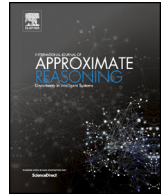


Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

International Journal of Approximate Reasoning

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

Enriching interactive explanations with fuzzy temporal constraint networks

Mariña Canabal-Juanatey^a, Jose M. Alonso-Moral^{a,b,*}, Alejandro Catala^{a,b},
Alberto Bugarín-Diz^{a,b}

^a Centro Singular de Investigación en Tecnoloxías Intelixentes (CITIUS), Universidade de Santiago de Compostela, 15782 Santiago de Compostela, Spain

^b Departamento de Electrónica e Computación, Universidade de Santiago de Compostela, 15782 Santiago de Compostela, Spain

ARTICLE INFO

Keywords:

Fuzzy temporal constraint networks
Fuzzy temporal reasoning
Knowledge graphs
Language models
Conversational agents

ABSTRACT

Humans often use expressions with vague terms which play a fundamental role for effective communication. These expressions are successfully modeled with fuzzy technology, but they are not usually integrated yet with Natural Language Processing models and techniques. Large-scale pre-trained language models yield excellent results in many language tasks, but they have some drawbacks such as their lack of transparency and thorough temporal reasoning capabilities. Therefore, the use of such models may provoke inconsistent or incorrect dialogues in the context of conversational agents which were aimed at providing users of intelligent systems with interactive explanations.

In this paper, we propose a model for fuzzy temporal reasoning to overcome some inconsistencies detected in pre-trained language models in a specific application domain of a conversational agent carefully designed for providing users with explanations which are endowed with a good balance between naturalness and fidelity. More precisely, starting from a knowledge graph that provides an intuitive representation of the entities and relations in the application domain, we describe how to map the temporal information onto a fuzzy temporal constraint network. This formalism allows to represent imprecise temporal information and provides mechanisms for checking consistency in conversations.

In addition, as a proof of concept, we have developed TimeVersa, a conversational agent which integrates the proposed model into an application domain (i.e., a virtual assistant for tourists) that requires handling imprecise temporal constraints. We illustrate in a use case how the agent can identify temporal inconsistencies and answer queries related to temporal information properly. Results after a user study report that users' perception of consistency is significantly higher in a conversation with TimeVersa than in a similar conversation using the well-known GPT-3 Large Language Model, when vague temporal information is involved. The proposed approach is a step forward for developing conversational agents operating in application domains that require temporal reasoning under uncertainty.

* Corresponding author.

E-mail addresses: marina.canabal.juanatey@usc.es (M. Canabal-Juanatey), josemaria.alonso.moral@usc.es (J.M. Alonso-Moral), alejandro.catala@usc.es (A. Catala), alberto.bugarin.diz@usc.es (A. Bugarín-Diz).

<https://doi.org/10.1016/j.ijar.2024.109128>

Received 13 January 2023; Received in revised form 8 November 2023; Accepted 14 January 2024

Available online 17 January 2024

0888-613X/© 2024 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY-NC license (<http://creativecommons.org/licenses/by-nc/4.0/>).

1. Introduction

The emergence of disruptive Deep Learning techniques and tools has revolutionized the approach to applications, services and tasks supported by Natural Language Technologies. Currently, the Natural Language Processing (NLP) field is undergoing a paradigm shift with the rise of Neural Language Models, also known as pre-trained Language Models [1], such as those based on the Bidirectional Encoder Representations from Transformers (BERT) [2] and variants like RoBERTa [3] or ALBERT [4], but also Generative Pre-trained Transformers (GPT) [5]. These models are trained on broad data at scale and are adaptable to a wide range of monolingual and multilingual downstream tasks. For instance, GPT-3 [5] was launched in 2021, it has 175,000 million parameters and 96 layers trained on a corpus of 499,000 million tokens of web content. Other models have more than three times the size, with more than 530,000 million parameters [6], and GPT-4 is announced to be with 1.76 trillion parameters. From the data used for training, basically Internet-based corpora, from Wikipedia to the New York Times, primarily in English, the ability of these models to transform natural language specifications into websites, create basic financial reports or solve language puzzles has been very promising so far.

This quick paradigm shift means that we have only just started to discover the new possibilities and concerns raised by these models. Compared to the state of the art, the results in many areas are so good that systems are close to human level performance in laboratory benchmarks when testing some difficult language tasks such as machine translation or summarization [7,8]. In addition, recent work has shown that pre-trained language models can robustly perform NLP tasks in a few-shot or even in zero-shot fashion when given an adequate task description in its natural language prompt [5,9].

However, despite their impressive capabilities, these pre-trained language models raise severe concerns. First, training these models requires massive computing capabilities what means a huge energy consumption, and this issue is arguable from a sustainable viewpoint. Second, due to their opaque nature, there is a lack of clear understanding of how they work, how they were trained and validated, as well as what emergent properties they present.

In addition, these language models have some limitations. In the case of GPT-3, the following limitations have been identified [10, 11]: output may lack semantic consistency, resulting in text that is gibberish and increasingly nonsensical as it grows longer; outputs embody all the biases that might be found in training data; and outputs may correspond to assertions that are not consonant with the truth. Moreover, the performance exhibited in NLP tasks with low-resources languages (e.g., Vasque, Galician or Gaelic) is much worse than in English, because such languages were underrepresented in the training data. It is worth noting that even if GPT-4 is announced to solve some of the previous limitations, its opaque nature remains. GPT-4 is a black-box model distributed in the form of software as a service and it is impossible to validate its performance from a rigorous scientific viewpoint. These shortcomings are common to other pre-trained language models which could also return inconsistent outputs in some cases. This is certainly a problem, especially when the models are embedded in the core of either question-answering systems or advice-giving systems or any dialog system in general where it is important that the resulting answer is true and correct.

Possible inconsistency in the output of pre-trained language models could be explained by their lack of logical reasoning or very simple deductive and inductive abilities. For instance, such models have no clear understanding of temporal reasoning, and their memory sometimes falls short, so they may easily forget even the simplest temporal constraint that could have been introduced a few messages before. They also suffer from the so-called catastrophic forgetting when are fine-tuned for specific tasks [12]. Besides, an added difficulty is the continuous presence of imprecise expressions in the use of natural language by humans. We often use expressions which include vague or imprecise terms such as “more or less”, “a little bit more”, “not so much”, “approximately”, etc. These expressions capture the imprecision of language and summarize the information in a suitable way for human communication. Notice that automatic reasoning and, more particularly, temporal reasoning become more difficult to manage when dealing with this kind of expressions [13–16].

Considering all the above, the main contribution of this work is enriching interactive explanations with a model that can make reasoning with imprecise temporal information. With this aim, we propose a model that combines a Fuzzy Temporal Constraint Network with a knowledge graph, and we describe how it operates into a conversational agent carefully designed for providing users of intelligent systems with interactive explanations within a specific application domain. Namely, we develop an illustrative use case for the tourism sector to show how the agent handles some potential temporal inconsistency problems properly. Moreover, to conduct a thorough validation process of the model described, we have also carried out an empirical user study in which participants can evaluate some aspects of the conversations related to temporal reasoning.

The rest of the manuscript is organized as follows. Section 2 presents the background on conversational agents until the emergence of pre-trained language models. We also revisit related work with the focus on how to improve the reasoning capabilities of pre-trained language models. In Section 3, we define the preliminary concepts needed to understand the rest of the work. In Section 4, we introduce the model proposed for fuzzy temporal reasoning. Section 5 describes how such a model operates when it is embedded in a conversational agent. We develop a use case within the specific application domain under consideration, i.e., a virtual assistant for tourists. In Section 6, we report an empirical study with human evaluators to examine the possible effects of the lack of a fuzzy temporal reasoning model in a conversation with a conversational agent. Finally, Section 7 summarizes the main conclusions and points out future work.

2. Related work

The first implementation of a conversational agent was achieved in 1966 with the development of ELIZA [17], which relied heavily on linguistic rules and pattern matching techniques. Although its scope of knowledge was very limited, ELIZA was a landmark

system that stimulated further research in the field. Following ELIZA, PARRY and Racter were introduced in 1972 and 1988 [18], respectively. Later, in 1995, the Artificial Linguistic Internet Computer Entity (ALICE) was introduced in [19]. Its heuristic matching patterns based on the Artificial Intelligence Mark-up Language proved a substantial upgrade with respect to previous conversational agents. Here, each pattern (i.e., each user input) is associated with an output (referred to as a template). The main limitation of rules and pattern matching-based conversational agents is that they are domain dependent. Accordingly, they lack flexibility because they rely on manually written rules for specific domains.

With the advances in Artificial Intelligence techniques and NLP tools combined with the availability of computational power, new frameworks and algorithms were created to implement more advanced conversational agents, also called chatbots. In this context, two other chatbots are Jabberwacky and its successor Cleverbot,¹ released in 1997 and 2006, respectively. They belong to a category that can be referred to as information retrieval-based conversational agents. They generate their responses by selecting a suitable sentence from a large dialogue corpus, i.e., a database of stored conversations. Therefore, their main advantage is that they ensure the quality of the responses, but creating the necessary knowledge can be costly and time consuming.

While conversational agents in the two previous categories rely on existing utterances, the so-called generative model-based conversational agents generate their responses word by word using statistical models. When the objective is modelling probability distributions over language (e.g., in terms of which words are more likely to appear before, after or in between some other words), the underlying generative models are called language models. Beyond using only single response generation approaches, some researchers attempt to hybridize the retrieval and rule-based methods with generative-based methods [20].

Originally designed for Machine Translation, one of the most famous models for developing generative-based methods is the neural network model sequence to sequence [21], which is used at the heart of several state-of-the-art conversational agents (e.g., MILABOT [22], Google's Meena [23] or Microsoft's XiaoIce [24]). Furthermore, one of the most outstanding innovations in Deep Learning language models has been the introduction of Transformers [25]. The Transformer-based architectures are endowed with training parallelization and this fact permits training on larger datasets than was originally achievable. This led to the development of large-scale pre-trained models such as BERT [2] (Bidirectional Encoder Representations from Transformers), GPT [26] (Generative Pre-trained Transformer) and their successors (ALBERT [4], RoBERTa [3], BART [7], GPT-3 [5] or GPT-4 [27]), which were trained with huge datasets, such as Common Crawl,² and may be fine-tuned for downstream task without training a new model from scratch. These pre-trained models have outperformed previous state-of-the-art results on almost all NLP tasks such as machine translation, summarization [7,8], named entity recognition [28] or sentiment analysis [29], some of which have been reported to be close to human-level performance.

Despite their powerful capabilities, pre-trained models also present limitations. For instance, in accepting large Internet-based datasets as training data we risk perpetuating dominant viewpoints, increasing power imbalances and further reifying inequality, and introducing stereotypical and derogatory associations among gender, race, ethnicity or disability status that may come with them [30]. In addition, regarding environmental wellbeing, building such models from scratch is very demanding from a computational viewpoint and comes with a very large carbon footprint [31]. The most relevant shortcomings for us, on which we will focus in this paper, are related to the possibility of returning inconsistent or incorrect responses. This was criticized in the case of GPT-3 [10,11,32], but the limitations are common to all pre-trained models mentioned previously and can be explained by a lack of logical reasoning or very simple deductive and inductive abilities in current language models. For instance, some previous publications have shown that pre-trained models lack thorough temporal reasoning [33–35]. However, recognition and understanding of temporal expressions could be crucial for conversational agents, e.g., in scheduling reminders or meetings and booking events, what shapes our motivation behind conducting this work.

