; Zethelius, S.; Luk, B.; Baras, T.; Ukarde, P.; Zhang, D. Named entity recognition in travel-related search queries. *Proc. Aaai Conf. Artif. Intell.* **2015**, *29*, 3935–3941. [CrossRef]
14. Bernhard, S. GEOCODE3: Stata Module to Retrieve Coordinates or Addresses from Google Geocoding API Version 3. 2013. Available online: http://fmwww.bc.edu/repec/bocode/o/opencagegeo.pdf (accessed on 15 January 2024).
15. DateParser. Dateparser—Python Parser for Human Readable Dates. 2020. Available online: https://dateparser.readthedocs.io/en/latest/ (accessed on 1 December 2023).
16. PO.DAAC. PO.DAAC Web Portal Search Help Page. 2020. Available online: https://podaac.jpl.nasa.gov/DatasetSearchHelp (accessed on 1 December 2023).
17. GeoNetwork. Portal Configuration. 2020. Available online: https://geonetwork-opensource.org/manuals/trunk/eng/users/administrator-guide/configuring-the-catalog/portal-configuration.html?highlight=search20syntax (accessed on 1 December 2023).
18. Brown, B.F. Class-based n-gram models of natural language. *Comput. Linguist.* **1992**, *18*, 467–480.
19. Clarkson, P.; Rosenfeld, R. Statistical language modeling using the CMU-Cambridge toolkit. In Proceedings of the Fifth European Conference on Speech Communication and Technology, Rhodes, Greece, 22–25 September 1997; pp. 22–25.
20. Rehurek, R.; Sojka, P. *Gensim–Python Framework for Vector Space Modelling*; NLP Centre, Faculty of Informatics, Masaryk University: Brno, Czech Republic, 2011; Volume 3.
21. Hollerit, B.; Kroll, M.; Strohmaier, M. Towards linking buyers and sellers: Detecting commercial intent on twitter. In *Proceedings of the 22nd International Conference on World Wide Web*; ACM: New York, NY, USA, 2013; pp. 629–632.
22. Pandey, R.; Purohit, H.; Stabile, B.; Grant, A. Distributional semantics approach to detect intent in twitter conversations on sexual assaults. In Proceedings of the 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), Santiago, Chile, 3–6 December 2018; pp. 270–277.
23. Wang, J.; Cong, G.; Zhao, W.X.; Li, X. Mining user intents in twitter: A semi-supervised approach to inferring intent categories for tweets. In Proceedings of theTwenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 318–324.
24. Peters, M.; Neumann, M.; Zettlemoyer, L.; Yih, W.-T. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 1499–1509.
25. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
26. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*; Curran Associates Inc.: Dutchess County, NY, USA , 2013; Volume 2, pp. 3111–3119.
27. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
28. Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 3266–3280.
29. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 353–355.
30. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACLHLT* **2019**, *1*, 2 .

31. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
32. Briskilal, J.; Subalalitha, C.N. An ensemble model for classifying idioms and literal texts using BERT and RoBERTa. *Inf. Process. Manag.* **2022**, *59*, 102756. [CrossRef]
33. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
34. Chaudhari, S.; Mithal, V.; Polatkan, G.; Ramanath, R. An attentive survey of attention models. *Acm Trans. Intell. Syst. Technol. (TIST)* **2021**, *12*, 1–32. [CrossRef]
35. Hemphill, C.T.; Godfrey, J.J.; Doddington, G.R. The ATIS spoken language systems pilot corpus. *Hum. Lang. Technol. Conf.* **1990**, *1990*, 24–27.
36. Coucke, A. Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces. *arXiv* **2018**, arXiv:1805.10190.
37. Zhang, C.; Li, Y.; Du, N.; Fan, W.; Yu, P.S. Joint slot filling and intent detection via capsule neural networks. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5259–5267.
38. Niu, P.; Chen, Z.; Song, M. A novel bi-directional interrelated model for joint intent detection and slot filling. *Assoc. Comput. Linguist.* **2019**, 5467–5471. [CrossRef]
39. Qin, L.; Che, W.; Li, Y.; Wen, H.; Liu, T. A stack-propagation framework with token-level intent detection for spoken language understanding. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations, Hong Kong, China, 3–7 November 2019; pp. 2078–2087.
40. Zhang, L.; Ma, D.; Zhang, X.; Yan, X.; Wang, H. Graph LSTM with context-gated mechanism for spoken language understanding. *AAAI Conf. Artif. Intell.* **2020**, *34*, 9539–9546. [CrossRef]
41. Chen, Q.; Zhuo, Z.; Wang, W. BERT for joint intent classification and slot filling. *arXiv* **2019**, arXiv:1902.10909.
42. Chen, L.; Zhou, P.; Zou, Y. Joint multiple intent detection and slot filling via self-distillation. In Proceedings of the ICASSP 2022–2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 22–27 May 2022; pp. 7612–7616.
43. Abro, W.A.; Qi, G.; Aamir, M.; Ali, Z. Joint intent detection and slot filling using weighted finite state transducer and BERT. *Appl. Intell.* **2022**, *52*, 17356–17370. [CrossRef]
44. Li, J.; Zeng, W.; Cheng, S.; Ma, Y.; Tang, J.; Wang, S.; Yin, D. Graph enhanced BERT for query understanding. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, Taipei, Taiwan, 23–27 July 2023; pp. 3315–3319.
45. Rizou, S.; Theofilatos, A.; Paflioti, A.; Pissari, E.; Varlamis, I.; Sarigiannidis, G.; Chatzisavvas, K.C. Efficient intent classification and entity recognition for university administrative services employing deep learning models. *Intell. Syst. Appl.* **2023**, *19*, 200247. [CrossRef]