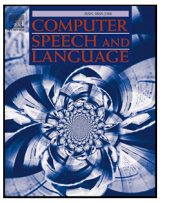
Computer Speech and Language 89 (2025) 101693

Contents lists available at ScienceDirect

Computer Speech & Language

journal homepage: www.elsevier.com/locate/csl



On improving conversational interfaces in educational systems

Yuyan Wu *, Romina Soledad Albornoz-De Luise, Miguel Arevalillo-Herráez

Departament d'Informàtica, Universitat de València, Spain

ARTICLE INFO

Keywords:
Conversational agent
Intent classification
Entity extraction
Entity–quantity matching
Ensemble models
Semantic accuracy

ABSTRACT

Conversational Intelligent Tutoring Systems (CITS) have drawn increasing interest in education because of their capacity to tailor learning experiences, improve user engagement, and contribute to the effective transfer of knowledge. Conversational agents employ advanced natural language techniques to engage in a convincing human-like tutorial conversation. In solving math word problems, a significant challenge arises in enabling the system to understand user utterances and accurately map extracted entities to the essential problem quantities required for problem-solving, despite the inherent ambiguity of human natural language. In this study, we propose two possible approaches to enhance the performance of a particular CITS designed to teach learners to solve arithmetic–algebraic word problems. Firstly, we propose an ensemble approach to intent classification and entity extraction, which combines the predictions made by two distinct individual models that use constraints defined by human experts. This approach leverages the intertwined nature of the intents and entities to yield a comprehensive understanding of the user's utterance, ultimately aiming to enhance semantic accuracy. Secondly, we introduce an adapted Term Frequency-Inverse Document Frequency technique to associate entities with problem quantity descriptions. The evaluation was conducted on the AWPS and MATH-HINTS datasets, containing conversational data and a collection of arithmetical and algebraic math problems, respectively. The results demonstrate that the proposed ensemble approach outperforms individual models, and the proposed method for entity–quantity matching surpasses the performance of typical text semantic embedding models.

1. Introduction

With the rapid advancement of technology, approaches to teaching have gradually changed. In contrast to the traditional method of teachers imparting knowledge to students in a classroom and students passively receiving it, today's students can actively acquire knowledge through Intelligent Tutoring Systems (ITS). These systems incorporate artificial intelligence techniques into educational software to emulate human tutors, teach various subjects, and design personalized learning plans for students based on their proficiency levels to enhance their learning experiences.

Conversational Intelligent Tutoring Systems (CITS) are software applications that employ natural language to have text or voice interactions with users, acting as virtual tutors. Among the most renowned examples of CITS are AutoTutor (Nye et al., 2014) and its descendants, including Atlas (VanLehn et al., 2000) and Watson Tutor (Ahn et al., 2019). The benefits of incorporating natural language interfaces into ITS go beyond merely being a more convenient form of interaction. In addition to reported educational gains (Paladines and Ramirez, 2020), they present a valuable opportunity for delivering unsensorized emotional support by analyzing user expressions, with the potential to outperform the benefits of existing proposals, such as Arevalillo-Herráez et al. (2017).

* Corresponding author.
E-mail addresses: yuyan.wu@uv.es (Y. Wu), romina.albornoz@uv.es (R.S. Albornoz-De Luise), miguel.arevalillo@uv.es (M. Arevalillo-Herráez).

https://doi.org/10.1016/j.csl.2024.101693
Received 25 October 2023; Received in revised form 27 March 2024; Accepted 21 July 2024
Available online 23 July 2024

Recent advancements in Natural Language Processing (NLP), Natural Language Understanding (NLU), Natural Language Generation (NLG), and Artificial Intelligence (AI), coupled with the increasing availability of powerful computers, have led to the widespread application of conversational systems across various domains. These developments have facilitated the creation of tools that streamline and expedite the processing of the aforementioned tasks within specialized systems. The creation of virtual agents that can mimic real conversations is supported by a variety of existing toolkits and frameworks (Katchapakirin and Anutariya, 2018; González-Castro et al., 2021; Paladines et al., 2021; Topal et al., 2021; Albornoz-De Luise et al., 2023), including DialogFlow,[1] Amazon Lex,[2] IBM Watson,[3] and Rasa,[4] among others.

Within this domain, the NLU component plays a crucial role in understanding user utterances. It involves critical tasks such as intent classification and entity extraction. Intent classification consists of predicting the purpose or goal behind user inputs, while entity extraction focuses on detecting and classifying specific words or portions of text into predefined entity labels. Many recent studies have been dedicated to enhancing the performance of NLU (Hendrycks and Gimpel, 2018; Zhang et al., 2019; Abro et al., 2020; Xin et al., 2021; Varshney et al., 2022; Abro et al., 2022; Lim et al., 2022), reflecting its pivotal role across diverse applications (Jung, 2019; Dowlagar and Mamidi, 2023).

In mathematics education, word problems pose one of the most challenging mathematical problems for learners (Verschaffel et al., 2020). Mathematical Word Problems (MWPs) depict the properties of entities and their relationships in natural language and terminate with one or more quantitative questions. Quantities are essential variables that make up a math problem to solve. A MWP consists of known quantities (information given in the problem statement), unknown quantities (variables that need to be solved through known quantities) and their relationships. In CITS, the system must understand the semantic information of the user's input and match specific extracted portions of text with a quantity of the problem. We call this problem the entity–quantity matching problem. The typical way to solve this problem is to compute the text similarity between the user-input descriptions and the quantities of the problem. One solution is using text semantic embedding models converting the natural language to numerical features and comparing them with similarity functions such as Reimers and Gurevych (2020). However, if the length of the quantity description is short, it leads to an increase in the ambiguity of the meaning comprehension.

This study proposes an enhancement of the NLU component of an existing conversational system (Albornoz-De Luise et al., 2023), whose operation is supported by the open-source Rasa framework (Bocklisch et al., 2017). This system has recently been integrated into an ITS known as Hypergraph-Based Problem Solver (HBPS) (Arevalillo-Herraez et al., 2013; Arnau et al., 2013, 2014), which marked a turning point in the development of computational tools for resolving algebraic and arithmetic word problems. It adopts an unguided approach that offers students the freedom to choose the letters used to represent the quantities needed to tackle a problem, granting them significant autonomy in choosing their problem-solving strategies. In addition, HBPS does not impose restrictions on the solution path, the order in which quantity values are computed or the problem-solving reasoning applied.

Even though the conversational system of HBPS achieved acceptable performance (Albornoz-De Luise et al., 2023), there still exists untapped potential for enhancing its overall effectiveness. Although entity labels and intents are often intertwined, with specific entities in a sentence influenced by the underlying intent expressed, current joint NLU methods do not fully exploit the interconnected nature of entities and intents. Our study suggests that employing two distinct NLU models, each trained for the combined tasks of intent classification and entity extraction, and integrating their outputs through post-processing with domain-specific knowledge from human experts could help identify and rectify incongruent predictions, ensuring consistent results. Consequently, this approach could enhance the semantic accuracy of the conversational system.

Moreover, we propose to refine the alignment of user-input entity descriptions with problem quantities as required by the HBPS system. This refinement process entails mapping specific types of descriptions, identified by the NLU component, to predefined quantity descriptions involved in a problem solution. We hypothesize that an adapted Term Frequency-Inverse Document Frequency (TF-IDF) technique or employing recent semantic language models, such as SGPT and SBERT, for NLU tasks could enhance the effectiveness of the TF-IDF approach currently utilized in the HBPS system.

Our hypothesis proposes a dual approach to enhance the performance of the CITS HBPS: first, by refining the conversational system through adaptation to the specific domain and the injection of knowledge provided by human experts; second, by optimizing the entity–quantity matching process in the HBPS with an alternative method.

The remainder of this article is organized as follows. Section 2 introduces the related work of NLU and CITS. Section 3 presents an overview of a CITS that we used for this study. Section 4 elaborates on the proposed methods. The experiment results and accompanying discussion are described in Section 5. Finally, the conclusion and future works are presented in Section 6.

2. Related work

CITS represents cutting-edge e-learning platforms that employ a natural language interface to provide adapted tutorial conversations. While conversational agents (CA) facilitate intuitive communication with learners through the use of natural language and enable the ability to ask questions, its integration introduces considerable complexity and consequently extends the development time for CITS (Latham, 2022). To engage in a convincing human-like tutorial conversation, CITS requires the application of sophisticated natural language techniques. Various approaches exist for developing CA, including rule-based methods, generative methods (supervised or reinforcement learning from conversation data), pattern matching (O'Shea et al., 2011), latent semantic analysis (Wiemer-Hastings et al., 2004), and NLP (Rus et al., 2013).

[1] https://dialogflow.com.

[2] https://aws.amazon.com/es/lex.

[3] https://www.ibm.com/watson.

[4] https://rasa.com.

Table 1
Example of an utterance, accompanied by their corresponding intent label and entities.

Utterance	We	can	define	x	as	the	current	age	of	Noah
Tag token predictions	O	O	O	U-letter	O	O	B-description	I-description	I-description	L-description
Entities	{**letter**, x} {**description**, current age of Noah}									
Intent	define_letter									

2.1. Natural language understanding

Within the complex field of NLP, NLU constitutes a subset concerned with the machine comprehension of human language (Rozga, 2018). Intent classification and entity extraction are the main tasks in the NLU module (Ni et al., 2020; Rafiepour and Sartakhti, 2023; Ni et al., 2023). Intent classification can be considered as a text classification task whose aim is to analyze sentences with the origin of spoken or written utterances in the conversation context (Weld et al., 2022). On the other hand, entity extraction, also known as slot filling, is a sequence-to-sequence task that translates raw user input information into a sequence of entity names. As exemplified in Table 1, given an utterance from the user, "We can define x as the current age of Noah", intent classification operates at the utterance level, assigning a categorical intent target to the entire utterance, such as the target "define_letter" in this case. On the contrary, entity extraction works at the token level, considering words as the basic tokens and marking tokens with specific entity types. The prediction result is then converted into a readable format. The BILOU technique (Ramshaw and Marcus, 1999) uses a B tag for the beginning of the entity, an I tag for tokens inside the entity, an L for the end of the entity, an O for non-entity tokens, and a U for entities consisting of a single word. Table 1 provides an illustrative example of the application of the BILOU schema, followed by the classification of text spans into their respective entity types. For instance, the word "x" is placed as an entity of type "letter", and "current age of Noah" is determined as an entity of type "description".