It is worth noting that much work has already been done on the recognition, extraction, and normalization of temporal expressions [36–39]. Nevertheless, to the best of our knowledge, there has been little work proposing solutions to overcome the possible inconsistencies caused by a lack of temporal reasoning in the context of conversational agents. Existing academic work mostly focuses on improving the mathematical capabilities of pre-trained models [40,41], but not specifically on temporal reasoning, let alone when temporal expressions include imprecise terms. Thus, in this work we will introduce a new model based on a knowledge graph for semantic representation and a fuzzy temporal constraint network for reasoning, that enriches conversational agents with the ability to detect and avoid temporal inconsistencies.

There are several approaches with the aim of integrating knowledge graphs into dialog systems for different purposes, for instance tracking user goal during a dialog [42] or incorporating external knowledge and capturing semantics of user inputs [43]. Furthermore, there have been some attempts on modelling temporal information in knowledge graphs with the introduction of Temporal Knowledge Graphs [44–48], which also incorporate the start and end timestamps of each entity. The main difference with our approach is that we intend to use a knowledge graph to represent temporal constraints that restrict the occurrence of events, but they do not necessarily have to have happened. In addition, as we will describe later, we will handle the imprecision present in many temporal expressions in natural language by mapping the knowledge graph onto a fuzzy temporal constraints network.

Finally, we would like to conclude this section with the following clarification. Commercial chatbots (e.g., ChatGPT released by OpenAI, BARD from Google, or Bing from Microsoft) are not language models. Indeed, they include several engineering layers which are aimed at fixing undesired behavior of the underlying language models. Unfortunately, there is no public information about how their associated engineering architectures are developed and tested.

¹ <https://www.cleverbot.com/>.

² <https://commoncrawl.org/>.

3. Preliminaries

Before presenting the new model in Section 4, it is necessary to define here some preliminary concepts related to fuzzy sets and possibility theory [49].

We consider that time is a discrete set $\tau = \{t_0, t_1, \dots, t_i, \dots\}$, where t_0 is the time origin and t_i represents a precise time instant, for every $i \in \mathbb{N}$. Without losing generality, we assume $t_0 < t_1 < \dots < t_i < \dots$, with $t_{i+1} - t_i$ being constant for every $i \in \mathbb{N}$. We denote as \mathcal{M} the set of possible units of time.

Definition 1. Following [50], we define a *fuzzy time instant* a (also called *date* in [14,15]) as a possibility distribution:

$$\pi_a : \tau \rightarrow [0, 1],$$

where, given a precise time instant $t \in \tau$, $\pi_a(t) \in [0, 1]$ represents the possibility of a being precisely t .

For simplicity, even if other possibility distributions are allowed, here we assume that any fuzzy time instant a is given by a trapezoidal distribution, i.e., a function of the form:

$$\pi_a(t) = \begin{cases} \frac{t - \alpha}{\beta - \alpha} & \text{if } \alpha \leq t < \beta, \\ 1 & \text{if } \beta \leq t < \gamma, \\ \frac{t - \delta}{\gamma - \delta} & \text{if } \gamma \leq t < \delta, \\ 0, & \text{otherwise,} \end{cases}$$

with $\alpha, \beta, \gamma, \delta \in \tau$ such that $\alpha \leq \beta \leq \gamma \leq \delta$. We denote a trapezoidal distribution as the 4-tuple $(\alpha, \beta, \gamma, \delta)$. This function becomes a triangular distribution in case that $\beta = \gamma$.

Definition 2. Following [14,50], we define a *fuzzy time extent* e as a possibility distribution:

$$\pi_e : \mathcal{M} \rightarrow [0, 1],$$

where, given $m \in \mathcal{M}$, $\pi_e(m) \in [0, 1]$ represents the possibility of e being precisely a duration of m units of time. Again, we will consider that any fuzzy time extent is given by a trapezoidal distribution.

Let us suppose that a and b are two fuzzy time instants whose trapezoidal distributions are determined by the parameters $\alpha, \beta, \gamma, \delta$ and $\alpha', \beta', \gamma', \delta'$, respectively. Then, following [15,50], we can represent the fuzzy temporal distance between them, $d(a, b)$, by means of the fuzzy time extent:

$$\pi_{d(a,b)}(m) := \max_{\substack{m=t-s \\ t,s \in \tau}} \{ \min\{\pi_a(s), \pi_b(t)\} \}, \forall m \in \mathcal{M}. \tag{1}$$

It is important to remark that $\pi_{d(a,b)}$ and $\pi_{d(b,a)}$ are symmetric distributions with respect to the y-axis because the distance is directed.

Note that a fuzzy time instant a can always be considered as the fuzzy time extent $e = d(t_0, a)$. As an example, Fig. 1 shows the graphical representation of two fuzzy time instants: $a =$ “around 12 PM” (Fig. 1(a)) and $b =$ “a little bit before 12:30 PM” (Fig. 1(b)), given by the trapezoidal distribution $\pi_a = (11.45, 11.55, 12.05, 12.15)$ and the triangular distribution $\pi_b = (12.10, 12.25, 12.25, 12.30)$, respectively. We are considering that the time axis τ is formed by the precise time instants from 11.30 AM to 12.30 PM and the unit of time is 5 minutes.

In Fig. 2, we present the graphical representation of two examples of fuzzy time extents: $e =$ “about 35 minutes” (Fig. 2(a)), represented by the trapezoidal distribution $\pi_e = (20, 30, 40, 50)$, and $f = d(a, b)$, with a different shape obtained as the fuzzy temporal distance between fuzzy time instants a and b given by the expression (1) (as defined in Fig. 2(b)), with an associated semantics of “a little bit less than approximately 30 min”.

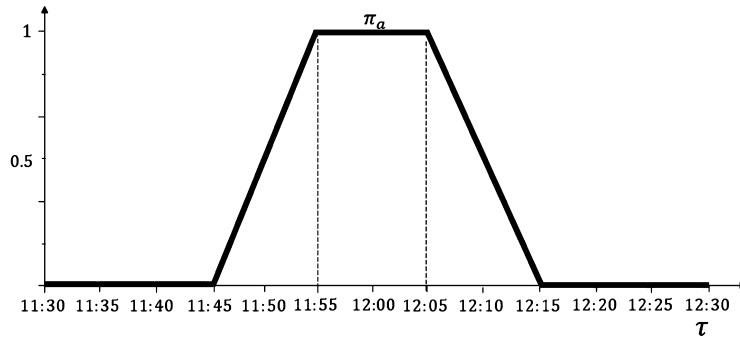
Following [50], we now introduce some basic binary operations on fuzzy time extents which will be used later on.

Definition 3. We define the *intersection* of two time extents e and e' , noted as $e \cap e'$, as the fuzzy time extent given by the possibility distribution:

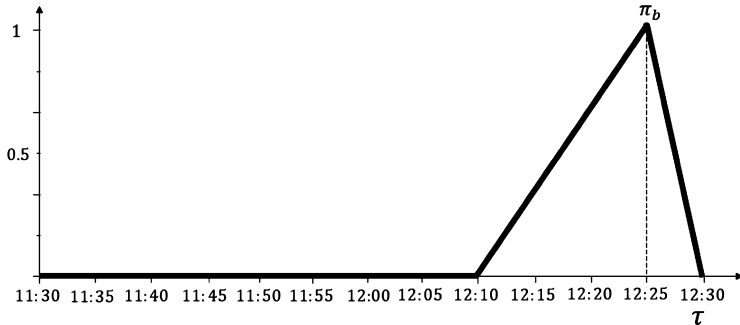
$$(\pi_e \cap \pi_{e'})(m) := \min\{\pi_e(m), \pi_{e'}(m)\}, \forall m \in \mathcal{M}.$$

Definition 4. We define the *composition* of two time extents e and e' , noted as $e \circ e'$, as the fuzzy time extent given by the possibility distribution:

$$(\pi_e \circ \pi_{e'})(m) := \max_{\substack{m=m_1+m_2 \\ m_1, m_2 \in \mathcal{M}}} \{ \min\{\pi_e(m_1), \pi_{e'}(m_2)\} \}, \forall m \in \mathcal{M}.$$



(a) Fuzzy time instant $a = \text{“around 12 PM”}$.



(b) Fuzzy time instant $b = \text{“a little bit before 12.30 PM”}$.

Fig. 1. Examples of possibility distributions of fuzzy time instants.

Note that the composition operator is nothing but the addition operation of two fuzzy quantities using the extension principle described in [51].

4. A model for fuzzy temporal reasoning

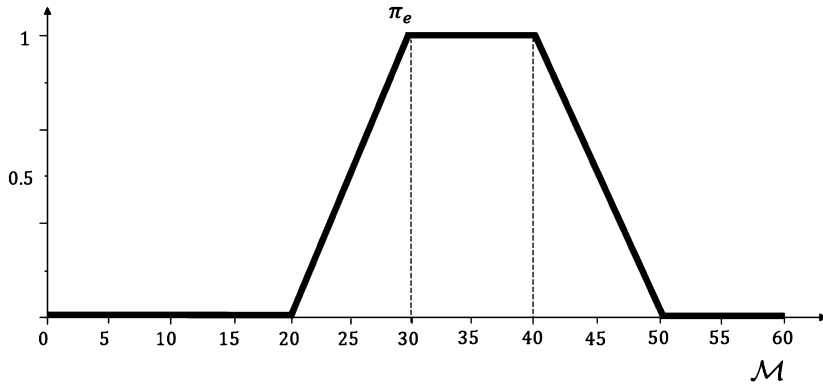
In this section, we present a model for managing fuzzy temporal expressions and detecting inconsistencies between them. Starting from a knowledge graph representative of the specific domain, the model projects the temporal information onto a fuzzy temporal constraint network, which provides reasoning mechanisms for checking consistency and answering explanatory queries.

More precisely, we assume a knowledge graph KG is known and given beforehand. KG employs a graph-based data model to represent knowledge about a specific application domain, providing a concise and intuitive abstraction of data involved [52,53]. It is worth noting that there is no single commonly accepted formal definition and representation of knowledge graphs [54,55]. Accordingly, we simply define KG as a directed labeled graph whose nodes are entities of interest in the application domain. These entities represent information which may be temporal or not. If $\{n_1, \dots, n_m\}$ represents the set of m nodes in KG , for each triplet (n_i, r, n_j) , n_i is taken as the parent node, which has the relation r with the child node n_j . We distinguish between entities that include temporal information, which we will denote temporal entities, and those that do not. Thus, we consider precise time instants, fuzzy time instants or fuzzy time extents. Moreover, we consider both temporal relations (e.g., “starts at” or “has a duration of”) and non-temporal relations (e.g., “is a subclass of” or “is located in”). In the case of dealing with temporal information, the relation r is represented as an arc linking the two given nodes in a Fuzzy Temporal Constraint Network \mathcal{N} (see formal definitions and further details in Section 4.1). The procedure for mapping KG onto \mathcal{N} is described in Section 4.2.

