EXTRAcT: EXtracting Temper and Relations among Agents from Text — Design and Initial Experiments

Roberto Lazzarini[1], Angelo Ferrando[2][0000−0002−8711−4670], Andrea Gatti[1][0009−0003−0992−4058], and Viviana Mascardi[1][0000−0002−2261−9926]

[1] University of Genoa, Genoa, Italy
4937188@studenti.unige.it andrea.gatti@edu.unige.it viviana.mascardi@unige.it
[2] University of Modena and Reggio Emilia, Modena, Italy
angelo.ferrando@unimore.it

Abstract. Intelligent software agents conceived and implemented in terms of cognitive and emotional traits have been studied for more than thirty years. In the current era of Machine Learning and Agentic Artificial Intelligence, their relevance is increasing considerably.
This renewed interest can be explained by two complementary reasons. On the one hand, cognitive software agents – typically grounded in formal theories and logic-based implementations – can help address the lack of control, safety guarantees, and explainability found in opaque ML-based and Agentic AI systems. On the other hand, agents capable of understanding and expressing emotions provide an effective interface between backend software systems and human users, in all applications where empathy and believability are required.
Among these applications, those employing Virtual and Extended Reality for training, rehabilitation, or entertainment, play a particularly important role.
In our recent work, we addressed the problem of combining cognitive agents, Virtual Reality, and chatbots. The resulting framework, VEsNA, has since been extended with personality traits, giving rise to VEsNA-Pro.
In this paper, we present the design of EXTRAcT, a software framework under development aimed at EXtracting Temper and Relations among Agents from Text.
In our vision, the personality traits extracted by EXTRAcT can be assigned to skeletons of VEsNA agents, while extracted relations can supply VEsNA-Pro code with information capable of driving agent behavior.
The purpose of EXTRAcT is therefore to convert narrative documents (novels, theater plots, science fiction) into a playable representation, in a semi-automatic manner.
While the EXTRAcT stage devoted to extraction of temper and roles from text is still in its design phase, the stage for generating VEsNA-Pro skeletal code has been implemented. To this aim, the ontology where information extracted during Stage 1 is to be stored in order to feed Stage 2 has been simulated, keeping all Stage 1's requirements into account.

Keywords: Virtual Agents, Cognitive Agents, Emotional Agents, Emergent Storytelling, Temper, Propensities, VEsNA, VEsNA-Pro

1 Introduction and Motivation

Intelligent software agents conceptualized and implemented in terms of cognitive [15,37,38,39] and emotional [9,40,48] traits have been studied for more than thirty years. In the Machine Learning and Agentic Artificial Intelligence era, their relevance is witnessing a significant increase, as shown by recent surveys [1,50,51] and interdisciplinary position papers [17].

The reason of this resurged interest lies in the fact that, on the one hand, cognitive software agents — being usually rooted on formal and rigorous theories and on logic-based implementations — may mitigate the lack of control, safety guarantees, and explainability that characterize opaque ML-based and Agentic AI systems [16]. On the other hand, agents that understand and show emotions are a perfect interface between the software system in the backend and the human user in all those applications where empathy and believability are mandatory requisites [14,31,42]. Among these applications, those that exploit Virtual and Extended Reality for training, rehabilitation, or entertainment, play a very important role [5,27,44].

In our recent work we addressed the problem of merging cognitive agents, Virtual Reality and chatbots [19,22,23]. The implemented framework that supports these features, VEsNA , has been extended with personality traits, resulting into VEsNA-Pro [20].

In recent years, the videogame industry has increasingly explored the concept of emergent storytelling, a paradigm in which narratives arise dynamically from the interactions among autonomous agents, players and the surrounding environment, rather than being predetermined by static scripts. Titles such as City 20 [46], developed by Untold Games [47], represent a significant step in this direction. In City 20, each citizen - whether controlled by a human or an artificial agent - exhibits autonomy, reactivity and adaptability, leading to a virtually infinite range of possible storylines. Each simulation unfolds differently as new, unexpected relationships and consequences emerge from these interactions.

Our ultimate goal is to exploit VEsNA-Pro as a framework to develop proofs of concepts of scenarios where emergent storytelling plays a significant role, like in the City 20 videogame. Once the VEsNA-Pro scenario has been set up and experimented, and the concept has been validated, a translation into a program to be managed by a professional videogame engine may start.