In traditional approaches, intent classification and entity extraction are handled as separate tasks using pipelined models (Bhargava et al., 2013; Jose and Lakshmi, 2018; Weld et al., 2022). The most straightforward approach to implementing intent classification involves manually designed rule templates and category information (Li and Roth, 2006). While this method can achieve relatively high accuracy without the need for extensive data, it becomes increasingly complex as the dataset grows, demanding substantial human resources for rule updates and making manual maintenance more challenging (Li and Roth, 2006). To address this issue, researchers have explored alternative methods. For instance, Zhang et al. (2015) employed Convolutional Neural Networks (CNN) to tackle text classification problems. Ravuri and Stolcke (2015) integrated Recurrent Neural Networks (RNN) and Long Short-Term Memory Networks (LSTM) to improve prediction performance. Xia et al. (2018) introduced a self-attention mechanism for semantic feature extraction and employed a capsule-based neural network for intent classification.

Regarding entity extraction approaches, RNNs hold significant prominence. They incorporate Conditional Random Fields (CRF) in conjunction with LSTM to enhance the experimental outcomes of sequence annotation. Kurata et al. (2016) proposed an encoder-labeler LSTM approach, leveraging an encoder model to transform input sequences into fixed-length vectors. These vectors were subsequently fed into an LSTM for predicting sequence labels. Zhu and Yu (2017) introduced the attention mechanism into encoder–decoder architectures. Their design featured a Bidirectional LSTM as the encoder model and an LSTM as the decoder model, resulting in a commendable performance.

Nevertheless, treating intent classification and entity extraction as separate tasks increases the complexity of NLU and overlooks their interdependence. To address this challenge, researchers have devised joint models with parallel training strategies. Liu and Lane (2016) introduced an attention module within Bidirectional LSTM networks, improving both intent classification and entity extraction by providing precise focus. Kane et al. (2020) proposed a new approach called CoBiC, using the combination of CNN, Bidirectional LSTM, and CRF for intent classification and entity extraction.

Furthermore, since the introduction of transformer-based architectures by Vaswani et al. (2017), there has been a significant advancement in NLP tasks (Patwardhan et al., 2023). Particularly, in recent years, numerous highly effective unsupervised training methods have emerged to address the challenge of label sparsity. These are primarily based on large-scale language models renowned for their remarkable generalization capabilities. These models excel in extracting semantic features from text, which can be effectively utilized in various downstream tasks. Notable examples include BERT (Devlin et al., 2018), GPT (Radford et al., 2018, 2019; Brown et al., 2020), and XLNet (Yang et al., 2019). These techniques have also found successful application in the domains of intent classification and entity extraction, yielding promising experimental results. Chen et al. (2019) integrated CRF atop the BERT model to jointly address intent classification and entity extraction. Wang et al. (2021) introduced a transformer encoder-based approach that incorporates syntactical knowledge via multitask learning to enhance intent classification and entity extraction.

However, the utilization of ensemble approaches for jointly addressing intent classification and entity extraction remains underexplored. In this work, we introduce a novel ensemble approach using transformer-based models to enhance the performance of both tasks. We incorporate BERT (Devlin et al., 2018) and DistilBERT (Sanh et al., 2020) in the NLU pipeline of RASA to evaluate the performance on the AWPS dataset. This approach combines predictions from two distinct models, adhering to predefined constraints incorporated by human experts.

2.2. Conversational intelligent tutoring systems and semantic models

About CITS, one major distinctive feature lies in evaluating students' responses (Brajković and Vasić, 2017; Emil Brajković, Daniel Vasić, 2018). Different approaches are used to assess the semantics of students' natural language answers, and they can be used inside modules of ITS (Volarić et al., 2017). For example, AutoTutor (Nye et al., 2014) relied on latent semantic analysis, regular expressions, and frequency-weighted content word overlap for semantic models that compare students' verbal answers to expectations and misconceptions. More recently, several computational semantic models have been developed which use neural network approaches to represent natural language text in continuous vector space.

Computational semantic models can be categorized into two primary groups: word-level and sentence-level approaches. Word-level approaches involve the amalgamation of word embedding models, encompassing well-known ones like word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and fastText (Joulin et al., 2016). These models are then enhanced by incorporating weighting functions, such as averaging, to create sentence embeddings. However, this method tends to overlook the crucial aspect of word order within text representations. In response to this limitation, sentence-level approaches have emerged as a solution. The prevailing approach to training sentence-level embedding models entails the fusion of a language model with Siamese networks during the training process. Subsequently, a single model is employed to generate sentence embeddings, effectively addressing the challenge of capturing both semantic meaning and word order. For instance, Infersent (Conneau et al., 2017) leverages a supervised recurrent neural network to predict semantic relationships between pairs of phrases. Universal Sentence Encoder (Cer et al., 2018) follows a similar architectural approach to Infersent, but it replaces bidirectional Long Short-Term Memory with transformers and a deep averaging network. Sentence-BERT (SBERT) (Reimers and Gurevych, 2020), introduced by Reimers and Gurevych, utilizes siamese networks with BERT as its foundation. Meanwhile, SGPT (Muennighoff, 2022) employs a decoder-transformer framework to generate sentence embeddings. Nevertheless, within the context of short sentence text similarity, there is still room for improvement in these methods.

In recent years, the hypergraph-based approach has gained prominence as a method for representing word problem-solving paths. This approach has shown significant potential in capturing intricate relationships between data (Arnau et al., 2013; Arevalillo-Herraez et al., 2013; Arnau et al., 2014). When a conversational system is integrated into a system with these specific requirements, it is crucial to map the user entities extracted by the NLU component to predefined descriptions that refer to the quantities involved in a problem solution. This alignment process is known as entity–quantity matching. In the context of problem-solving tutoring, students are required to define these quantities and establish their relationships within the problem context. This task is defined as identifying the most semantically similar quantity that corresponds to the student's input entity. The mapping of entity and quantity makes the system control and guide the student's problem-solving process. To solve the entity–quantity matching, we propose an adapted TF-IDF technique that can effectively match a student's entity with quantities leveraging problem constraint-based information.

It is important to note that the NLU modules are crucial for the task of understanding user utterances. The correlation of the modules of the system itself can make the occurrence of errors continue to accumulate and thus impact the overall performance (Li et al., 2017). In our system, the main errors that propagate to the other components come from an error in the detection of the intents and/or the entities, together with errors in assessing sentence similarity.

The suggested strategic approaches serve a dual purpose, not only enhancing the overall performance of the system but also enhancing the user experience by alleviating potential frustration from ambiguous or unrecognized input. Therefore, the combination of these intelligent response mechanisms greatly improves the effectiveness and user-friendliness of the system.

3. Overview of HBPS CITS

HBPS aims to tutor individuals in solving problems using natural language (Albornoz-De Luise et al., 2023; Arnau-González et al., 2023; Albornoz-De Luise et al., 2022b,a). It is supported by a domain-specific inference engine that uses a knowledge representation of the problem based on hypergraphs (Arevalillo-Herraez et al., 2013; Arnau et al., 2013, 2014). Its engine is capable of supervising the resolution of arithmetic–algebraic word problems that can be reduced to a set of ternary relations between quantities, by using a knowledge base for each problem that consists of a set of quantities and a set of existing relations between quantities, which are internally represented as a hypergraph. This knowledge base also contains a series of textual descriptions for each quantity, containing various ways a student could refer to them.

A screenshot of the system is shown in Fig. 1. The Graphical User Interface (GUI) is divided into four sections. The problem statement is displayed at the top of the screen. To the left, is the Notes section, which contains descriptions of the identified problem quantities and any supplementary information for the user's reference. Below, the Equations section houses all the correct equations introduced by the student. Finally, on the right-hand side, there is a chat interface.

An illustrative example is given to help better understand the tasks related to the improvement of the HBPS CITS in this work. Consider the problem statement "Noah's mother is four times as old as Noah. Six years ago, she was ten times older. How old is Noah?", presented at the top of the screen. Users have the flexibility to determine the order in which quantities are declared, assign to them either numbers or letters, and decide which unknown quantity will be represented by a letter, as well as the number of letters they will employ.

When the user submits a new utterance, such as "We can define x as the current age of Noah", it is forwarded to the implemented Rasa service (Albornoz-De Luise et al., 2023). This service returns the corresponding intent and entities extracted from the message. In this case, the system would predict the intent "define_letter", and extract the letter "x" along with the description "current age of

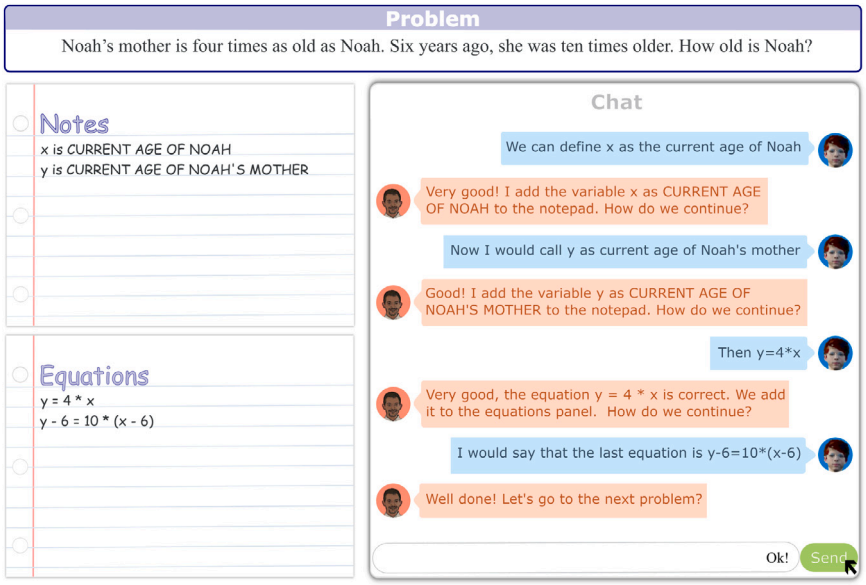


Fig. 1. Example of a conversation between the CITS and a student.