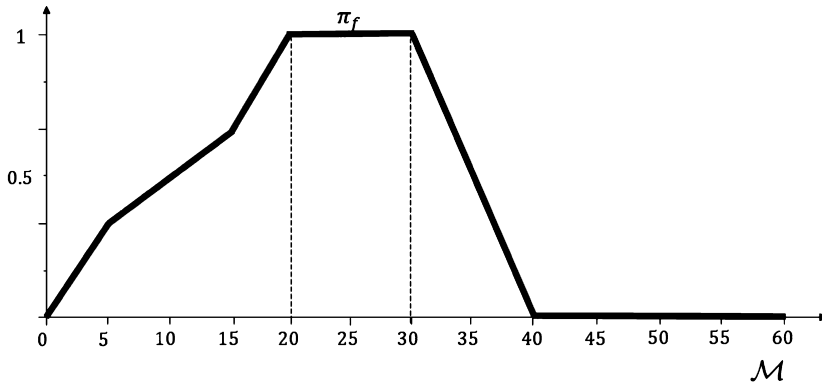
4.1. Fuzzy temporal constraint networks

First, we focus on defining a formalism for knowledge representation of, and reasoning with, temporal information which is not precise but vague or/and uncertain.

Let us assume we have a set of n instantaneous events whose precise time instants of occurrence, $v_1, \dots, v_n \in \tau$, are unknown. If we add some imprecise information about one of the time instants, v_i , we are establishing a constraint over the possible values for v_i , called unary constraint and denoted as C_i . This constraint can be represented as a fuzzy time instant by means of a possibility distribution π_{C_i} defined over τ . If this imprecise information establishes a duration constraint over the possible values of the temporal distance between two precise time instants v_i and v_j , then we call it binary constraint and we denote it as C_{ij} . In this case, the



(a) Fuzzy time extent $e =$ “about 35 minutes”.



(b) Fuzzy time extent $f = d(a, b) =$ “a little bit less than approximately 30 minutes”.

Fig. 2. Examples of possibility distributions of fuzzy time extents.

constraint can be seen as a fuzzy time extent given by a possibility distribution $\pi_{C_{ij}}$ over \mathcal{M} . Introducing a precise origin of times, $v_0 = t_0$, we can represent every unary constraint C_i as the binary constraint C_{0i} . For simplicity, we will use the notation π_{ij} instead of $\pi_{C_{ij}}$, $\forall i, j \in \{0, \dots, n\}$.

Definition 5. Following [14,16,50], we define a *Fuzzy Temporal Constraint Network* (FTCN) \mathcal{N} as a pair (\mathcal{V}, C) , where $\mathcal{V} = \{v_0, \dots, v_n\}$ is a finite set of nodes, with each node representing an unknown precise time instant $v_i \in \tau$, $\forall i \in \{1, \dots, n\}$ and $v_0 = t_0$ the time origin, and a finite set of directed arcs $C = \{C_{ij} \mid i, j \in \{0, \dots, n\}\}$ that represent binary constraints between the nodes. Each constraint C_{ij} is a fuzzy time extent that has associated a possibility distribution π_{ij} .

Definition 6. Given $\alpha \in [0, 1]$, a *solution* [16] of an FTCN \mathcal{N} at least to the degree α is a n -tuple $S = \{t_1, \dots, t_n\}$ such that $\mathcal{N}(S) \geq \alpha$, where $t_i \in \tau$, $\forall i \in \{0, \dots, n\}$ and

$$\mathcal{N}(S) := \min_{i,j \in \{0, \dots, n\}} \{\pi_{ij}(t_j - t_i)\}.$$

Definition 7. Given $\alpha \in [0, 1]$, a precise time instant $t \in \tau$ is a *feasible value* [16] at least to the degree α for a node v_i of an FTCN \mathcal{N} if there exists a solution S such that $v_i = t$ and $\mathcal{N}(S) \geq \alpha$.

Definition 8. The *degree of consistency* [16] of an FTCN \mathcal{N} is defined by the following expression:

$$Cons(\mathcal{N}) := \max_{S \in \tau^n} \mathcal{N}(S).$$

If $Cons(\mathcal{N}) = \alpha$, then we say that \mathcal{N} is α -consistent. \mathcal{N} is *consistent* if it is 1-consistent and \mathcal{N} is *inconsistent* if there is no solution ($\alpha = 0$).

Calculating the feasible values to a certain degree for a node and the consistency of the entire network using the expressions defined above implies first finding all possible solutions $\mathcal{N}(S)$ at any degree. This process means combining all the values of the axis of precise time instants considered, τ , in the particular case that it is finite. With the aim of making the process more efficient, the computation of queries on FTCNs is based on transforming the network into an equivalent representation in which the query answers can be more directly obtained [16,50].

Two networks defined over the same set of nodes are equivalent if they represent the same solution set with the same degree. We will say that an FTCN $\mathcal{N} = (\mathcal{V}, C)$ is tighter than another one $\mathcal{N}' = (\mathcal{V}, C')$ with the same set of nodes if for all $C'_{ij} \in C'$ there exists a constraint $C_{ij} \in C$ such that their associated possibility distributions satisfy that $\pi_{ij}(m) \leq \pi'_{ij}(m)$, $\forall m \in \mathcal{M}$. We will look for the tightest equivalent network, called the *minimal network*.

Between each pair of nodes in \mathcal{N} , v_i and v_j , there is a direct constraint C_{ij} . There may be additional induced constraints, each corresponding to a possible path connecting the two nodes. The induced constraint from the node v_i to v_j in a path of length k , $1 \leq k \leq n$, is given by the composition of the direct constraints between each pair of consecutive nodes belonging to the path. We will use the notation C_{ij}^k to represent the intersection of the induced constraints corresponding to all the paths of length k from v_i to v_j , given by the following possibility distribution:

$$\pi_{ij}^k := \bigcap_{\substack{i_0, i_1, \dots, i_k \in V_n^k \\ i_0 = i \\ i_k = j}} \pi_{i_0 i_1} \circ \pi_{i_1 i_2} \circ \dots \circ \pi_{i_{k-1} i_k}, \quad (2)$$

where V_n^k represents the set of k -element variations of n elements. Finally, we will denote as C_{ij}^{\min} the intersection of all the induced constraints for paths from v_i to v_j of any length k , given by the following possibility distribution:

$$\pi_{ij}^{\min} := \bigcap_{k=1}^n \pi_{ij}^k \quad (3)$$

Note that the expressions above can only be calculated if there exists a single constraint between each pair of nodes in the network. Hence, it is necessary to take into account the following aspects:

- The absence of constraints between two nodes v_i and v_j is equivalent to consider a constraint C_{ij} whose associated function is the constant function $\pi_{ij}(m) := 1, \forall m \in \mathcal{M}$. Therefore, there always exists a constraint between each pair of nodes in the FTCN.
- If π_{ij} stands for the possibility distribution associated to the constraint C_{ij} for two nodes v_i and v_j , we will also consider that there exists a constraint C_{ji} between v_j and v_i such that $\pi_{ji}(m) = \pi_{ij}(-m)$, $\forall m \in \mathcal{M}$.
- For each node v_i with itself, we consider a constraint C_{ii} such that $\pi_{ii}(m) = 1$ if $m = 0$ and $\pi_{ii}(m) = 0$ otherwise.
- If there exists more than one constraint between a pair of nodes in the network, we reduce them into an equivalent single constraint applying the intersection of time extents.

The FTCN $\mathcal{N}_{\min} = (\mathcal{V}, C^{\min})$, where $C^{\min} = \{C_{ij}^{\min} | 0 \leq i, j \leq n\}$ and each constraint C_{ij}^{\min} is associated with π_{ij}^{\min} , $\forall i, j \in \{0, \dots, n\}$, is the minimal network corresponding to the FTCN \mathcal{N} . Such minimal network contains $n(n-1)/2$ edges, with n being the number of nodes. According to [50], the calculation of C^{\min} by means of the aforementioned expressions is exponential with the number of nodes. Nevertheless, some authors have applied path-consistency algorithms, which are executed with a polynomial increase of time, to different types of precise temporal networks [13]. The same idea can be applied to the minimization of an FTCN [16,50]. We have adapted Floyd-Warshall's all-pairs-shortest-paths algorithm [56] to compute the minimal network associated to an FTCN \mathcal{N} in a polynomial time (see the pseudocode in Algorithm 1).

Algorithm 1 Pseudocode for computing a minimal network.

```

1: function MINIMALNETWORK( $n, \mathcal{N} = (\mathcal{V}, C)$ )
2:   for  $k = 1, \dots, n$  do
3:     for  $i = 1, \dots, n$  do
4:       for  $j = 1, \dots, n$  do
5:          $\pi_{ij} \leftarrow \pi_{ij} \cap (\pi_{ik} \circ \pi_{kj})$ 
6:         if  $\pi_{ij} = 0$  then
7:           Exit (The FTCN is inconsistent)
8:   return  $\mathcal{N}_{\min}$ 

```

The minimal network allows us to determinate in an easy way the consistency of the network and feasible values for each node. However, edges in this network lack semantic interpretability because they are the result of intersection of the constraints that were initially mapped from the knowledge graph.

From now on, we consider that $\mathcal{N} = (\mathcal{V}, C)$ represents a minimal network. The construction of the minimal network ensures that there exists a solution $S_{\max} = \{t_1, \dots, t_n\}$, such that $t_i \in \tau$, $\forall i \in \{1, \dots, n\}$ and:

$$\pi_{ij}(t_j - t_i) = \max_{m \in \mathcal{M}} \{\pi_{ij}(m)\}, \quad \forall i, j \in \{0, \dots, n\}.$$

Hence, following the Definition 8, the degree of consistency can be directly obtained as follows:

$$\text{Cons}(\mathcal{N}) = \min_{i,j \in \{0, \dots, n\}} \left\{ \max_{m \in \mathcal{M}} \{ \pi_{ij}(m) \} \right\}.$$

Similarly, since we manage the tightest constraint for each pair of nodes, computing the feasibility of a value t for a node v_i is reduced to calculate the value $\pi_{0i}(t - t_0)$.

In addition, the minimal network simplifies the introduction of two new concepts: the degree of compatibility of a new constraint with the network and the feasible values for a distance between two nodes.