Being able to extract elements relevant for a VEsNA-Pro multiagent system — for example, the agents' temper and their relations — from a textual script (or, more ambitiously, from a full novel), would definitely speed up the achievement of that goal.

In this paper we present the design and partial implementation of **EXTRAcT**, a software framework aimed at **EX**tracting **T**emper and **R**elations

among **A**gents from **T**ext. EXTRAcT has been conceived to address the challenges raised by (semi-)automatic extraction of VEsNA-Pro skeletal components from scripts.

The implementation and testing of some of the **EXTRAcT** components is already under way, while other components have just been designed.

The paper is organized in the following way: Section 2 introduces the Belief-Desire-Intentions architecture and VEsNA-Pro, and discusses works aimed at extracting temper and relations from text. Section 3 provides a gentle introduction to EXTRAcT. Section 4 presents the design of the first EXTRAcT stage, namely knowledge graph extraction and ontology population. Section 5 presents the second stage of such pipeline, namely VEsNA-Pro compatible, knowledge graph-driven agent generation. Section 6 reports the results of our initial experiments with the ontology-to-VEsNA component (namely, Stage 2 of the EXTRAcT pipeline). Section 7 concludes and outlines the future directions of this work.

2 Background and Related Work

2.1 The Belief-Desire-Intentions Architecture and AgentSpeak(L)

The Belief-Desire-Intention (BDI) architecture [38] is a cognitive software model that conceptualizes and models autonomous agents using human-like mental states. AgentSpeak(L) [36] is a programming language inspired by Prolog, designed to operationalize the theoretical BDI model into execution. AgentSpeak(L) represents an agent's knowledge through beliefs (similar to Prolog fact) and its behavioural knowledge through declarative plans. Plans are characterized by a triggering events, and applicable only if their context is a logical consequence of the current belief base. When an external or internal event occurs, the AgentSpeak(L) reasoning cycle looks for those agent's plans whose triggering event matches the occurred event and whose context is a logical consequence of the beliefs. Then, it picks one and executes its body.

2.2 VEsNA and VEsNA-Pro

VEsNA is a general-purpose agent-based framework for managing virtual environments through natural-language. It integrates a natural language interface for agent to agent and agent to human interaction (ChatBDI [22,23]), a dynamic virtual environment supporting agents' perception and action (Godot [24] or Unity [45]), and a cognitive framework that enables symbolic reasoning on agents' goals, beliefs, and plans implemented in either Jason [11], an interpreter of AgentSpeak(L), or JaCaMo [10].

While the Jason `Agent` class contains all the methods the agent uses to actually make decisions, the `AgArch` class acts as the reasoning orchestrator. VEsNA extends the default `Agent` class connecting it natively to the body situated in the Virtual Reality (VR) environment and managing all the messages between

Jason mind and VR body. In addition to these extensions, VEsNA provides a set of features like situated artifacts and communication [18,21].

Pro-AgentSpeak(L) [20] extends the AgentSpeak(L) language with the agents' temperament, i.e. those personality traits and attitudes that constitute their propensities. Unlike beliefs, that model the agents' epistemic state, propensities are related with the agents' personality and emotional state. In order for these traits and emotional attitudes to drive deliberation, plans are annotated with weighted propensities and plan selection is guided by compatibility with the agent's current propensities; execution, in turn, updates mutable propensities through post-effects.

VEsNA-Pro extends VEsNA by programming agents with Pro-AgentSpeak(L) instead of AgentSpeak(L): VEsNA-Pro agents are cognitive agents immersed in a Virtual Reality environment, whose behaviour is also driven by their propensities and temper. In the scenarios and experiments we run in our recent research activities, personality traits followed the *OCEAN* (or Big Five) model [25] — where letters in the acronym stay for *Open*, *Conscientious*, *Extrovert*, *Agreeable*, *Neurotics*.

2.3 Extracting Personality Traits from Text

Traditional character trait analysis relied on static lexicon matching or coarse-grained global classification. Recent research emphasizes fine-grained, dynamic understanding and model interpretability.

Moving beyond treating character traits as fixed vectors, Yu et al. [49] established the PersoNet framework to model situated personality understanding. Their work demonstrates that character personality perceptions fluctuate over narrative arcs as models process new events, dialogues, and context shifts.