Noah" as entities associated with the intent. This output is then examined for correctness and used by other components to formulate an appropriate response.

At this specific stage, the system verifies that the designated letter has not been used previously and must align the entity of the type description with one of the predefined entries in the knowledge base of the problem statement. In this case, there are three predefined quantity descriptions in the problem domain, namely, "Noah's current age", "Noah's present age", and "actual age of Noah". When the system extracts the entity of type description with the value "current age of Noah", it needs to associate this with the quantity description "Noah's current age" which has the strongest semantic relevance. It then delivers a predefined response based on these evaluations. Moreover, in cases where an extracted entity of type description does not readily align with any of the predetermined description quantities, it is the system's responsibility to proactively propose alternative options to the user such as an "I don't understand" response.

If the student correctly assigns a value, letter, or expression to an unknown quantity, the name of the quantity and the expression that has been associated with it are represented in the Notes section. If a correct expression is entered without a description, the quantity that it represents is inferred, and a textual description is automatically assigned to it. For instance, if a user submits a message containing an equation, such as "Then, $y = 4 \cdot x$", and the NLU component returns the intent "equations" along with an entity type equation containing the value "$y = 4 \cdot x$", the program verifies its validity and, if applicable, provides a message confirming the correctness of the formulation.

The hypergraph representation of the mathematical structure for this word problem is depicted in Fig. 2. In this example, the nodes with the labels A and B represent the unknown quantities of the current ages of Noah and the current age of Noah's mother, respectively; and the nodes with the labels C and D represent the unknown quantities of the past ages of Noah and past age of Noah's mother, respectively.

The horizontal pink edge represents the existing relationship between the current and past ages of Noah 6 years ago ($A = C + 6$), and the horizontal green edge represents the same relationship between the current and past ages of Noah's mother 6 years ago ($B = D + 6$). In contrast, the vertical edges express the relationships between Noah's age and his mother's, both now and six years ago ($B = 4 \cdot A$ and $D = 10 \cdot C$). When these relationships are revealed, the problem is solved.

To sum up, the problem-solving procedure can be delineated into four stages. Initially, the system identifies the intent and extracts entities from the user's input. Next, match the quantity of entities and map the entities to the quantity of corresponding problems to update the progress of solving the problem. Following this, the acquired information is transformed and input into the program for validation. Finally, a corresponding response is generated based on the processed information.

4. Proposed methods

In this section, we elaborate on our strategies for enhancing intent classification and entity extraction. Furthermore, we provide an explanation of the proposed adapted TF-IDF method designed to tackle the entity–quantity matching task.

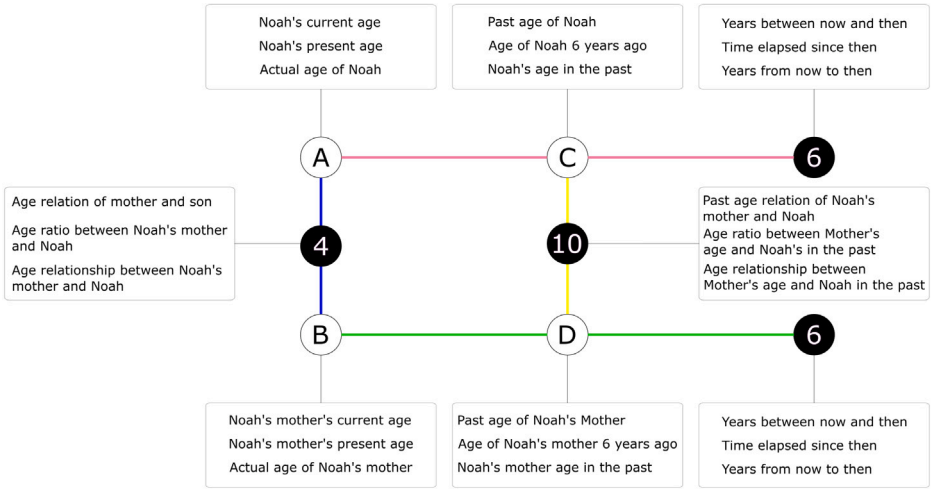


Fig. 2. Hypergraph which represents the mathematical structure of the example problem depicted in Fig. 1. Known quantities are represented by black-filled dots, while unknown quantities by clear ones. Each color of the connection line represents a relationship between the quantities. Each square contains multiple descriptions, all of which refer to the same quantity. In this illustration, the pink edge represents Noah's current age which is defined as the sum of his age six years ago and six ($A = C + 6$). The blue edge defines Noah's mother's current age, which is four times Noah's current age ($B = 4 \cdot A$). The yellow edge represents the past age of Noah's mother, which was ten times Noah's past age ($D = 10 \cdot C$), and the green edge defines Noah's mother's current age as her age six years ago plus six ($B = D + 6$).

4.1. Strategies for enhanced intent classification and entity extraction

Given a tokenized input utterance $x = \{w_1, w_2, \ldots, w_T\}$, where T is the total number of words. Let us assume that our NLU system considers a set of intent classes $I = \{i_1, i_2, \ldots, i_N\}$, and a set of entity classes $E = \{e_1, e_2, \ldots, e_L\}$, where N is the total number of intent classes and L is the total number of entity classes. Formally, intent classification involves assigning an intent label i_j to the input utterance, while entity extraction focuses on assigning entity class e_k to each token w_l and classifies text spans into their respective entity types. The prediction result is then converted into a readable format, indicating the entity type e_k and its corresponding value v_k. This value can comprise one or more words, as shown in Table 1.

To elucidate the relationship between intent and the permissible quantity of associated entities, we use propositional logic to define correlation constraints between intents and entities and introduce two approaches, namely, the constraint-based approach and the ensemble-based approach. Consider the scenario shown in Table 1, where an input pertains to a define_letter query, necessitating both a letter and a description entity. This requirement can be formally expressed using propositional logic as follows:

$$\text{letter} = 1 \wedge \text{description} = 1. \tag{1}$$

Here, the logical operator *and* is denoted by "$\wedge$" and when any entity type is absent from the formulation, it is assumed to satisfy the condition "necessarily equals 0". The complexity of the logical structure underlying each intent may exhibit considerable variation due to the inherent specificities of the current system. As an example, Eq. (1) could be interpreted with slightly broader constraints. For instance, a system could accept the definition of one or two letters in the same sentence at the same time:

$$1 \leq \text{letter} \leq 2 \wedge \text{letter} = \text{description}. \tag{2}$$

This rigorous encoding of constraints is methodically applied to every available intent yielding a robust specification that outlines the allowable entities associated with each specific intent. It is fundamental to highlight the indispensable role played by human experts with profound insights into the intricate interplay between intent and entities during the construction of their normative constraints. These constraints are critical in the post-processing phase, where the output of the NLU component undergoes thorough scrutiny to evaluate the coherence between the extracted intent and entities and the predefined constraints.

In practice, this method generates a dictionary of entity counts, denoted as EC, which maps each entity type to its corresponding occurrences in the set of entities E. It simplifies the evaluation of constraints associated with each intent, as defined by human experts. In joint models combining intent classification and entity extraction, the intent classification output, represented as dictionary $\hat{I}$, assigns a confidence score denoted by s_k to each specific intent i_k satisfying the condition $1 \leq k \leq N$. This confidence score, s_k, provides a likelihood estimate for the given intent. Additionally, the entity extraction output comprises a set of labeled entities, $\hat{E} = \{(e_k, v_k), 1 \leq k \leq l\}$.

Based on this idea, we propose two methods to enhance the joint task of intent classification and entity extraction. The first one is founded on applying constraints developed by human experts to single-model predictions. The second one suggests the use of ensemble models, capitalizing on the complementary strengths and weaknesses inherent in two distinct models. This approach involves applying the defined constraints in the prediction decision to further reinforce the relationship between intent and entities.

4.1.1. Constraint-based approach

Within the constraints of a specified intent condition, instances that violate the condition invoke the prioritization mechanism (Albornoz De Luise et al., 2024). In this mechanism, the system traverses the alternatives based on the ranked list of intents to find the most compatible intent with its corresponding constraints. If none of the intents are compatible with their corresponding constraints, the NLU component searches the list of sorted intents in decreasing order of confidence and selects the first intent that satisfies its corresponding constraints. In case none of the intents satisfy their respective constraints or the score associated with the first matching intent is below a certain threshold, the system selects a fallback intent for the session, indicating the inability to conclusively ascertain an outcome.

Subsequently, the intents in the dictionary $\hat{I}$ are sequentially examined, and organized in a descending order based on their associated scores. The evaluation of the constraint linked to each visited intent involves the utilization of entity count information derived from the entity count dictionary EC. If the constraint is satisfied, the evaluated intent is returned, together with the set $\hat{E}$ of tagged entities, and the process ends. If the aforementioned constraint is not fulfilled, the subsequent intent is addressed, thereby initiating a recurrent cycle. This cycle persists until either all remaining intents have been processed or the score associated with the currently addressed intent descends to a predetermined threshold. In either scenario, a fallback response is issued.

In cases where the initially returned intent from the NLU component aligns harmoniously with the set of entities, this algorithm preserves the original response. Conversely, when such alignment is absent, the algorithm seeks an alternative intent that demonstrates compatibility with the entities that have been extracted. This strategy is designed to preclude the introduction of undesirable consequences and to secure an outcome characterized by non-deterioration. The method systematically seeks to rectify responses that have been conclusively identified as erroneous, thereby adhering to an objective criterion.