Definition 9. Let C_{ij}^* be a fuzzy temporal constraint given by a possibility distribution π_{ij}^* . The *degree of compatibility* [16] of C_{ij}^* with \mathcal{N} is defined by the following expression:

$$\text{Comp}(C_{ij}^*, \mathcal{N}) := \max_{m \in \mathcal{M}} \left\{ \min \{ \pi_{ij}^*(m), \pi_{ij}(m) \} \right\}.$$

Definition 10. Given $\alpha \in [0, 1]$, a value of $m \in \mathcal{M}$ is a *feasible value* for the distance between two nodes v_i and v_j in an FTCN \mathcal{N} at least to the degree α if it is satisfied that:

$$\pi_{ij}(m) \geq \alpha.$$

To sum up, once we have built the minimal network for an FTCN, we can obtain the following information: (i) consistency of the network; (ii) compatibility of a new constraint with the network; and (iii) feasible values both for a node and for any distance between two nodes.

4.2. Mapping a knowledge graph onto an FTCN

An FTCN provides us with a formalism for representing of, and reasoning with, fuzzy temporal information. However, in the context of conversational agents, information is not usually provided in the form of precise time instants and constraints on their temporal distance. Furthermore, some additional constraints may have to be added to the network if we consider certain semantic relations between entities in the domain. Therefore, we propose a knowledge graph (KG) as an initial point to represent in a more natural way the information of a specific application domain.

A lot of work has been done on entity extraction from KGs [57], refining the information gathered [53] and completion [58]. Assuming we start from a KG that includes the information of interest from the domain consistently related, the question now is how to project it onto an FTCN in order to allow queries such as those specified in section 4.1.

First of all, it is important to point out that not all the entities and relations embedded in the KG need to be transferred to the FTCN. It would be sufficient to map only those parts of the graph that are related to temporal information, thus facilitating the implementation of the mapping procedure (see pseudocode in Algorithm 2). For that reason, we will pay major attention to the procedure to map those nodes in the KG which are connected by a temporal relation.

Let us suppose that the KG contains the triplet (n_i, r, n_j) , r being a temporal relation. Then, we consider the following situations:

- **n_i is a non-temporal entity and n_j is a temporal entity.** We will suppose that any non-temporal entity has a start time instant and an end time instant. In the absence of information on their duration, we will also assume that the two instants have the same value (possibly unknown). Thus, the existence of this triplet in KG implies the creation of two nodes associated with n_i in the FTCN, one for the precise time instant of start and another one for the precise time instant of end (provided they do not already exist). In addition, a constraint is added on the time elapsed between them, which in the absence of information is assumed to be zero. If the child node is temporal, due to the scope of this work we consider three possible relations between both nodes:
 - One option is that n_i “starts at” n_j . On the one hand, if n_j is a precise time instant, the node associated with the precise time instant at which the entity n_i starts is set to the value of the precise time instant represented by n_j . As an example, we can figure out that in the application domain we collected the information about the boat trip to go to Cíes Island and we had the triplet (*outbound trip, starts at, 11.15 AM*). On the other hand, if n_j is a fuzzy time instant, this relation defines a constraint over the temporal distance between the origin of times (the node v_0 in the FTCN) and the node in the FTCN associated with the precise time instant of start of entity n_i . This would be the case of a triplet such as (*outbound trip, starts at, around 12 PM*).
 - The second option is that n_i “finishes at” n_j and the procedure is analogous to that described above, replacing the node representing the precise time instant at which the entity n_i starts by the node representing its precise time instant of end.
 - The other alternative is that n_i “has a duration of” n_j , with n_j being a fuzzy time extent, as in the example (*outbound trip, has a duration of, about 35 minutes*). In this case, n_j is projected into the FTCN by means of a constraint between the nodes associated with its precise time instant of start and its precise time instant of end. This constraint expresses the semantics of the temporal expression described in n_j .

Algorithm 2 Pseudocode for mapping a KG onto an FTCN.

```

1: function GRAPHTONETWORKMAPPING(KG,  $\mathcal{N} = (\mathcal{V}, C)$ )
2:   for  $(n_i, r, n_j)$  in KG do
3:     if  $\{v_{i_{start}}, v_{i_{end}}\} \not\subseteq \mathcal{V}$  then
4:        $\mathcal{V} \leftarrow \mathcal{V} \cup \{v_{i_{start}}, v_{i_{end}}\}$ 
5:        $C_{i_{start}^i_{end}} \leftarrow \pi_U$ 
6:       addConstraint( $C_{i_{start}^i_{end}}, C$ )
7:     if isTemporalNode( $n_j$ ) then
8:       if  $r = \text{"has a duration of"}$  then
9:          $C_{i_{start}^i_{end}} \leftarrow \text{createNodeConstraint}(n_j)$ 
10:        addConstraint( $C_{i_{start}^i_{end}}, C$ )
11:       if  $r = \text{"starts at"}$  then
12:          $C_{0_{i_{start}}} \leftarrow \text{createNodeConstraint}(n_j)$ 
13:         addConstraint( $C_{0_{i_{start}}}, C$ )
14:       if  $r = \text{"finishes at"}$  then
15:          $C_{0_{i_{end}}} \leftarrow \text{createNodeConstraint}(n_j)$ 
16:         addConstraint( $C_{0_{i_{end}}}, C$ )
17:     else
18:       if  $\{v_{j_{start}}, v_{j_{end}}\} \not\subseteq \mathcal{V}$  then
19:          $\mathcal{V} \leftarrow \mathcal{V} \cup \{v_{j_{start}}, v_{j_{end}}\}$ 
20:          $C_{j_{start}^j_{end}} \leftarrow \pi_U$ 
21:         addConstraint( $C_{j_{start}^j_{end}}, C$ )
22:       if  $r = \text{"happens <modifier> <m units of time> before"}$  then
23:          $C_{i_{end}^j_{start}} \leftarrow \text{createRelationConstraint}(r)$ 
24:         addConstraint( $C_{i_{end}^j_{start}}, C$ )
25:       if  $r = \text{"happens during"}$  then
26:          $C_{i_{start}^i_{start}} \leftarrow \text{createRelationConstraint}(\text{"happens before"})$ 
27:          $C_{i_{end}^i_{end}} \leftarrow \text{createRelationConstraint}(\text{"happens before"})$ 
28:         addConstraint( $C_{i_{start}^i_{start}}, C$ )
29:         addConstraint( $C_{i_{end}^i_{end}}, C$ )
30:   return  $\mathcal{N}$ 

```

- n_i and n_j are non-temporal entities. Analogously to the previous case, two nodes and a constraint between them associated to each non-temporary entity are created in the FTCN. Two possible relations are considered:
 - The first one is the relation “happens <modifier> <m units of time> before”, where $m \in \mathcal{M}$ and the modifier could be an expression (e.g., “approximately”, “more or less”, “a little more than”, etc.) which adds information about the temporal distance between the nodes. Notice that, if n_i happens after n_j instead of before, then the relation can be converted into the above expression just by inverting the parent and the child nodes. This relation between the nodes adds a constraint to the FTCN between the precise time instant of end of n_i and the precise time instant of start of n_j by means of a fuzzy time extent that represents the semantics of the temporal expression described in r . If no modifier is specified or if the modifier is “exactly” (or a synonym), the associated fuzzy time extent will consist of a possibility distribution taking the value 1 at m and 0 otherwise. In any other case, the possibility distribution associated with the constraint will depend on the definition of the modifier, which could be different according to the application domain. Following on the previous example, this case would represent a triplet such as (*outbound trip*, *happens before*, *return trip*).
 - The other possibility we consider is that n_i “happens during” n_j . In this case, the relation adds two new constraints to the FTCN: one constraint to express that the precise time instant of start of n_j happens before (or at the same time as) the precise time instant of start of n_i and another one to express that the precise time instant of end of n_i happens before (or at the same time as) the precise time instant of end of n_j . One example would be a triplet such as (*bar service*, *happens during*, *outbound trip*).
- **Other cases.** The case when n_i is a temporal entity and n_j is a non-temporal entity as well as the case when n_i and n_j are both temporal entities do not make sense in the context of this work, so we will not consider them.

In the case that it is necessary to add a new constraint between two nodes in the network, v_i and v_j , when one already exists, the constraint C_{ij} is modified so that it represents the intersection between the existing one and the new one.

Initially, it might seem that projecting only the nodes directly connected through temporal relations is sufficient to transfer all the temporal information to the FTCN. However, there could be non-temporal relations in the graph that caused the inheritance of certain relations, including temporal ones. An illustrative example is the relation “is a subclass of”, for which it is obvious that the node representing the subclass must inherit the relations of the superclass node.

Accordingly, before projecting those nodes of interest to the FTCN, it is required to transfer all temporal relations inherited in this way, if necessary. Following an approach based on going through all the nodes by exploring and relocating their relations has exponential complexity on the number of nodes. Therefore, at this point it is advisable to apply some existing graph search algorithm to reduce computational complexity. Moreover, evaluating whether any semantic relation has the property of inheriting temporal

Table 1

Fragment of a real conversation between TimeVersa and a user. Underlined, temporal expressions relevant for the conversation.

Move	Participant	Utterance
m_1	User	Hello! I would like to be in Cíes Islands <u>around 12 PM</u> . What time should I take the boat from Vigo?
m_2	TimeVersa	There is a boat <u>at 11.15 AM</u> that is suitable for you.
m_3	User	Why do you think this boat is suitable for me?
m_4	TimeVersa	The boat trip from Vigo to Cíes Islands lasts <u>about 35 minutes</u> , so if you take the boat <u>at 11.15 AM</u> you can be there <u>around 12 PM</u> .
m_5	User	Could you specify what “ <u>around 12 PM</u> ” means for you?
m_6	TimeVersa	It means <u>between 11.50 AM and 12.10 PM</u> .
m_7	User	OK. Book the boat for me, please.
m_8	TimeVersa	Could I help you with anything else?
m_9	User	Yes. Once we have reached Cíes Islands <u>around 12 PM</u> , we had thought of doing the hiking route “Faro de Cíes”. Do you think we can be back in Vigo <u>before 2 PM</u> ?
m_{10}	TimeVersa	No, this is not possible.
m_{11}	User	Why not?
m_{12}	TimeVersa	Because the route will take you <u>about 3 hours</u> . If you arrive the Islands <u>around 12 PM</u> , you can't be back in Vigo <u>before 2 PM</u> .
m_{13}	User	OK, I understand. So, what time can we be back?
m_{14}	TimeVersa	You can be back in Vigo <u>around 4.35 PM</u> if you take the boat <u>at 4 PM</u> .

information about its entities is a domain-dependent problem. Thus, this important step, which concerns the inheritance of certain relations, deserves a deeper analysis beyond the scope of this work.

Finally, given a conversation, the use of the FTCN to represent, and to make reasoning with, fuzzy temporal information comprises the following steps (which will be instantiated and described in detail for the illustrative use case in the following section): (i) handling imprecision; (ii) checking consistency; (iii) checking compatibility with a new constraint; (iv) setting values for nodes; and (v) returning feasible values.

5. An illustrative use case

In this section, we present an example for showing how to integrate the model described in the previous section into a conversational agent which provides users with interactive explanations. To facilitate reproducibility, and in accordance with guidelines for developing responsible AI, we have made the code for running this illustrative use case available at <https://nextcloud.citius.usc.es/s/CcAeEo9YSNEsLTM>.