The emergence of instruction-tuned transformers enabled direct continuous trait profiling. Peters and Matz [32,33] systematically evaluated the capacity of LLMs to measure and infer psychometric traits (such as OCEAN dimensions) from textual behavior, establishing benchmark procedures for persona extraction.

To ensure neural classifiers rely on psychologically sound linguistic cues rather than artifacts, Saeteros et al. [41] applied interpretability metrics (such as Integrated Gradients) to transformer models, confirming that transformer-extracted traits align with established human psychometric models.

2.4 Extracting Character Relationships from Text

The first obstacle to overcome when characters must be related with each other, is to detect them in a correct and unambiguous way. In their 2022 work, Amalvy et al. [2] introduced specialized data augmentation techniques designed to improve character entity detection in literary texts, going beyond the use of Named Entity Resolution (NER) alone. A follow up of that work was published in 2025 [4], where the same authors carried out a study on the impact of low-level natural

language processing tasks on automatic extraction of character networks from literary texts performance. Again, they focused on role of NER and coreference resolution when extracting co-occurrence networks.

Dynamic Character Networks (DCNs) represent a major shift in computational literary analysis by treating character relationships as fluid, time-varying interactions rather than static connections. By replacing simple co-occurrence counts with edge weights that reflect temporal sentiment and interaction polarity, DCNs capture how alliances, conflicts, and social structures evolve across a narrative timeline [3,26,29].

3 EXTRAcT: Introduction and Assumptions

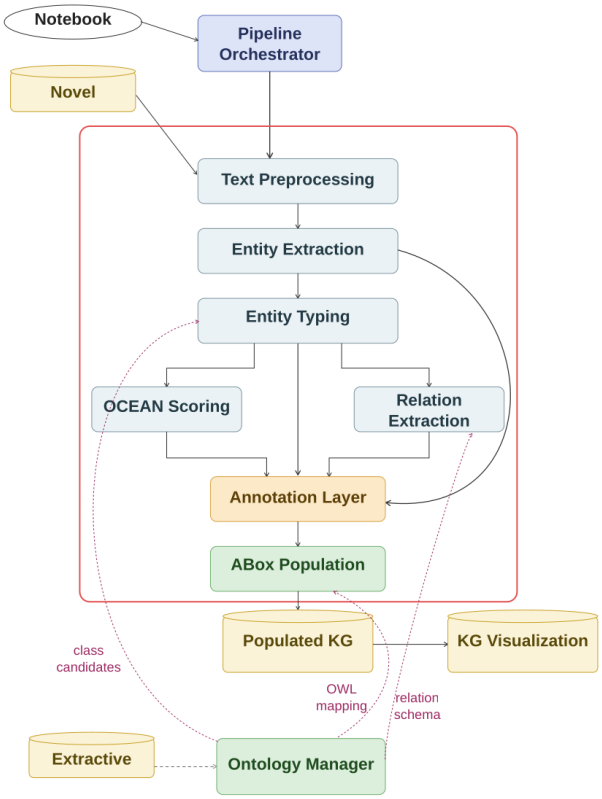

Fig. 1: Envisioned steps of Stage 1.

The EXTRAcT process consists of two stages. The first stage aims at extracting the knowledge graph from text and populating an ontology with information on characters' personality traits and their relations. The second stage acts as

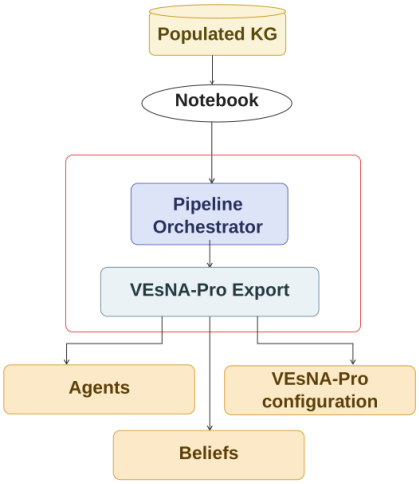

Fig. 2: Implemented steps of Stage 2.

a compilation step: it uses information stored in the populated ontology produced by Stage 1 and converts it into files usable by the VEsNA-Pro framework. Some assumptions on the ontology are needed, to ensure the coherence of the EXTRAcT process.