4.1.2. Ensemble-based approach

Ensemble learning methods are techniques where multiple compatible learning models are used to perform a single task to get better predictive performance (Zhou, 2012). In this case, the ensemble approach is applied in the inference phase. First, we train two NLU models separately. Divergent predictions within the models trigger a mechanism through which the system assesses whether either of the two predictions aligns with its corresponding criteria.

Algorithm 1 presents a pseudo-code representation of the approach, assuming a function called *Compatible* whose primary purpose is to evaluate whether the intent and entities supplied as inputs satisfy the constraint associated with the particular intent.

In cases where both models produce valid predictions, or only the prioritized model (or the model with the highest overall performance) predicts an intent that adheres to its specific constraints, the system retains the response originating from the model that we establish as a priority. If the model assigned a priority does not meet the criteria, but the secondary model meets a valid constraint, the system retains the prediction of the secondary model. If neither prediction from the two models meets the criteria, the intent determined by the prioritized model is merged with the entity extracted by the other. If this merged prediction meets the criteria, the system corrects the response based on the merged prediction. If this does not happen, the intent determined by the secondary model is merged with the entity extracted by the prioritized model. If this merged prediction meets the criteria, the system corrects the response based on the merged prediction. However, if no such consistency exists, the system returns the prediction of the prioritized model.

Algorithm 1: Ensemble-based Algorithm

Input : I_1: predicted intent of model 1
I_2: predicted intent of model 2
$E_1 = \{(e_k, v_k), 1 \le k \le l\}$: set of l tagged entities of model 1, with labels e_k and values v_k
$E_2 = \{(e_r, v_r), 1 \le r \le s\}$: set of s tagged entities of model 2, with labels e_r and values v_r
Output: Output Intent
Output Entities

```
if Compatible(I1, E1) then
    return I1, E1
else if Compatible(I2, E2) then
    return I2, E2
else if Compatible(I1, E2) then
    return I1, E2
else if Compatible(I2, E1) then
    return I2, E1
else
    return I1, E1
```

The constraint-based approach offers a precise method for fine-tuning the relationships between intents and entities using predefined rules. This is crucial in cases where intent classification and entity extraction are closely intertwined. However, its effectiveness may hinge on the accurate definition of rules and restrictions.

On the other hand, the ensemble-based approach capitalizes on the complementarity of two NLU models to enhance the performance but requires additional resources to implement two models in parallel. The choice between these approaches will be contingent on the specific nature of the application and the requirements of the CA.

4.2. Adapted TF-IDF

Entity–quantity matching task states mapping user inputs to quantities under word problems context. Consider a word problem P that consists of multiple quantities $Q = \{q_1, q_2, \ldots, q_N\}$, where N is the total number of quantities. Each quantity q_i is made up of descriptions $d_i = \{d_{i,1}, d_{i,2}, \ldots, d_{i,M}\}$, where M is the total number of descriptions that convey the same meaning of quantity. When a user gives an input x, the task is to find the description $d_{i,j}$ that is most semantically similar to x and identify the corresponding q_i. Our focus is to develop efficient methods for computing text similarity, with the aim of improving quantity-matching performance.

In the context of TF-IDF, each of the descriptions of each quantity is treated as a separate document. Meanwhile, the corresponding quantities serve as the class labels for text classification. The mathematical definition of the classic TF-IDF (Joachims et al., 1997) is depicted as follows:

$$\text{TF-IDF} = \text{TF} \cdot \text{IDF} \tag{3}$$

The Term Frequency (TF) calculates the word frequency in each document.

$$\text{TF} = \frac{\text{Number of words in a document}}{\text{Total words in the document}} \tag{4}$$

The Inverse Document Frequency (IDF) gives the measure of word importance.

$$\text{IDF} = \frac{\text{Total number of documents}}{\text{Number of documents has word}} \tag{5}$$

The TF-IDF metric indicates how significant or relevant a term is in a document, that identifies words or sets that frequently appear in a single document yet are uncommon throughout the text corpus. However, due to the generally short length of the description, which usually contains only a few words, they often lack the ability to effectively represent the relative importance of each word in the text. To address this problem, we proposed an adapted TF-IDF, which merges multiple descriptions of the same quantity into a single document, thereby enhancing the information's comprehensiveness through the addition of alternative expressions.

Algorithm 2: Adapted TF-IDF Algorithm for a word problem

Input : x: input description of a quantity provided by user
Q: list of descriptions of quantities of a math word problem

Output: Predicted input description's quantity label

```
docs ← []
for quantity in Q do
    doc ← ""
    for description in quantity do
        doc ← doc + description
    docs.append(doc)
docs_features ← TFIDF(docs)
input_features ← TFIDF_Transform(x)
scores_tfidf ← ComputeCosineSimilary(input_features, docs_features)
result ← GetPredictedQuantityCategory(scores_tfidf)
return result
```

The proposed adapted TF-IDF algorithm to choose the most similar description to the one provided by the user is presented as pseudo-code in Algorithm 2. The + operator has been used to denote string concatenation, and *append* represents the addition of a new element to a list. The algorithm proceeds by first combining the descriptions of each quantity as a single document. Then, the *TFIDF()* function learns the vocabulary and the IDF for each term by using the generated documents according to Eqs. (3), (4) and (5). The output of *TFIDF()* is a document-term matrix $docs\_features \in \mathbb{R}^{N \times V}$, where N is the number of documents and V is the vocabulary size. Each row of this matrix represents the textual features of one document. Afterward, *TFIDF_Transform()* is used to transform the input description x to a feature vector $input\_features$, based on the learned vocabulary and the IDF values. Next, *ComputeCosineSimilarity()* is used to compute the cosine similarity between each row in $docs\_features$ and the $input\_features$ vector, by using expression (6).

$$\cos\left(\vec{v}, \vec{w}\right) = \frac{\vec{v} \cdot \vec{w}}{\|\vec{v}\| \cdot \|\vec{w}\|} = \frac{\sum_{i=1}^{n} v_i \times w_i}{\sqrt{\sum_{i=1}^{n} v_i^2} \sqrt{\sum_{i=1}^{n} w_i^2}} \tag{6}$$

where $\vec{v}$ and $\vec{w}$ are document-term vectors. This function returns a vector *scores_tfidf* of size N, where each element represents the similarity between the input text and the corresponding document. Finally, *GetPredictedQuantityCategory()* returns the quantity that corresponds to the highest similarity score.

4.3. Datasets

All experiments concerning the strategies for enhanced intent classification and entity extraction were carried out using the AWPS dataset (Albornoz-De Luise et al., 2023). For entity–quantity matching, we utilized our MATH-HINTS dataset.

The AWPS dataset captures conversations during the problem-solving process within authentic teaching scenarios, where students engage with an ITS through a chat interface to solve algebraic word problems (Albornoz-De Luise et al., 2023). This dataset comprises a total of 25 intents and 5 entities, exhibiting a noticeable intent-entity dependency, denoting robust constraints on the co-occurrence of intents and entities. Following a similar methodology to Albornoz-De Luise et al. (2023), we take the dataset with 6293 utterances and extract 10% of the entries (629 utterances) to use as a validation set, to control the generalization error and decide when to stop training to avoid overfitting. While a separate dataset of 1973 sentences is reserved for evaluating the generated models. Given that the origin of the data in both the training and test sets is attributed to different students, this contributes to a natural data partitioning that enhances the independence between these sets. Consequently, it enables a more impartial and fair assessment of the generated models in a real-world context.

The MATH-HINTS dataset is a proprietary collection of algebraic math word problems. These problems are represented using a hypergraph structure, where links delineate the relationship between quantities. Each quantity has its description and several alternative descriptions are presented to enrich quantity representation. This dataset provides a total of 219 problems with 8559 quantity descriptions, and 47 807 words. The vocabulary was composed of 927 words.

4.4. Construction of NLU pipelines

To evaluate the performance of the proposed approaches aimed at enhancing intent classification and entity extraction, first, we examine different NLU pipelines for training various NLU models. To train the required NLU models, we used the open-source Rasa conversational framework (Bocklisch et al., 2017), specifically the Rasa NLU tool. This tool offers a larger degree of freedom and flexibility for configuring NLU pipelines, by providing a wide range of components including tokenization, feature extraction, intent classification, and entity extraction. The selection and arrangement of these components within a configuration file define the NLU pipeline, executed sequentially during both the training and prediction phases of an NLU model.

Rasa incorporates the Dual Intent and Entity Transformer (DIET) (Bunk et al., 2020) multi-task architecture for intent classification and entity extraction. The suggested configuration selects the DIET component as the default. One notable feature of DIET is its seamless integration of pre-trained word embeddings from language models, such as BERT, GloVe, or ConveRT, combined with sparse word and character-level n-gram features, a process facilitated by the specified feature extraction components within the NLU pipeline.

The DIET architecture, based on a transformer, leverages a shared transformer output to fulfill both intent classification and entity extraction tasks. To predict intent, the model used the transformer output for the sentence-level token. Simultaneously, the same transformer output sequence undergoes processing through a CRF (Lafferty et al., 2001) to predict entity labels. To capitalize on the inherent interdependencies between intents and entities, a unified loss function that jointly considers both sets of predictions is employed.

The DIET architecture's performance as an NLU component has been previously evaluated on the AWPS dataset, consistently demonstrating competitive performance. We maintained the pipeline configuration and hyperparameter settings that had yielded the best results reported in a previous evaluation of the AWPS dataset (Albornoz-De Luise et al., 2023), along with Rasa version 3.6.4.

The best results were achieved by combining the spaCy tokenizer with a different set of sparse features with word embeddings generated by spaCy.[5] These sparse features encompass aggregated lexical attributes, token-level one-hot encodings, multi-hot encodings with $2 \leq n \leq 4$, and features related to the presence of automatically extracted relevant regular expressions. In our experiments, we denote this approach as SS.