The main objective of this illustrative use case is to show how the model presented in the previous section tackles several inconsistencies related with temporal reasoning. Namely, we have designed and developed TimeVersa which represents a proof of concept of a virtual assistant for tourists. More precisely, TimeVersa is a conversational agent oriented to recommend, book, and give information about the boat trip from Vigo to Cíes Islands, as well as related activities there. The agent can respond properly to questions about duration and schedule of the boats, restaurant availability, and duration of the hiking routes. It also assists users to book the restaurant or buy the tickets for the trip. We employ Rasa,³ an open-source conversational AI engine, for the development of this proof-of-concept agent.

Table 1 shows an example of interaction with TimeVersa, in which the user inquires more detailed explanations for the given recommendations. This use case evidences the necessity of having a model for fuzzy temporal reasoning and shows how the agent naturally conducts the most representative queries to the FTCN.

The implementation of the conversational agent relies on a global knowledge graph, which contains all available knowledge of the application domain as illustrated in Fig. 3. However, it is common that only certain entities and relations in the domain are relevant for a specific conversation, so it is necessary to manage a sub-graph with the information of interest for the user according with the current conversation. Once the agent has identified the relevant entities in the user message, it selects the corresponding information from the global knowledge graph and adds new entities and relations as needed. Then, this sub-graph is projected onto an FTCN following the procedure described in Section 4.2. New entities and relations can only be added to the sub-graph if they assure the consistency of the FTCN. Otherwise, the agent informs the user that the current combination of temporal constraints is not possible. It is worth noting that we have highlighted in Fig. 3 the sub-graph associated with the entities and relations added in agreement with the conversation previously recorded in Table 1.

In addition, Fig. 4 shows the evolution of the FTCN as the conversation in Table 1 progresses. It is worth noting that, in this case, the minimal network contains $n = 7$ nodes and $n(n - 1)/2 = 21$ edges. This is a fully connected synthetic network where edges lack semantic temporal meaning because they result from applying operations described in equations (2) and (3) over the possibility distributions of original edges that were mapped from the knowledge graph. The procedure for computing the minimal network is described in previous section (see Algorithm 1).

Likewise, information about entities connected through non-temporal relations is omitted, as it does not influence network operations and the simplified representation of the graph is easier to visualize.

Below, we detail step by step how this model is used to represent, and to make reasoning with, fuzzy temporal knowledge regarding the conversation depicted in Table 1. For this example, we set $\alpha = 0.5$ as the threshold for determining consistency,

³ <https://rasa.com/docs/>.



Fig. 3. Knowledge graph associated with the use case under consideration. On a continuous line, the entities and relations that make up the global knowledge graph from which TimeVersa starts. Nodes colored in yellow represent temporal entities, while white nodes are associated to non-temporal entities. The elements circled and highlighted in red constitute the sub-graph with the information of interest for the conversation shown in Table 1. The entities and relations painted with a dashed line are those added during the conversation. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

compatibility, and feasibility. Notice that setting this threshold α amounts to convert an FTCN into a crisp TCN by taking the alpha-cuts of the fuzzy constraints in the network. This reduces computational complexity and thus facilitates graphical representation in the use case, when comparing both fuzzy and crisp versions of the proposed temporal model.

- **Handling imprecision.** The FTCN works implicitly with imprecise constraints from the global knowledge graph (e.g., “about 35 minutes”, which can be defined as the fuzzy time extent in Fig. 2(a)). Furthermore, when the user sends the first message (move m_1) asking for a boat to arrive the islands “around 12 PM” (which can be defined as the fuzzy time instant in the Fig. 1(a)), the model facilitates handling this imprecise expression and allows the agent to respond adequately. Similarly, the agent can specify how “around 12 PM” is understood in this context (move m_6) when the user asks for more details about the recommendation.
- **Checking consistency.** Some dialogue moves (m_1 , m_7 , and m_9) provoke that new nodes are added on the fly to the FTCN. This implies recomputing the corresponding minimal network and checking its consistency for each corresponding dialogue move.
- **Checking compatibility with a new constraint.** Both move m_1 and move m_9 (see Figs. 4(a) and 4(c), respectively) involve calculating constraint compatibility with the network. In the first case the degree of compatibility is 1, therefore the constraint is added to the current network. However, move m_9 involves a non-compatible constraint (the degree of compatibility is 0), so it is not added in this case.
- **Setting a value for a node.** Once the user confirms when he/she is going to take the boat from Vigo (move m_7), the node associated to this precise time instant is set to the corresponding value. This operation is allowed because the value is feasible for that node.

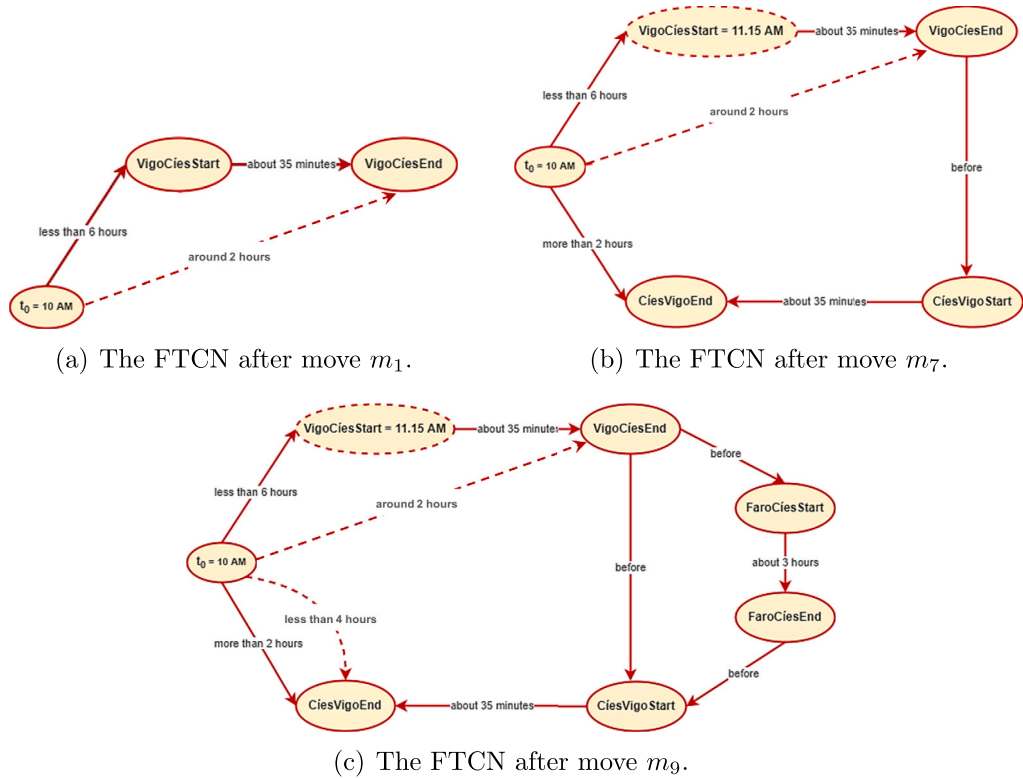


Fig. 4. The FTCN development following the conversation in Table 1. The information that was already present in the global knowledge graph is represented with a continuous line, while dashed lines represent information added on the fly during the conversation between the user and TimeVersa. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

- **Returning feasible values.** The last user message (move m_{13}) does not change the network, but the agent should answer properly to the given question. This requires finding the feasible values for the node associated with the precise time instant when the user could be back in Vigo after the boat trip.

Let us now compare the behavior of the proposed model with that of one of the most popular existing pre-trained language models (i.e., GPT-3) in a similar use case. For this purpose, we try to reproduce, as far as possible,⁴ our use case by means of a conversation with OpenAI API. Namely, we chose the chat task with GPT-3 as a language model with all default parameters and we provided it with the instruction: “The following is a conversation with an AI assistant oriented to book and give information about the activities and the boat trip to go to Cíes Islands. The assistant is helpful, creative, clever, and very friendly. The assistant knows the following information: the boat from Vigo to the Cíes Island lasts about 35 minutes and the hiking route called Faro de Cíes, located in Cíes Islands, takes about 3 hours. There are boats from Vigo to Cíes Islands at 10.00 AM, 11.15 AM, 12.45 PM, 14.45 PM, 16.00 PM and 18.00 PM and there are boats from Cíes Islands to Vigo at 12.00 PM, 13.15 PM, 16.00 PM, 17.15 PM, 18.00 PM and 19.15 PM”.

As it can be appreciated in Table 2, the result is a conversation in which several inconsistencies related with fuzzy temporal reasoning can be observed. The agent recommends the 10 AM boat as a suitable option to arrive at the islands “around 12 PM” (move m_2) and the user engages in a dialogue to inquire an explanation for the given recommendation (move m_3). First, the agent clarifies that “around 12 PM” is defined as some time between 12 PM and 1 PM (move m_6), which does not seem to be in accordance with intuition and this fact is evidence that GPT-3 does not support imprecision properly. Then, move m_{10} shows how the agent is not able to consider and handle the temporal constraints introduced during the conversation and reasoning properly with them. Consequently, the user does not trust the agent’s capabilities and decides to book another boat. Finally, the user adds inconsistent temporal information in move m_{13} (being back in Vigo after doing the hiking route “Faro de Cíes”), but the agent does not detect the inconsistency and answers incorrectly (move m_{14}).

⁴ Notice that the comparison illustrates the usual lack of temporal reasoning in most large language models as mentioned previously. In addition, conversations with OpenAI have some random nature and change dynamically from one to another trial. Even if OpenAI introduced ChatGPT (supported by GPT-3.5) by end of 2022, in a preliminary trial we confirmed that it was not mature enough yet for our purpose. Moreover, we wanted to analyze the use of a language model in conversation but without the assistance of the additional engineering layers which fix undesired behavior in ChatGPT. Accordingly, we carried out the empirical study by December 2022 with Prolific and the OpenAI API (<https://beta.openai.com/playground>).

Table 2

Fragment of a real conversation with the OpenAI API using GPT-3 (i.e., what we call the “OpenAI agent”). Underlined, temporal expressions relevant for the conversation.

Move	Participant	Utterance
m_1	User	Hello! I would like to be in Cíes Islands <u>around 12 PM</u> . What time should I take the boat from Vigo?
m_2	OpenAI agent	You should take the boat <u>at 10 AM</u> .
m_3	User	Why do you think this boat is suitable for me?
m_4	OpenAI agent	The <u>10 AM</u> boat is suitable for you because it will get you to Cíes Islands <u>around 12 PM</u> .
m_5	User	Could you specify what “ <u>around 12 PM</u> ” means for you?
m_6	OpenAI agent	This means that you will likely arrive at the islands sometime <u>between 12 PM and 1 PM</u> .
m_7	User	What is the duration of the boat travel?
m_8	OpenAI agent	The boat from Vigo to the Cíes Island lasts <u>about 35 minutes</u> .
m_9	User	So, why don't you recommend me the <u>11.15 AM</u> boat?
m_{10}	OpenAI agent	The <u>11.15 AM</u> boat would get you to Cíes Islands <u>a little later than 12 PM</u> , so if you're aiming to arrive at the islands <u>around 12 PM</u> the <u>10 AM</u> boat is the better option.
m_{11}	User	I'm not sure about your estimations. Anycase, book the <u>11.15 AM</u> boat for me, please.
m_{12}	OpenAI agent	The <u>11.15 AM</u> boat has been booked for you. Could I help you with anything else?
m_{13}	User	Yes. Once we have reached Cíes Islands, we had thought of doing the hiking route “Faro de Cíes”. Do you think we can be back in Vigo <u>before 2 PM</u> ?
m_{14}	OpenAI agent	Yes, the hiking route “Faro de Cíes” is located in Cíes islands and takes <u>about 3 hours</u> . You should be back in Vigo <u>before 2 PM</u> .