Additionally, the semantic representation of the inputs is enhanced by incorporating pre-trained embeddings like BERT and DistilBERT. This process entails replacing only the spaCy embedding with these advanced models. In the experiments, we designate these pipelines as SB and SD, respectively. Furthermore, when the spaCy tokenizer is replaced with the White tokenizer in these pipelines, we refer to them as WB, and WD, respectively. Moreover, where the spaCy tokenizer is replaced with the White tokenizer while maintaining the original DIET hyperparameters and only using sparse feature settings, this approach is denoted as W in our experiments.

To illustrate the configurational flexibility provided by the Rasa NLU tool, Fig. 3 presents a unified graphical representation of the different NLU pipelines employed in our experiments. The diagram specifically illustrates the choice of a tokenizer (either White or spaCy[6]) and the application of a dense features extraction method such as spaCy, BERT (Devlin et al., 2018), or DistilBERT (Sanh et al., 2020), which depend on the tokenizer chosen. Additionally, all pipelines include Lexico-Syntactic Features and CountVectors Features (Words and N-grams), creating a robust input to the DIET Classifier. Table 2 complements this visual representation by summarizing the specific features used for each pipeline. Table 3 provides a brief description of each component discussed in this study.

[5] https://spacy.io/api.

[6] https://spacy.io/api/tokenizer.

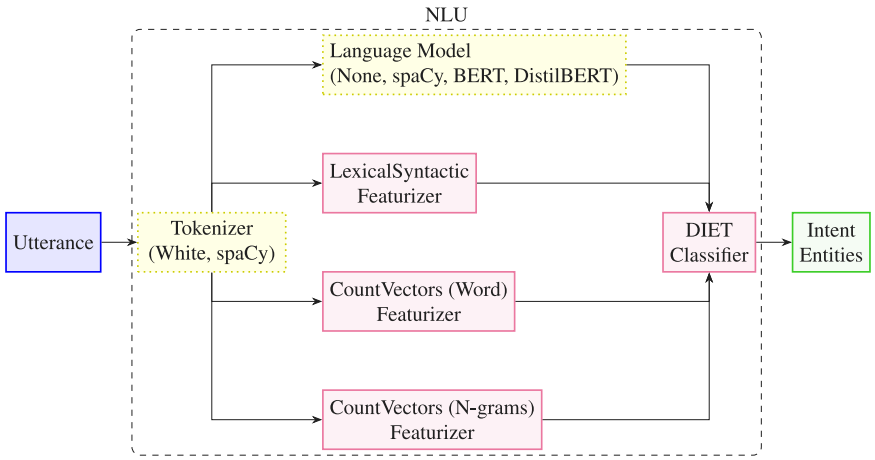


Fig. 3. A unified graphical representation of the diverse NLU pipelines used. Yellow boxes with dashed-line borders indicate the selection of only one option from those presented in brackets. It is worth noting that the 'None' language model is exclusive to the white tokenizer, while 'spaCy' requires spaCyTokenizer, while 'BERT' and 'DistilBERT' are compatible with either the white or spaCy tokenizer. Pink boxes have been used to specify that these components remain fixed across all pipeline configurations. The blue box denotes the input and the green box the output, showcasing the intended results after processing through the NLU pipeline.

Table 2
Summary of main features of all pipelines evaluated.

Pipeline name	Tokenizer		Sparse features		Dense features		
	WhiteTokenizer	spaCyTokenizer	Word Counts + N-grams	Lexical and Syntactic	spaCy	BERT	DistilBERT
W	✓	✗	✓	✓	✗	✗	✗
WB	✓	✗	✓	✓	✗	✓	✗
WD	✓	✗	✓	✓	✗	✗	✓
SS	✗	✓	✓	✓	✓	✗	✗
SB	✗	✓	✓	✓	✗	✓	✗
SD	✗	✓	✓	✓	✗	✗	✓

Table 3
A brief description of each component in NLU pipelines.

Component	Description
WhiteTokenizer	Splits text into tokens, creating a token for every whitespace-separated character sequence, while also substituting certain characters with whitespace based on specific conditions.
spaCyTokenizer	Create tokens using spaCy open-source library (Honnibal and Montani, 2017).
Word Counts + N-grams	Creates a bag-of-words representation (Qader et al., 2019). The first featurizer counts whole words, while the second examines sub-word character sequences.
Lexical and Syntactic	Generates sparse lexical and syntactic features for each token, tagging them with details like sentence position, digit presence, title status, part-of-speech, etc.
spaCy	Extracts dense vector representations of utterances using the spaCy language model.[a]
BERT	Leverages a pre-trained BERT model (Devlin et al., 2018) to compute dense features.
DistilBERT	Utilizes a pre-trained DistilBERT model (Sanh et al., 2020) to create dense vector representations.

[a] https://spacy.io/usage/models.

4.5. Experimental procedure

All our experiments were conducted on a computer equipped with a 13th-generation i7 processor with 128 RAM and a single NVIDIA RTX 3090 GPU with 24 GB of memory, running Ubuntu 20.04.4 LTS. The required model implementations used Python 3.9 with version 2.0 of the open-source Pytorch.

To assess the impact and advantages introduced by the proposed approaches for enhancing the performance of joint intent classification and entity extraction in a realistic environment, a series of experiments were performed on the AWPS dataset, discussed in Section 4.3.

For each NLU pipeline that was developed, a corresponding model was trained using the training and validation datasets from the AWPS dataset. These pipelines encompass W, WB, WD, SS, SB, and SD. Subsequently, the impact of the proposed methods was evaluated by analyzing their performance on the test set.

To evaluate the effectiveness of the proposed adapted TD-IDF for entity–quantity matching, we conducted experiments on the MATH-HINTS dataset employing leave-one-out problem-exclusive validation. Specifically, for each problem file, we used one quantity description as the user entity, and the remainder were grouped by quantity. Descriptions of the same quantity were concatenated in the same document. Later, these documents were fed into adapted TF-IDF to obtain the predicted quantity category. A user entity was considered correct if it exhibits that the predicted quantity coincides with its true quantity category. For comparison, we chose two well-performed pre-trained language models in sentence similarity to realize the same problem: SBERT (Reimers and Gurevych, 2020) and SGPT (Muennighoff, 2022).

SBERT (Reimers and Gurevych, 2020), introduced by Reimers and Gurevych, addresses text similarity tasks. In this context, the model assesses the degree of similarity between a pair of sentences. SBERT's architecture incorporates a pooling operation applied to BERT's output, yielding a fixed-size sentence embedding. During fine-tuning, SBERT employs both a siamese network and a triple network for enhanced training, facilitating more effective learning of semantic information within the sentences.

SGPT (Muennighoff, 2022), introduced by Muennighoff, employed a decoder-based transformer to address sentence embeddings and semantic search tasks through prompting and fine-tuning. Similar to the SBERT setting, supervised contrastive learning with in-batch negatives was performed. In SGPT's architecture, a key distinction arises from the inclusion of a causal attention mask within an auto-regressive decoder transformer, causing tokens not to attend to future tokens as in an encoder transformer. Consequently, only the last token in a sequence attends to all preceding tokens. To rectify this information mismatch, they used a position-weighted mean pooling method, assigning greater weight to later tokens in the sequence.

In the evaluation of extracting semantic representations of descriptions using a pre-trained language model, it is used directly to extract sentence embeddings without any training process. Because in the context of word problems, the sentences described are short in length, the training model is prone to overfitting. We first used the pre-trained model to compute the embeddings of all descriptions, and then compute the cosine similarity between the input description and the remaining quantity descriptions in the problem. Of these, the quantity description with the highest similarity is the prediction quantity category. For SBERT, we chose the top five pre-trained models in the order of the performance of sentence embeddings in the 14 datasets. As for SGPT, we tested all available pre-trained models with a model size of 125 million parameters. The 1.3 billion parameters version of the best-performing 125 million parameters model was also evaluated to assess the potential improvement achieved by using a larger model. Our code was based on Huggingface[7] implementation.

4.6. Performance metrics

For intent classification, intent accuracy is the commonly used evaluation metric. It is defined as the ratio of the number of correct predictions of intent to the total number of sentences (Weld et al., 2022). Concerning entity extraction, the performance is measured with F1 score (Tjong Kim Sang and De Meulder, 2003):

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

In this formula, precision is the ratio of entities found by the entity extraction system that are correct. Recall is the ratio of entities present in the corpus that are found by the system. An entity is considered correct only if it is an exact match of the corresponding entity in the data file.

To assess the overall performance of an NLU system and offer a comprehensive evaluation in a single metric, semantic accuracy is commonly employed (Goo et al., 2018; Chen et al., 2019). Semantic accuracy is defined as the ratio of correctly analyzed utterances to the total number of utterances. In this context, an utterance is considered correctly processed only if both its intent and all associated entities are predicted correctly.

About entity–quantity matching, we use accuracy as an evaluation metric. Entity–quantity matching is similar to a multi-class classification task, where the number of classes varies for each word problem. Accuracy provides a global performance measure evaluation because it calculates the ratio of correctly matched entity–quantity samples divided by all matching samples.

[7] https://huggingface.co/.

5. Results

5.1. Results of NLU performance

The comparative analysis of the overall performance of the NLU component with and without the proposed methods has been presented in Table 4. It also offers a comprehensive overview of the performance metrics for every presented approach on the AWPS dataset. The results for every single model are provided under the column "Baseline Model".

The "Constraint-based method" column presents the outcomes of applying the Constraint-based approach to the "models" column, where each model has been trained according to the configurations specified in the corresponding row on the left. The results for the ensemble approach are located on the right-hand side of the table, in the "Ensemble-based Method 1 - Method 2" column. Results specified in row i column j correspond to those obtained when selecting the model configurations in row i and column j when giving priority to the model in row i.

It is clear from the "Baseline Model" column that the results vary depending on the NLU pipeline configuration used. However, all of them indicate that there is still room for improvement, despite the fact that the reference method jointly predicts intents and entities, thus already taking into account the existing relationships between them. Based on the results presented in Table 4, the individual model with the highest scores is the one trained with the SD configuration, achieving an intent accuracy of 97.11%, an entity F1 score of 96.79%, and a semantic accuracy of 93.06%. The second-best performing model is associated with the SB configuration, attaining an intent accuracy of 96.20%, an entity F1 score of 96.59%, and a semantic accuracy of 91.99%. It is evident that employing the dense features provided by the BERT and distilBERT language models, in conjunction with the spaCy tokenizer, leads to improvement.

5.1.1. Constraint-based approach

As seen in the results presented in the column "Constraint-based method" in the aforementioned table, the constraint-based approach consistently enhances semantic accuracy, at the cost of a reduction in intent accuracy, underscoring the delicate balance between intent classification and entity extraction, while still maintaining the entity F1 score. This method only exerts an effect when the intent and entities identified by the NLU component are not compatible. Furthermore, it solely acts based on the intent, and therefore, by its nature, it cannot rectify any errors in entities. This implies that a potential correction is only possible when the intent and entities are incompatible, the entities have been successfully identified, and the intent has been misclassified. Consequently, the intent accuracy is reduced if the entity count does not correspond to any intent, resulting in the system returning fallback intent.

Table 4 reveals that the constraint-based method's effects on configurations (W, WB, WD, SS, SB, SD) differ, particularly in intent and semantic accuracies. Compared with the "Baseline Model", the W configuration suffers the largest drop in intent accuracy, falling by 36.34%, yet it benefits from a 1.4% increase in semantic accuracy. The WB configuration stands out for the greatest boost in semantic accuracy, at 1.83%, despite experiencing a 34.11% fall in intent accuracy. On the other hand, the SB configuration shows a minimal decline in intent accuracy, at just 1.22%, and sees a rise in semantic accuracy by 0.81%. This nuanced performance across configurations illustrates the complex trade-offs involved in employing the constraint-based method, highlighting its potential to enhance semantic understanding while underscoring the challenges in maintaining high intent accuracy.

5.1.2. Ensemble-based approach

The ensemble-based approach improves intent accuracy, entity F1 score, and the semantic accuracy of the prioritized model, surpassing the metrics of both the individual prioritized model and the constraint-based approach, supporting the effectiveness of this dual approach in improving overall system understanding.

The aim of this ensemble approach involves the integration of two models, with a preference given to the superior-performing one. This preference is evident in the lower triangular matrix in Table 4, which contains metrics derived from the ensemble model, where the model in the corresponding row (demonstrating higher performance) is paired with the model in the corresponding column, achieving improved scores in both the intent accuracy, F1 entity score, and semantic accuracy.

In particular, the model employing SD configuration demonstrates that the semantic accuracy of this model can be further improved by incorporating any of the other trained models. However, the model that best fits is the one employing the SS configuration, used in previous works (Albornoz-De Luise et al., 2023; Arnau-González et al., 2023). This ensemble model refers to the combination that incorporates both the SD and SS models, which achieves an intent accuracy of 97.21%, an entity F1 score of 97.45%, and the highest semantic accuracy of 94.37%. It is worth noting that the intent accuracy experiences a slight reduction only in the ensemble model with SD and WD models achieving an intent accuracy of 97.06% (this value is indicated in red in Table 4).

5.2. Results of adapted TF-IDF

Table 5 shows the results of the entity–quantity matching task using the pre-trained model approach, the traditional TF-IDF approach, and the proposed adapted TF-IDF approach. Since no language model was trained, for a better evaluation, we used several pre-trained models from different large corpora for the evaluation. The best-performing model in the SGPT-based approach is SGPT-125M-mean-nli with an accuracy of 91.8%. Furthermore, using the same training data but increasing the model size did not improve the performance. On the contrary, the performance of SGPT-1.3B-mean-nli is 1.6% lower than that of SGPT-125M-mean-nli. Among the SBERT-based methods, the best performance is SBERT-420M-multi-qa-mpnet-base-dot-v1 with 91.6%. Compared to the best-performing SGPT model, it has a 0.2% decrease in performance, while its model parameters are three times larger.

Table 4
Performance comparison of the different approaches on the AWPS dataset. Intent accuracy is represented in blue cells, entity F1 scores are shown in green cells, and semantic accuracy is represented in gray cells. The best results for intent accuracy, entity F1 score, and semantic accuracy for each approach (Baseline Model, Constraint-based Model, and Ensemble-based) are highlighted in bold, while the second-best scores are underlined. The red value denotes a reduction in the respective metric compared to the baseline model.

Model	Baseline Model	Constraint-based Method
W	86.11	49.77
	67.98	67.98
	48.30	49.77
WB	87.58	53.47
	69.04	69.04
	49.21	51.04
WD	88.14	53.17
	68.87	68.87
	50.03	50.94
SS	94.88	92.85
	93.28	93.28
	89.15	89.66
SB	96.20	**94.98**
	96.59	96.59
	91.99	92.80
SD	**97.11**	94.83
	96.79	**96.79**
	93.06	**93.11**

Ensemble-based Model 1 - Model 2

Model 1 \ Model 2	W	WB	WD	SS	SB	SD
W		87.58	87.73	94.37	94.53	94.98
		68.70	68.84	95.23	95.79	96.13
		50.48	50.79	91.38	91.69	92.34
WB	88.85		89.10	95.34	95.74	96.05
	69.30		69.37	95.63	96.23	96.56
	51.14		51.39	91.59	92.09	92.85
WD	88.80	88.69		95.34	95.84	95.74
	69.04	69.14		95.76	95.96	96.23
	51.14	51.09		91.99	92.19	92.50
SS	95.34	95.59	95.39		95.74	95.84
	95.00	95.26	95.13		95.43	95.43
	91.38	91.64	91.64		91.94	91.99
SB	96.30	96.65	96.55	96.76		97.01
	96.89	97.15	96.95	97.09		97.38
	93.01	93.26	93.31	93.82		94.02
SD	97.21	**97.42**	97.06	97.21	97.31	
	97.48	97.42	97.18	97.45	97.32	
	94.17	94.22	94.07	**94.37**	94.02	

Intent Acc | Entity F1 | Sem Acc

Table 5
Accuracy (in %) for entity–quantity matching in MATH-HINTS.

Pre-trained model	Accuracy (%)
SGPT-125M-mean-nli-linear5	67.8
SGPT-125M-lasttoken-nli	69.2
SGPT-125M-lasttoken-msmarco-specb	71.5
SGPT-125M-weightedmean-msmarco-specb-bitfit	76.4
SGPT-125M-weightedmean-nli-bitfit-linearthenpool1-noact	80.3
SGPT-125M-weightedmean-msmarco-specb-bitfitwte	81.8
SGPT-125M-weightedmean-nli-bitfit	83.6
SGPT-125M-weightedmean-nli	85.6
SGPT-125M-mean-nli-bitfit	89.7
SGPT-125M-scratchmean-nli	91.5
SGPT-125M-mean-nli	91.8
SGPT-1.3B-mean-nli	90.2
SBERT-420M-all-mpnet-base-v2	88.4
SBERT-120M-all-MiniLM-L12-v2	89.0
SBERT-290M-all-distilroberta-v1	89.3
SBERT-250M-multi-qa-distilbert-cos-v1	91.0
SBERT-420M-multi-qa-mpnet-base-dot-v1	91.6
TF-IDF	78.4
Adapted TF-IDF	**95.1**

In the TF-IDF-based approach, the accuracy is 78.4% using the conventional TF-IDF method, which uses only a single document representation. The multi-document representation using adapted TF-IDF shows a relative performance improvement of 16.7% over the conventional TF-IDF. In addition, the proposed adapted TF-IDF method obtains the highest accuracy of 95.1% among all the validation methods, which is 3.3% better than the best SGPT method and 3.5% better than the best SBERT method.

Overall, our method was the one that obtained the best performance although the pre-training-based language models also show good results. In terms of model complexity, using the proposed adapted model is simpler and more effective. Moreover, in the case of word problem-solving, the amount of data is not very large, so our approach can be effectively applied to text-based dialogues related to mathematical problem-solving.

5.3. Discussion

In summary, the proposed approaches provide valuable enhancements that can be seamlessly integrated into conversational intelligent tutoring systems to strengthen both the NLU and NLP components. While our experiments provide valuable insights into the effectiveness of the proposed methods, several limitations need to be considered. The major limitation of this study is that the proposed methods were only tested on the MATH-HINTS dataset. This was partly due to the unavailability of suitable databases for testing entity–quantity matching approach. Another limitation is the decrease in intent classification accuracy when applying the constraint-based to each model used. This happens because an error in entity extraction can lead to the incorrect modification of an otherwise correct intent. In nearly all instances, the ensemble method avoided the typical decrease in intent accuracy observed with the fixed model under the constraint-based approach. There was just one scenario where employing the ensemble approach led to a decline in the intent accuracy score of the original model. In every other situation, it improved all metrics for the prioritized model, as well as the metrics when the constraint-based method was used.

Creating manual constraints to determine the permissible number of entities for each intent raises scalability concerns. Such constraints may grow increasingly complex with datasets containing a broad and detailed assortment of intents and entities. In our situation, the modest and well-defined group of intents with entities in our dataset allowed for swift rule creation. This emphasizes the importance of thorough dataset knowledge for the successful application of constraint-based methods. The additional limitation is that the use of an adapted TF-IDF approach presupposed the need for the existence of multiple descriptions that differ for the same quantity. Its performance depends heavily on the quality of the descriptions. It should be emphasized that this method still fails to solve the problem of out-of-vocabulary (OOV), unknown words that appear in the testing conversation but not in the recognition vocabulary (Yang et al., 2018). This is because the method does not contain semantic-level features. However, since each match is context-specific, the vocabulary range is limited to a certain extent, which makes the occurrence of OOV smaller. Even more, if the system fails to recognize the user entity, it can ask the student to provide additional clarification, thereby ensuring a more accurate understanding.