6. Empirical user study

Beyond the illustrative example shown in the previous section, readers may wonder whether the observed lack of a fuzzy temporal reasoning model in a conversational agent may cause a significant negative effect (or not) on the interaction with real users. Accordingly, we set out an empirical study with human evaluators to examine differences in users' perception of the consistency of a conversation with an agent, depending on whether the agent is provided with a fuzzy temporal reasoning model or not.

First, Section 6.1 introduces the experimental setting of the study, including the formal design of the experiment, details about the participants and the procedure. Then, Section 6.2 presents the main results. Finally, Section 6.3 discusses the lessons learned from this study.

6.1. Method

First of all, we aimed to make a comparison of the perceived consistency in an interaction between a user and TimeVersa (which incorporates the temporal reasoning model described in Section 4) and the perceived consistency in an interaction between a user and the OpenAI API with GPT-3 (what we call the “OpenAI agent”). Consequently, we carried out a human evaluation study to validate the following research hypothesis: H_1 : “In case of involving vague temporal information, the interaction with TimeVersa is perceived as more consistent than the interaction with the OpenAI API.” To validate this research hypothesis, we asked participants to rate the level of consistency in several conversations between a conversational agent and a fictitious user.

In addition to studying the impact of incorporating a temporal reasoning model, we paid attention to the extent to which users consider it necessary for this model to implement mechanisms to handle the imprecision of natural language. For this purpose, we considered a crisp version of the temporal reasoning model described in Section 4, which implements the same operations for the particular case where the possibility distributions only take the values 0 or 1. In a more formal way, we set out a second experiment to validate the following research hypothesis: H_2 : “In case of involving vague temporal information, the interaction with TimeVersa incorporating the temporal reasoning model is perceived as more consistent than the interaction with TimeVersa incorporating the crisp version of the temporal reasoning model.”

6.1.1. Experimental design

Based on experimental design theory [59], both experiments in our study were integrated into a single experimental setting.

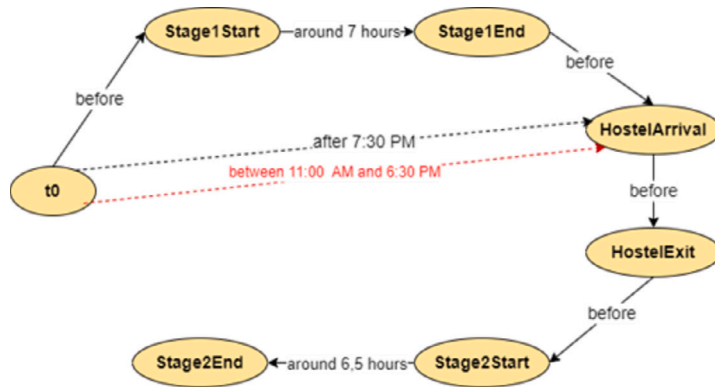
The first experiment follows an experimental design with two factors as independent variables at two levels each (2x2), where “agent” is treated as a between-subject factor (“TimeVersa agent” vs. “OpenAI agent”) and the FTCN Complexity is a within-subject factor (“Simple” vs. “Complex”). Thus, among the four possible combinations, a single participant would therefore only face a trajectory involving both complexities but only an agent as described in the procedure.

The second experiment addressed the need to include fuzzy techniques in the agent's reasoning model, so the independent variable in its experimental design turned into a single two-level between-subjects factor (“TimeVersa agent with fuzzy model” vs. “TimeVersa agent with crisp model”).

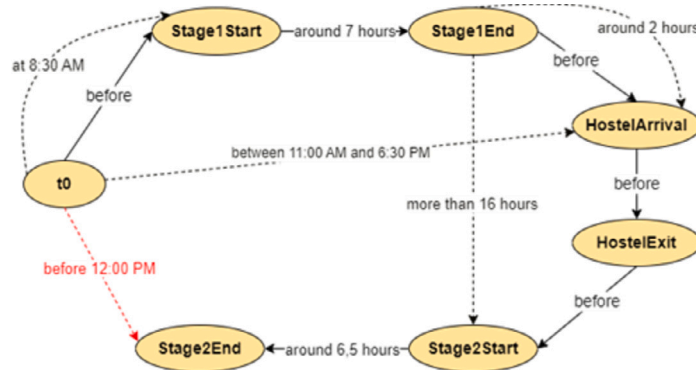
The dependent variable was in both cases the consistency of a conversation which was measured with the following 5-points Likert Scale [60]: “very inconsistent”, “inconsistent”, “somewhat consistent”, “consistent” and “very consistent”.

The use case under consideration was related to the Saint James' way. The user may naturally establish temporal constraints by asking for recommendations on hostels and timings to follow while completing the different stages in the way.

On the one hand, we considered two contexts in which to validate the hypothesis H_1 , giving place to sub-experiments 1.1 and 1.2. Firstly, in experiment 1.1, the resultant FTCN from each conversation is simple enough so that a person can solve it quickly without difficulty (see Fig. 5(a)). In this way, a participant in the experiment can always know whether the agent is responding



(a) Resultant FTCN after the conversation in experiment 1.1.



(b) Resultant FTCN after the conversation in experiment 1.2.

Fig. 5. FTCN associated with each experiment. Information already known to the agent is represented by black continuous lines, while dashed lines represent information added on the fly during the conversations. The red dashed line corresponds to the last constraint added, which is inconsistent with the FTCN. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

consistently or not. Secondly, in experiment 1.2, the resolution of the FTCN is harder and a participant cannot easily perform the consistency check (see Fig. 5(b)). Note that the complexity of the FTCNs is not given by the number of nodes, which is the same, but by the number of arcs between them, which represent temporal constraints between time instants of occurrences of certain events of interest.

On the other hand, only one context is needed to validate the hypothesis H_2 because the model deals with imprecision always in the same way no matter the complexity of the related FTCN. The user, who interacts with the agent in the given conversation, introduces several consecutive temporal constraints that differ by very few time units. The fuzzy model allows the agent to vary its response gradually, while the crisp model changes its response abruptly.

To summarize, the study involved six possible scenarios: four scenarios for the first experiment, i.e., the result of combining the two levels of the independent variable (“TimeVersa agent” versus “OpenAI agent”) with the two levels of FTCN complexity (“Simple” versus “Complex”), and two scenarios for the second experiment, corresponding to the two levels of the independent variable (“TimeVersa agent with fuzzy model” versus “TimeVersa agent with crisp model”).

The participants were exposed to these scenarios (or stimulus) through a screenshot that showed one conversation in each case, without allowing real-time interaction with the agents to avoid the effects of other uncontrolled factors that are not subject of study. After carefully reading each conversation, participants were required to rate the consistency they perceived between the agent’s responses. For each case, if the answer was “inconsistent” or “very inconsistent”, the participant had also to add a justification for this evaluation.

6.1.2. Participants

A total of 130 people took part in the study. They were recruited via Prolific,⁵ an online research platform that provides the recruitment and management of participants for online research. In addition to being of legal age, we established on Prolific that a prerequisite for participation was to be fluent in English, so that participants could understand the conversations shown.

⁵ The empirical study was active in Prolific (<https://www.prolific.com/>) by 22-26 December 2022.

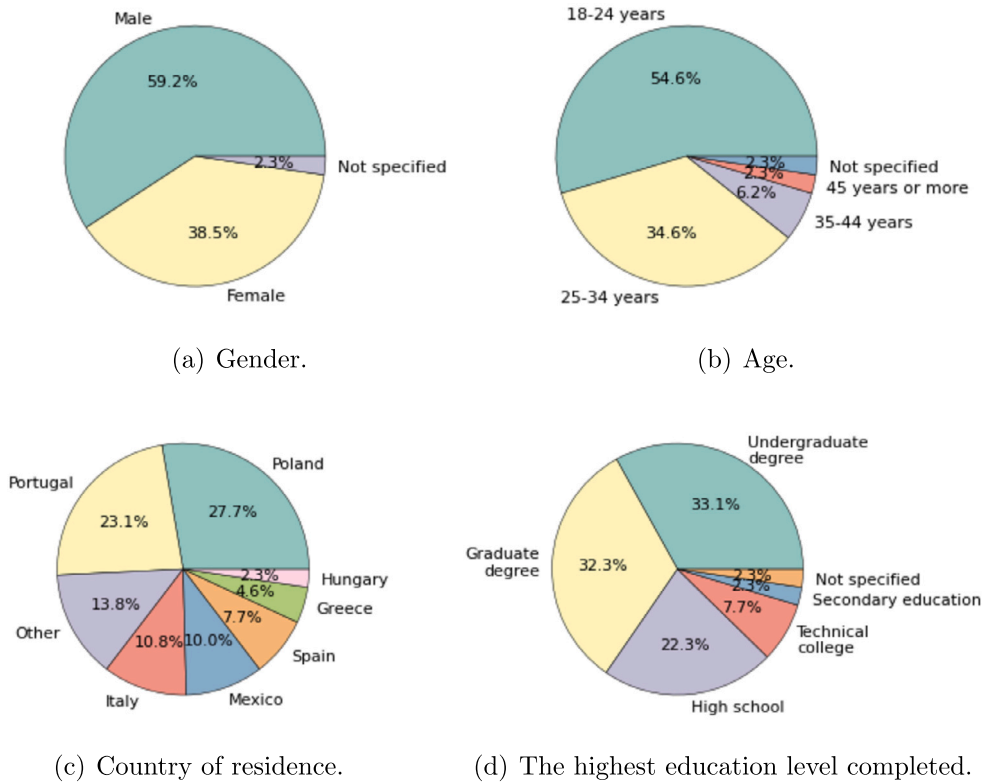


Fig. 6. Socio-demographic data on participants. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Fig. 6 details some socio-demographic characteristics of the sample of participants. Among the participants, almost 90% were under 35 years old, with more men than women. In terms of country of residence, people from at least 17 different countries participated in the study. Finally, almost two thirds of the participants had an undergraduate or graduate degree.

6.1.3. Procedure

Once all the details of the experiment had been defined, the scenarios described were integrated into a survey to collect participants' responses⁶ via Prolific platform. The survey was divided into the following three parts:

- **Informed consent and instructions.** Firstly, the context of the experiment is introduced, followed by an informed consent page which users should adhere to and the instructions for a successful completion, including an example scenario.
- **Scenarios.** The main part of the survey presents three similar consecutive screens corresponding to three of the six possible scenarios described above, based on the level assigned to the participant in accordance with the between-subjects factors. It is important to note that the sequence of appearance of the scenarios was chosen randomly, to reduce the effect of the order of display on participants' responses.
- **Farewell.** Lastly, a completion code was provided to each participant, what let her/him claim payment to Prolific.

6.2. Results

Fig. 7 shows the average consistency perceived in conversation for each experimental setting under consideration.

Reported results for experiment 1 (see Fig. 7(a)) indicate that conversations with TimeVersa are perceived as more consistent than conversations with the OpenAI agent. As for the effects of the FTCN complexity, the difference between the consistency values for each agent is greater when the FTCN is simple (4.35 versus 2.38 in average) than when the FTCN is complex (3.89 versus 3.28 in average).

Regarding experiment 2 (see Fig. 7(b)), we can observe that conversations generated by TimeVersa with the fuzzy temporal reasoning model are perceived as more consistent than those generated by TimeVersa with the crisp version of the model (3.02 versus 2.46 in average).

⁶ <https://tec.citius.usc.es/survey/>.

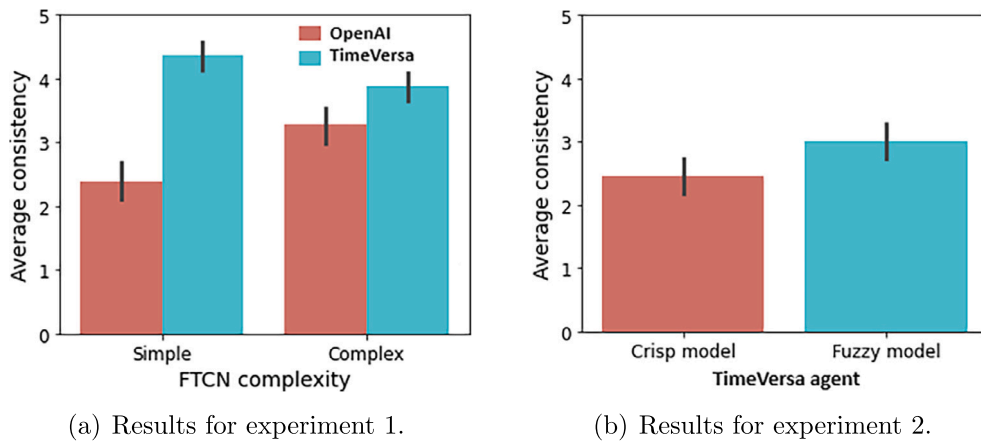


Fig. 7. Effects of the type of conversational agent and the FTCN complexity on consistency perceived in conversations. The error bars represent the 95% confidence interval for the mean. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

With the aim of assessing the goodness and significance of reported results, we carried out an independent-means *t*-test for sub-experiment 1.1 and experiment 2. In addition, in case of the sub-experiment 1.2, we applied a Welch's *t*-test, an alternative to the independent-means *t*-test when there is a violation in the assumption of equality of variances.

On average, the consistency perceived in conversations with the TimeVersa agent was significantly higher than consistency perceived in conversations with the OpenAI agent, no matter if the resultant FTCN after the conversation was simple ($t(128) = 10.43, p < 0.01$) or complex ($t(127.701) = 3.44, p < 0.01$). Furthermore, on average, the consistency perceived in conversations with TimeVersa incorporating the fuzzy temporal reasoning model was significantly higher than consistency perceived in conversations with TimeVersa incorporating the crisp version of the model ($t(128) = 2.64, p < 0.01$).

6.3. Discussion

The results of the study validate the two research hypotheses formulated above and reproduced here:

- H_1 : "In case of involving vague temporal information, the interaction with TimeVersa is perceived as more consistent than the interaction with the OpenAI API."
- H_2 : "In case of involving vague temporal information, the interaction with TimeVersa incorporating the temporal reasoning model is perceived as more consistent than the interaction with TimeVersa incorporating the crisp version of the temporal reasoning model."

On the one hand, we found that users' perception of consistency is significantly higher in conversations with the TimeVersa agent than in conversations with the OpenAI agent. Hence, the fuzzy temporal reasoning model increases the consistency of the conversation, when handling vague temporal information in the context of application. Although the differences are significant for both levels of the FTCN complexity, it is worth noting that in the case of the complex FTCN the average consistency, for both agents, is much closer than for the simple FTCN. In addition, the average consistency for conversations with the OpenAI agent and the complex FTCN exceeds the middle value of the scale considered, even though the agent returns an incorrect answer. It is likely that the complexity of the FTCN has sometimes caused the participant not to be able to reason properly with the temporal constraints involved in the conversation. As a result, this situation can lead to total blind reliability in the agent, even if the agent's response is incorrect.

On the other hand, we found a significantly positive effect of using a fuzzy model compared to using a crisp model on the perception of consistency of a conversation with an agent. However, although the observed differences in the average consistency for both models are statistically significant, they are small in absolute terms. Moreover, both values are surprisingly low, so it seems that the participants do not consider either conversation to be consistent enough. Notice that the highest value for the average consistency in experiment 2, which is obtained for the case that involves the TimeVersa agent with a fuzzy temporal reasoning model, barely exceeds 3, corresponding to "somewhat consistent" on the scale considered. However, in experiment 1 the average consistency reaches a value of 4.35 with TimeVersa and the simple FTCN and a value of 3.89 with TimeVersa and the complex FTCN.

7. Conclusions and future work

Large language models achieve great results in many NLP tasks such as machine translation, summarization, or sentiment analysis. Nevertheless, they have some limitations, including the possibility of returning inconsistent or factually incorrect responses, that make them less appropriate for other applications like conversational agents, since they may be misleading for users. Part of these inconsistencies arise from the lack of temporal reasoning capabilities of these models, which is also a key weakness, since

understanding and reasoning with temporal expressions is very relevant in the context of a conversational agent. In addition, temporal reasoning becomes more difficult to manage for these models when dealing with vague or imprecise information, which play a fundamental role for effective communication.

In this paper we have presented a model for fuzzy temporal reasoning that overcomes some inconsistency issues detected in existing pre-trained language models, in a specific application scenario of a conversational agent. Thanks to the use of an FTCN, this model supports the representation of imprecise temporal knowledge and provides designers with mechanisms to combine the information represented as well as detecting inconsistencies. The temporal model handles temporal constraints on both the time of occurrence of an event (e.g., “around 12 PM”) and the time interval between two events (e.g., “about 3 hours”), which are represented by means of possibility distributions. However, in the context of a conversational agent, we found it difficult to directly transform temporal information from a conversation in natural language into an FTCN. Thus, to provide a more comprehensive and intuitive representation of the application domain, we proposed to start from a knowledge graph, whose nodes represent entities of interest and whose edges can represent any semantic relation between these entities. This knowledge graph must be provided beforehand and validated by an expert. Then, we explained how to map the temporal information stored in a representative knowledge graph of the application domain onto an FTCN that is afterwards in charge of checking consistency and answering queries.

In addition, we have integrated our model into the TimeVersa conversational agent, which represents a proof of concept and operates in a practical use case related to tourism where the agent must deal with temporal information under uncertainty. As a result, TimeVersa can detect temporal inconsistencies and return correct answers for the given queries using the model proposed. In contrast, a similar conversation using the OpenAI API includes incorrect responses in relation to temporal consistency. Indeed, an empirical user study demonstrates that the observed differences in users’ perception of consistency between a conversation with the TimeVersa agent and a similar conversation with the OpenAI API are statistically significant. Therefore, an encouraging result and relevant conclusion of this work is that the model proposed and the prototype implemented entail a starting point to endow conversational agents with fuzzy temporal reasoning capabilities.

Despite this relevant concluding remark, we have identified some limitations in our model, which leave room for further research. As we mentioned in Section 4.2, a relevant step in the procedure for mapping the knowledge graph onto an FTCN involves an exponential computational complexity. It remains as an open problem exploring strengths and weaknesses of approximate search methods in graphs, because integrating an appropriate search method in our model, we may reduce the computational cost of the mapping procedure.

For the sake of generalization, it would be desirable to implement a method that built the knowledge graph with the information of interest automatically, instead of imposing the need of an expert to produce such a graph manually beforehand. Similarly, the projection onto the FTCN should be made automatic too. Namely, one of the hardest steps to make the process fully automatic is the semantic interpretation of possibility distributions which define fuzzy constraints associated to temporal information. Moreover, a future user study should include a greater range of conversations with different use cases, to determine whether the present results generalize well to other situations.

Finally, we envision that our conversational agent will be automatically connected to a pre-trained language model, that may be available online as open source and could run in backend, and then the discourse history of every conversation is automatically monitored by a model for fuzzy temporal reasoning like the one described in this paper.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Mariña Canabal-Juanatey reports financial support was provided by Government of Galicia Department of Culture Education and Universities. Alberto Bugarín-Diz reports financial support was provided by Government of Galicia Department of Culture Education and Universities. Alberto Bugarín-Diz reports was provided by Spain Ministry of Science and Innovation. Jose M. Alonso-Moral reports financial support was provided by Spain Ministry of Science and Innovation.

Data availability

Data will be made available on request.

Acknowledgement

Mariña Canabal-Juanatey is a PhD Researcher supported by the Galician Ministry of Culture, Education, Professional Training and University (ED481A 2022/212). All authors recognize the support of the Galician Ministry of Culture, Education, Professional Training and University (grants ED431G2019/04 and ED431C2022/19). This work is also supported by the Spanish Ministry of Science and Innovation (MCIN/AEI/10.13039/501100011033/) with grants PID2021-123152OB-C21, PID2020-112623GB-I00, and TED2021-130295B-C33. All previous grants are co-funded by the European Regional Development Fund (ERDF/FEDER program).

References

- [1] X. Han, et al., Pre-trained models: past, present and future, *AI Open* 2 (2021) 225–250, <https://doi.org/10.1016/j.aiopen.2021.08.002>.