6. Conclusions

In this work, we have introduced two improvements aimed at enhancing the performance of a hypergraph-based CITS. First, we proposed and compared the adoption of both a constraint-based approach and an ensemble-based approach to enhance joint models for intent classification and entity extraction. These techniques led to a substantial boost in overall performance metrics. Second, we advocate an adapted TF-IDF approach to facilitate the mapping process between a particular type of entity extracted and predefined descriptions that refer to the quantities involved in a problem solution.

We present two approaches based on predefined constraints that establish the permissible quantity of associated entities for a given intent. These approaches aim to identify and correct inconsistent predictions. The constraint-based approach involves the exclusive application of these constraints to the output of a single NLU model. On the other hand, the ensemble-based approach requires predictions from two NLU models. In cases where predictions diverge, predefined constraints are applied in an attempt to rectify them.

The evaluation of our proposed approaches involved using the AWPS dataset, and we employed the DIET architecture within the RASA framework. Experimental results demonstrate that the proposed framework significantly improves performance in both tasks, thereby boosting overall performance in NLU. This work demonstrates that the relationship can be more effectively modeled when additional information is incorporated by human experts. By doing so, the interdependency modeling can extend beyond what can be directly observed from the training data, leading to further performance improvements. In our approaches, both the design of propositional logic for constraint modeling and for the ensemble method significantly contribute to enhancing semantic accuracy. Additionally, the ensemble model also improves intent accuracy and F1 entity score.

In the entity–quantity matching task, we proposed an adapted TF-IDF approach based on the changes made in the traditional TF-IDF. In this method, each description was considered as a document, and the descriptions under the same quantity were merged into the same document to enable better computation of the importance between words, thus optimizing the performance of computing the similarity between documents for the matching task. The proposed method was evaluated on the MATH-HINTS dataset and compared with popular deep learning models for computing text similarity, SGPT, and SBERT. As a result, the proposed method achieved the best performance with an accuracy of 95.1%, which was improved by almost 3% compared to the pre-trained language model. In addition, the use of statistical-based methods requires less memory than neural network-based methods, which is simpler and more efficient, and it also limits the context in which the similarity was calculated.

In future work, we aim to further refine and extend the constraint-based and ensemble-based approaches. We also plan to test the robustness and generalizability of our proposed method using a wider range of datasets and different NLU architectures. To address the out-of-vocabulary problem in entity-matching, we intend to explore large-scale language generation models' capability to produce different expressions for synonymous descriptions. Finally, while the current work focused on mathematical language problem-solving, we seek to investigate the adaptability of hypergraph-based CITS to other educational disciplines or professional domains.

CRediT authorship contribution statement

Yuyan Wu: Data curation, Formal analysis, Investigation, Methodology, Project administration, Software, Validation, Visualization, Writing – original draft, Conceptualization, Writing – review & editing. **Romina Soledad Albornoz-De Luise:** Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Conceptualization, Writing – review & editing, Project administration. **Miguel Arevalillo-Herráez:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Visualization, Writing – review & editing.

Declaration of competing 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.

Data availability

Data will be made available on request.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT and Grammarly to improve language and readability. After using these services, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Acknowledgments

This research has been supported by project TED2021-129485B-C42, funded by MCIN/AEI/10.13039/501100011033 and the European Union "NextGenerationEU"/PRTR; grant PRE2019-090854, funded by MCIN/AEI/10.13039/501100011033 and "ESF Investing in your future"; grant ACIF/2021/439 funded by the Conselleria de Educación, Universidades y Empleo de la Generalitat Valenciana; and grant PID2023-150960NB-I00 funded by the Spanish Ministry of Science, Innovation and Universities and the European Union.

References

Abro, W.A., Qi, G., Aamir, M., Ali, Z., 2022. Joint intent detection and slot filling using weighted finite state transducer and BERT. Appl. Intell. 52, 17356–17370. http://dx.doi.org/10.1007/s10489-022-03295-9.

Abro, W.A., Qi, G., Ali, Z., Feng, Y., Aamir, M., 2020. Multi-turn intent determination and slot filling with neural networks and regular expressions. Knowl.-Based Syst. 208, 106428. http://dx.doi.org/10.1016/j.knosys.2020.106428.

Ahn, J., Kasireddy, V., Mukhi, N., et al., 2019. Interactive learning in a conversational intelligent tutoring system using student feedback, concept grouping and text linking. In: INTED2019 Proceedings. IATED, pp. 2820–2830.

Albornoz-De Luise, R.S., Arevalillo-Herráez, M., Arnau, D., 2023. On using conversational frameworks to support natural language interaction in intelligent tutoring systems. IEEE Trans. Learn. Technol. 16 (5), 722–735. http://dx.doi.org/10.1109/TLT.2023.3245121.

Albornoz De Luise, R., Arevalillo-Herráez, M., Wu, Y., 2024. Leveraging intent–entity relationships to enhance semantic accuracy in nlu models. Neural Comput. Appl. http://dx.doi.org/10.1007/s00521-024-09927-0.

Albornoz-De Luise, R.S., Arnau-González, P., Arevalillo-Herráez, M., 2022a. On providing natural language support for intelligent tutoring systems. In: Rodrigo, M.M., Matsuda, N., Cristea, A.I., Dimitrova, V. (Eds.), Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners' and Doctoral Consortium. Springer International Publishing, Cham, pp. 564–568. http://dx.doi.org/10.1007/978-3-031-11647-6_116.

Albornoz-De Luise, R.S., Arnau-González, P., Arevalillo-Herráez, M., 2022b. Conversational agent design for algebra tutoring. In: 2022 IEEE International Conference on Systems, Man, and Cybernetics. SMC, pp. 604–609. http://dx.doi.org/10.1109/SMC53654.2022.9945524.

Arevalillo-Herraez, M., Arnau, D., Marco-Giménez, L., 2013. Domain-specific knowledge representation and inference engine for an intelligent tutoring system. Knowl.-Based Syst. 49, 97–105. http://dx.doi.org/10.1016/j.knosys.2013.04.017.

Arevalillo-Herráez, M., Marco-Giménez, L., Arnau, D., González-Calero, J.A., 2017. Adding sensor-free intention-based affective support to an intelligent tutoring system. Knowl.-Based Syst. 132, 85–93. http://dx.doi.org/10.1016/j.knosys.2017.06.024.

Arnau, D., Arevalillo-Herráez, M., González-Calero, J.A., 2014. Emulating human supervision in an intelligent tutoring system for arithmetical problem solving. IEEE Trans. Learn. Technol. 7 (2), 155–164. http://dx.doi.org/10.1109/TLT.2014.2307306.

Arnau, D., Arevalillo-Herráez, M., Puig, L., González-Calero, J.A., 2013. Fundamentals of the design and the operation of an intelligent tutoring system for the learning of the arithmetical and algebraic way of solving word problems. Comput. Educ. 63, 119–130. http://dx.doi.org/10.1016/j.compedu.2012.11.020.

Arnau-González, P., Arevalillo-Herráez, M., Albornoz-De Luise, R., Arnau, D., 2023. A methodological approach to enable natural language interaction in an intelligent tutoring system. Comput. Speech Lang. 81, 101516. http://dx.doi.org/10.1016/j.csl.2023.101516.

Bhargava, A., Celikyilmaz, A., Hakkani-Tür, D., Sarikaya, R., 2013. Easy contextual intent prediction and slot detection. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 8337–8341. http://dx.doi.org/10.1109/ICASSP.2013.6639291.

Bocklisch, T., Faulkner, J., Pawlowski, N., Nichol, A., 2017. Rasa: Open source language understanding and dialogue management. arXiv:1712.05181.

Brajković, E., Vasić, D., 2017. Tree and word embedding based sentence similarity for evaluation of good answers in intelligent tutoring system. In: 2017 25th International Conference on Software, Telecommunications and Computer Networks. SoftCOM, pp. 1–5. http://dx.doi.org/10.23919/SOFTCOM.2017.8115592.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al., 2020. Language models are few-shot learners. Adv. Neural Inf. Process. Syst. 33, 1877–1901.

Bunk, T., Varshneya, D., Vlasov, V., Nichol, A., 2020. DIET: lightweight language understanding for dialogue systems. arXiv:2004.09936.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al., 2018. Universal sentence encoder. arXiv:1803.11175.

Chen, Q., Zhuo, Z., Wang, W., 2019. Bert for joint intent classification and slot filling. arXiv:1902.10909.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A., 2017. Supervised learning of universal sentence representations from natural language inference data. arXiv:1705.02364.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805.

Dowlagar, S., Mamidi, R., 2023. A code-mixed task-oriented dialog dataset for medical domain. Comput. Speech Lang. 78, 101449. http://dx.doi.org/10.1016/j.csl.2022.101449.

Emil Brajković, Daniel Vasić, T.V., 2018. Evaluation of methods for sentence similarity for use in intelligent tutoring system. Adv. Sci. Technol. Eng. Syst. J. 3 (5), 1–5. http://dx.doi.org/10.25046/aj030501.

González-Castro, N., Muñoz-Merino, P.J., Alario-Hoyos, C., Delgado Kloos, C., 2021. Adaptive learning module for a conversational agent to support MOOC learners. Aust. J. Educ. Technol. 37 (2), 24–44. http://dx.doi.org/10.14742/ajet.6646.

Goo, C.-W., Gao, G., Hsu, Y.-K., Huo, C.-L., Chen, T.-C., Hsu, K.-W., Chen, Y.-N., 2018. Slot-gated modeling for joint slot filling and intent prediction. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). pp. 753–757.

Hendrycks, D., Gimpel, K., 2018. A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv:1610.02136.

Honnibal, M., Montani, I., 2017. spaCy 2: Natural language understanding with bloom embeddings, convolutional neural networks, and incremental parsing. To appear. 7 (1), 411–420.