- [2] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), ACL, Minneapolis, Minnesota, 2019, pp. 4171–4186.
- [3] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: a robustly optimized BERT pretraining approach, arXiv preprint, arXiv:1907.11692, 2019, <https://doi.org/10.48550/arXiv.1907.11692>.
- [4] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: a lite BERT for self-supervised learning of language representations, arXiv preprint, arXiv:1909.11942, 2019, <https://doi.org/10.48550/arXiv.1909.11942>.
- [5] T. Brown, et al., Language models are few-shot learners, Adv. Neural Inf. Process. Syst. 33 (2020) 1877–1901, <https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfbcb4967418bfb8ac142f64a-Paper.pdf>.
- [6] J. Simon, Large language models: a new Moore's law?, <https://huggingface.co/blog/large-language-models>, october 2021. (Accessed 5 September 2022).
- [7] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, ACL, 2020, pp. 7871–7880.
- [8] S. Edunov, A. Baevski, M. Auli, Pre-trained language model representations for language generation, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), ACL, Minneapolis, Minnesota, 2019, pp. 4052–4059.
- [9] N. Ding, S. Hu, W. Zhao, Y. Chen, Z. Liu, H. Zheng, M. Sun, OpenPrompt: an open-source framework for prompt-learning, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL, Dublin, Ireland, 2022, pp. 105–113.
- [10] R. Dale, GPT-3: what's it good for?, Nat. Lang. Eng. 27 (1) (2021) 113–118, <https://doi.org/10.1017/S1351324920000601>.
- [11] L. Floridi, M. Chiriatti, GPT-3: its nature, scope, limits, and consequences, Minds Mach. 30 (2020) 1–14, <https://doi.org/10.1007/s11023-020-09548-1>.
- [12] Y. Zhai, S. Tong, X. Li, M. Cai, Q. Qu, Y.J. Lee, Y. Ma, Investigating the catastrophic forgetting in multimodal large language models, arXiv:2309.10313, 2023.
- [13] R. Dechter, I. Meiri, J. Pearl, Temporal constraint networks, Artif. Intell. 49 (1–3) (1991) 61–95, [https://doi.org/10.1016/0004-3702\(91\)90006-6](https://doi.org/10.1016/0004-3702(91)90006-6).
- [14] S. Barro, R. Marín, J. Mira, A.R. Patón, A model and a language for the fuzzy representation and handling of time, Fuzzy Sets Syst. 61 (2) (1994) 153–175, [https://doi.org/10.1016/0165-0114\(94\)90231-3](https://doi.org/10.1016/0165-0114(94)90231-3).
- [15] D. Dubois, H. Prade, Processing fuzzy temporal knowledge, IEEE Trans. Syst. Man Cybern. 19 (4) (1989) 729–744, <https://doi.org/10.1109/21.35337>.
- [16] L. Vila, L. Godo, On fuzzy temporal constraint networks, Mathw. Soft Comput. 1 (3) (1994) 315–334.
- [17] J. Weizenbaum, ELIZA—a computer program for the study of natural language communication between man and machine, Commun. ACM 9 (1) (1966) 36–45, <https://doi.org/10.1145/365153.365168>.
- [18] K.K. Nirala, N.K. Singh, V.S. Purani, A survey on providing customer and public administration based services using AI: chatbot, Multimed. Tools Appl. 81 (16) (2022) 22215–22246, <https://doi.org/10.1007/s11042-021-11458-y>.
- [19] R.S. Wallace, The anatomy of A.L.L.C.E., in: Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer, Springer Netherlands, Dordrecht, 2009, pp. 181–210.
- [20] L. Yang, J. Hu, M. Qiu, C. Qu, J. Gao, W.B. Croft, X. Liu, Y. Shen, J. Liu, A hybrid retrieval-generation neural conversation model, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Association for Computing Machinery, 2019, pp. 1341–1350.
- [21] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Proceedings of the 27th International Conference on Neural Information Processing Systems, NIPS'14, vol. 2, MIT Press, Cambridge, MA, USA, 2014, pp. 3104–3112, <https://proceedings.neurips.cc/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf>.
- [22] I.V. Serban, et al., A deep reinforcement learning chatbot, arXiv preprint, arXiv:1709.02349, 2017, <https://doi.org/10.48550/arXiv.1709.02349>.
- [23] D. Adiwardana, et al., Towards a human-like open-domain chatbot, arXiv preprint, arXiv:2001.09977, 2020, <https://doi.org/10.48550/arXiv.2001.09977>.
- [24] L. Zhou, J. Gao, D. Li, H.-Y. Shum, The design and implementation of Xiaolce, an empathetic social chatbot, Comput. Linguist. 46 (2020) 1–62, https://doi.org/10.1162/COLI_a_00368.
- [25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, vol. 30, Curran Associates, Inc., 2017, pp. 6000–6010, <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>.
- [26] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., Improving language understanding by generative pre-training, Tech. Rep., OpenAI, 2018.
- [27] OpenAI, GPT-4 technical report, 2023, arXiv:2303.08774.
- [28] A. Akbik, D. Blythe, R. Vollgraf, Contextual string embeddings for sequence labeling, in: Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 1638–1649, <https://aclanthology.org/C18-1139>.
- [29] P. Ke, H. Ji, S. Liu, X. Zhu, M. Huang, SentiLARE: sentiment-aware language representation learning with linguistic knowledge, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, Association for Computational Linguistics, 2020, pp. 6975–6988.
- [30] E.M. Bender, T. Gebru, A. McMillan-Major, S. Shmitchell, On the dangers of stochastic parrots: can language models be too big?, in: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 610–623.
- [31] E. Strubell, A. Ganesh, A. Mccallum, Energy and policy considerations for deep learning in nlp, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 3645–3650.
- [32] M. Zhang, J. Li, A commentary of GPT-3 in MIT technology review 2021, Fundam. Res. 1 (6) (2021) 831–833, <https://doi.org/10.1016/j.fmre.2021.11.011>.
- [33] S. Thukral, K. Kukreja, C. Kavouras, Probing language models for understanding of temporal expressions, in: Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 396–406.
- [34] S. Vashishtha, A. Poliak, Y.K. Lal, B. Van Durme, A.S. White, Temporal reasoning in natural language inference, in: Findings of the Association for Computational Linguistics: EMNLP, Online, ACL, 2020, pp. 4070–4078.
- [35] L. Qin, A. Gupta, S. Upadhyay, L. He, Y. Choi, M. Faruqui, TIMEDIAL: temporal commonsense reasoning in dialog, 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), Online, ACL, 2021, pp. 7066–7076.
- [36] W. Ding, G. Gao, L. Shi, Y. Qu, A pattern-based approach to recognizing time expressions, Proc. AAAI Conf. Artif. Intell. 33 (01) (2019) 6335–6342, <https://doi.org/10.1609/aaai.v33i01.33016335>.
- [37] S. Chen, G. Wang, B.F. Karlsson, Exploring word representations on time expression recognition, Tech. Rep. MSR-TR-2019-46, Microsoft Research, June 2019, <https://www.microsoft.com/en-us/research/publication/exploring-word-representations-on-time-expression-recognition/>.
- [38] L. Derczynski, J. Strötgen, M. Maynard, M.A. Greenwood, M. Jung, GATE-time: extraction of temporal expressions and events, in: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), European Language Resources Association (ELRA), Portorož, Slovenia, 2016, pp. 3702–3708, <https://aclanthology.org/L16-1587>.
- [39] O. Kolomiyets, M.-F. Moens, KUL: recognition and normalization of temporal expressions, in: Proceedings of the 5th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Uppsala, Sweden, 2010, pp. 325–328, <https://aclanthology.org/S10-1072>.
- [40] M. Geva, A. Gupta, J. Berant, Injecting numerical reasoning skills into language models, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, ACL, 2020, pp. 946–958.

- [41] P. Piekos, M. Malinowski, H. Michalewski, Measuring and improving BERT's mathematical abilities by predicting the order of reasoning, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Online, ACL, 2021, pp. 383–394.
- [42] Y. Ma, P.A. Crook, R. Sarikaya, E. Fosler-Lussier, Knowledge graph inference for spoken dialog systems, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 5346–5350.
- [43] S. Yang, R. Zhang, S. Erfani, GraphDialog: integrating graph knowledge into end-to-end task-oriented dialogue systems, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, Association for Computational Linguistics, 2020, pp. 1878–1888.
- [44] C. Xu, M. Nayyeri, F. Alkhoury, H. Shariat Yazdi, J. Lehmann, TeRo: a time-aware knowledge graph embedding via temporal rotation, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 1583–1593.
- [45] S. Liao, S. Liang, Z. Meng, Q. Zhang, Learning dynamic embeddings for temporal knowledge graphs, in: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, WSDM '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 535–543.
- [46] Y. Zhao, X. Wang, J. Chen, Y. Wang, W. Tang, X. He, H. Xie, Time-aware path reasoning on knowledge graph for recommendation, *ACM Trans. Inf. Syst.* (2022), <https://doi.org/10.1145/3531267>.
- [47] T. Jiang, T. Liu, T. Ge, L. Sha, B. Chang, S. Li, Z. Sui, Towards time-aware knowledge graph completion, in: Proceedings of the 26th International Conference on Computational Linguistics (COLING): Technical Papers, Osaka, Japan, 2016, pp. 1715–1724, <https://aclanthology.org/C16-1161>.
- [48] A. Saxena, S. Chakrabarti, P. Talukdar, Question answering over temporal knowledge graphs, 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), Online, ACL, 2021, pp. 6663–6676.
- [49] L. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets Syst.* 100 (1999) 9–34, [https://doi.org/10.1016/S0165-0114\(99\)80004-9](https://doi.org/10.1016/S0165-0114(99)80004-9).
- [50] R. Marín, S. Barro, A. Bosch, J. Mira, Modeling the representation of time from a fuzzy perspective, *Cybern. Syst.* 25 (2) (1994) 217–231, <https://doi.org/10.1080/01969729408902325>.
- [51] L. Zadeh, The concept of a linguistic variable and its application to approximate reasoning—I, *Inf. Sci.* 8 (3) (1975) 199–249, [https://doi.org/10.1016/0020-0255\(75\)90036-5](https://doi.org/10.1016/0020-0255(75)90036-5).
- [52] A. Hogan, et al., *Knowledge Graphs, Synthesis Lectures on Data, Semantics, and Knowledge*, vol. 22, Morgan & Claypool, 2021.
- [53] H. Paulheim, Knowledge graph refinement: a survey of approaches and evaluation methods, *SemanticWeb* 8 (3) (2016) 489–508, <https://doi.org/10.3233/SW-160218>.
- [54] L. Ehrlinger, W. Wöß, Towards a definition of knowledge graphs, in: Proceedings of the 12th International Conference on Semantic Systems (SEMANTICS), vol. CEUR-1695, 2016, <https://ceur-ws.org/Vol-1695/>.
- [55] G. Buchgeher, D. Gabauer, J. Martinez-Gil, L. Ehrlinger, Knowledge graphs in manufacturing and production: a systematic literature review, *IEEE Access* 9 (2021) 55537–55554, <https://doi.org/10.1109/ACCESS.2021.3070395>.
- [56] C.H. Papadimitriou, K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Courier Corporation, 1998.
- [57] T. Al-Moslimi, M. Gallofré Ocaña, A.L. Opdahl, C. Veres, Named entity extraction for knowledge graphs: a literature overview, *IEEE Access* 8 (2020) 32862–32881, <https://doi.org/10.1109/ACCESS.2020.2973928>.
- [58] Z. Chen, Y. Wang, B. Zhao, J. Cheng, X. Zhao, Z. Duan, Knowledge graph completion: a review, *IEEE Access* 8 (2020) 192435–192456, <https://doi.org/10.1109/ACCESS.2020.3030076>.
- [59] A. Field, G. Hole, *How to Design and Report Experiments*, Sage, 2002.
- [60] A. Joshi, S. Kale, S. Chandel, D.K. Pal, Likert scale: explored and explained, *Br. J. Appl. Sci. Technol.* 7 (4) (2015) 396–403, <https://doi.org/10.9734/BJAST/2015/14975>.