Joachims, T., et al., 1997. A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization. In: ICML. Vol. 97, Citeseer, pp. 143–151.

Jose, K.J., Lakshmi, K.S., 2018. Joint slot filling and intent prediction for natural language understanding in frames dataset. In: 2018 International Conference on Inventive Research in Computing Applications. ICIRCA, pp. 179–181. http://dx.doi.org/10.1109/ICIRCA.2018.8597328.

Joulin, A., Grave, E., Bojanowski, P., Mikolov, T., 2016. Bag of tricks for efficient text classification. arXiv:1607.01759.

Jung, S., 2019. Semantic vector learning for natural language understanding. Comput. Speech Lang. 56, 130–145. http://dx.doi.org/10.1016/j.csl.2018.12.008.

Kane, B., Rossi, F., Guinaudeau, O., Chiesa, V., Quénel, I., Chau, S., 2020. Joint intent detection and slot filling via CNN-LSTM-CRF. In: 2020 6th IEEE Congress on Information Science and Technology. CiSt, pp. 342–347. http://dx.doi.org/10.1109/CiSt49399.2021.9357183.

Katchapakirin, K., Anutariya, C., 2018. An architectural design of ScratchThAI: A conversational agent for computational thinking development using scratch. In: Proceedings of the 10th International Conference on Advances in Information Technology. In: IAIT 2018, Association for Computing Machinery, New York, NY, USA, http://dx.doi.org/10.1145/3291280.3291787.

Kurata, G., Xiang, B., Zhou, B., Yu, M., 2016. Leveraging sentence-level information with encoder lstm for semantic slot filling. arXiv:1601.01530.

Lafferty, J.D., McCallum, A., Pereira, F.C.N., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning. ICML '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 282–289.

Latham, A., 2022. Conversational intelligent tutoring systems: The state of the art. In: Women in Computational Intelligence: Key Advances and Perspectives on Emerging Topics. Springer, pp. 77–101.

Li, X., Chen, Y.-N., Li, L., Gao, J., Celikyilmaz, A., 2017. End-to-end task-completion neural dialogue systems. arXiv:1703.01008.

Li, X., Roth, D., 2006. Learning question classifiers: the role of semantic information. Nat. Lang. Eng. 12 (3), 229–249.

Lim, J., Son, S., Lee, S., Chun, C., Park, S., Hur, Y., Lim, H., 2022. Intent classification and slot filling model for in-vehicle services in Korean. Appl. Sci. 12 (23), http://dx.doi.org/10.3390/app122312438.

Liu, B., Lane, I., 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In: Interspeech 2016. pp. 685–689. http://dx.doi.org/10.21437/Interspeech.2016-1352.
Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv:1301.3781.
Muennighoff, N., 2022. Sgpt: Gpt sentence embeddings for semantic search. arXiv:2202.08904.
Ni, P., Li, Y., Li, G., Chang, V., 2020. Natural language understanding approaches based on joint task of intent detection and slot filling for IoT voice interaction. Neural Comput. Appl. 32, 16149–16166. http://dx.doi.org/10.1007/s00521-020-04805-x.
Ni, J., Young, T., Pandelea, V., Xue, F., Cambria, E., 2023. Recent advances in deep learning based dialogue systems: A systematic survey. Artif. Intell. Rev. 56 (4), 3055–3155.
Nye, B.D., Graesser, A.C., Hu, X., 2014. AutoTutor and family: A review of 17 years of natural language tutoring. Int. J. Artif. Intell. Educ. 24, 427–469.
O'Shea, J., Bandar, Z., Crockett, K., 2011. Systems engineering and conversational agents. In: Tolk, A., Jain, L.C. (Eds.), Intelligence-Based Systems Engineering. Springer Berlin Heidelberg, Berlin, Heidelberg, http://dx.doi.org/10.1007/978-3-642-17931-0_8.
Paladines, J., Ramirez, J., 2020. A systematic literature review of intelligent tutoring systems with dialogue in natural language. IEEE Access 8, 164246–164267. http://dx.doi.org/10.1109/ACCESS.2020.3021383.
Paladines, J., Ramírez, J., Berrocal-Lobo, M., 2021. Integrating a dialog system with an intelligent tutoring system for a 3D virtual laboratory. Interact. Learn. Environ. 1–14. http://dx.doi.org/10.1080/10494820.2021.1972012.
Patwardhan, N., Marrone, S., Sansone, C., 2023. Transformers in the real world: A survey on NLP applications. Information 14 (4), 242.
Pennington, J., Socher, R., Manning, C.D., 2014. Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. EMNLP, pp. 1532–1543.
Qader, W.A., Ameen, M.M., Ahmed, B.I., 2019. An overview of bag of words;importance, implementation, applications, and challenges. In: 2019 International Engineering Conference. IEC, pp. 200–204. http://dx.doi.org/10.1109/IEC47844.2019.8950616.
Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al., 2018. Improving language understanding by generative pre-training.
Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al., 2019. Language models are unsupervised multitask learners. OpenAI Blog 1 (8), 9.
Rafiepour, M., Sartakhti, J.S., 2023. CTRAN: CNN-transformer-based network for natural language understanding. Eng. Appl. Artif. Intell. 126, 107013. http://dx.doi.org/10.1016/j.engappai.2023.107013.
Ramshaw, L.A., Marcus, M.P., 1999. Text chunking using transformation-based learning. In: Natural Language Processing using Very Large Corpora. Springer, pp. 157–176.
Ravuri, S.V., Stolcke, A., 2015. Recurrent neural network and LSTM models for lexical utterance classification. In: Interspeech.
Reimers, N., Gurevych, I., 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. EMNLP, pp. 4512–4525.
Rozga, S., 2018. Chat bot natural language understanding. In: Practical Bot Development: Designing and Building Bots with Node.Js and Microsoft Bot Framework. A Press, Berkeley, CA, pp. 29–46. http://dx.doi.org/10.1007/978-1-4842-3540-9_2.
Rus, V., D'Mello, S., Hu, X., Graesser, A.C., 2013. Recent advances in conversational intelligent tutoring systems. AI Mag. 34 (3), 42–54. http://dx.doi.org/10.1609/aimag.v34i3.2485.
Sanh, V., Debut, L., Chaumond, J., Wolf, T., 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108.
Tjong Kim Sang, E.F., De Meulder, F., 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4. CONLL '03, Association for Computational Linguistics, USA, pp. 142–147. http://dx.doi.org/10.3115/1119176.1119195.
Topal, A.D., Eren, C.D., Gecer, A.K., 2021. Chatbot application in a 5th grade science course. Educ. Inf. Technol. 26 (5), 6241–6265. http://dx.doi.org/10.1007/s10639-021-10627-8.
VanLehn, K., Freedman, R., Jordan, P., Murray, C., Osan, R., Ringenberg, M., Rosé, C., Schulze, K., Shelby, R., Treacy, D., et al., 2000. Fading and deepening: The next steps for andes and other model-tracing tutors. In: Intelligent Tutoring Systems: 5th International Conference, ITS 2000 Montréal, Canada, June 19–23, 2000 Proceedings 5. Springer, pp. 474–483.
Varshney, N., Mishra, S., Baral, C., 2022. Towards improving selective prediction ability of NLP systems. arXiv:2008.09371.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.
Verschaffel, L., Schukajlow, S., Star, J., Van Dooren, W., 2020. Word problems in mathematics education: a survey. ZDM 52 (1), 1–16. http://dx.doi.org/10.1007/s11858-020-01130-4.
Volarić, T., Vasić, D., Brajković, E., 2017. Adaptive tool for teaching programming using conceptual maps. In: Hadžikadić, M., Avdaković, S. (Eds.), Advanced Technologies, Systems, and Applications. Springer International Publishing, Cham, pp. 335–347.
Wang, J., Wei, K., Radfar, M., Zhang, W., Chung, C., 2021. Encoding syntactic knowledge in transformer encoder for intent detection and slot filling. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 35, pp. 13943–13951.
Weld, H., Huang, X., Long, S., Poon, J., Han, S.C., 2022. A survey of joint intent detection and slot filling models in natural language understanding. ACM Comput. Surv. 55 (8), http://dx.doi.org/10.1145/3547138.
Wiemer-Hastings, P., Allbritton, D., Arnott, E., 2004. RMT: A dialog-based research methods tutor with or without a head. In: Lester, J.C., Vicari, R.M., Paraguaçu, F. (Eds.), Intelligent Tutoring Systems. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 614–623.
Xia, C., Zhang, C., Yan, X., Chang, Y., Yu, P.S., 2018. Zero-shot user intent detection via capsule neural networks. arXiv:1809.00385.
Xin, J., Tang, R., Yu, Y., Lin, J., 2021. The art of abstention: Selective prediction and error regularization for natural language processing. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, Online, pp. 1040–1051. http://dx.doi.org/10.18653/v1/2021.acl-long.84.
Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V., 2019. Xlnet: Generalized autoregressive pretraining for language understanding. Adv. Neural Inf. Process. Syst. 32.
Yang, X., Macdonald, C., Ounis, I., 2018. Using word embeddings in twitter election classification. Inf. Retr. J. 21, 183–207.
Zhang, S., Jiang, J., He, Z., Zhao, X., Fang, J., 2019. A novel slot-gated model combined with a key verb context feature for task request understanding by service robots. IEEE Access 7, 105937–105947. http://dx.doi.org/10.1109/ACCESS.2019.2931576.
Zhang, X., Zhao, J., LeCun, Y., 2015. Character-level convolutional networks for text classification. Adv. Neural Inf. Process. Syst. 28.
Zhou, Z.-H., 2012. Ensemble Methods: Foundations and Algorithms, first ed. Chapman & Hall/CRC, New York, http://dx.doi.org/10.1201/b12207.
Zhu, S., Yu, K., 2017. Encoder-decoder with focus-mechanism for sequence labelling based spoken language understanding. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 5675–5679.