The structure of Stage 1 is depicted in Figure 1 as a progressive annotation pipeline. An orchestrating component (the **Pipeline Orchestrator**) executes the pipeline components in sequence, making their dependencies explicit. This design exposes the incremental nature of the system, as each component consumes and builds upon the results produced by the one before it. The final component materializes these accumulated annotations into a fully organized and populated ontology.

Stage 2, sketched in Figure 2, has a narrower and more concrete scope than Stage 1. While Stage 1 has been only designed, a working prototype for Stage 2 exists. To test the working prototype, the Ontology that should be produced by Stage 1 has been created manually by the authors, respecting the assumptions on its structure.

This section provides a high-level overview of the two EXTRAcT stages, that are detailed in Sections 4 and 5, and on the assumptions on the ontology that acts as a bridge between them (Section 3.1).

3.1 Assumptions on the Ontology to be Populated

As the text is conceptualized as the narrative material, the ontology is conceptualized as the symbolic vocabulary through which the material can be represented. The ontology acts both as an enabler and as a structural boundary for the extraction process. This leads to the biggest conceptual limit and responsibility of the ontology: both the ontology typing and the relation extraction processes can

only select concepts that already exist in the ontology, thus making it clear that missing ontology concepts cause missing knowledge.

For example, if the ontology has no relation for killing, betrayal or allegiance, the pipeline cannot represent them. The ontology is indeed context-aware and the context must come from its intended use. What is learned are the ontology instances and the properties among them. The ontology schema should already exist.

The GOLEM Ontology for narrative and fiction represents a good starting point [34], being very recent and being available online[3]. GOLEM is too abstract for our purposes, but it may be extended with concepts coming from other research proposals, such as "The ontology of Drama" [13] and the ontological representation of fictional characters proposed in 2026 [12]. Both of them explicitly take emotions and personality traits of characters into account, aligning in their high level purpose to our "ontology-to-be".

To be accepted by the pipeline, we assume that the ontology provides natural-language descriptions for its object properties.

In our approach, taxonomic hierarchies among objects that appear in the textual input will not be represented inside the ontology, that is hence conceived to be almost flat.

Finally, the ontology must provide a class structure or sub-hierarchy related to the concept of 'personality', which is used to determine the entities for which OCEAN scores should be produced.

4 Stage 1, Knowledge Graph Extraction and Ontology Population

As shown in Figure 1, the textual document to be processed, that might for example be a novel, is sent to a **Text Preprocessing** module that will perform text tokenization and chunking.

In our designed pipeline, text tokenization is the grounding artifact of the pipeline: as later annotations become increasingly abstract, they remain anchored to the original tokenization and, therefore, to the source text. Each component's output is post-processed and filtered before being accepted, so that once an annotation is produced it can be treated as a reliable source from which subsequent components can safely read.

Chunking is performed immediately after tokenization. Its purpose is to define temporary processing portions that can be safely handled by neural components without exceeding their context constraints, while keeping the resulting annotations traceable to the original tokens.

All major pipeline components are conceived to share the same architectural pattern: each depends only on previously materialized annotations, applies a zero-shot language-model-based classifier, caches intermediate outputs for resilience against failure, and writes its result back into the shared annotation layer.

[3] `https://ontology.golemlab.eu/`, Accessed on November 10, 2026.

The first component after preprocessing is **Entity Extraction**, which identifies narrative mentions and performs co-reference resolution to group them into clusters of direct or indirect mentions. These mentions should correspond to the same narrative entity, such as characters, locations, or groups. In ABox terms, these clusters provide the basis for introducing named individuals.

The next component is ontology-guided **Entity Typing**. Each previously recognized entity cluster is mapped to a single type from the class taxonomy of the input ontology. Formally, this corresponds to asserting one class membership for each named individual.

Two components consume the annotation provided by entity typing. **OCEAN Scoring** assigns personality trait scores modeled after the OCEAN model, while **Relation Extraction** establishes relations between the previously identified entities. Ontologically, the former is materialized as a series of data property assertions for individuals typed as characters, while the latter is materialized as object property assertions between named individuals.

An **Annotation Layer** will store the accepted outputs produced by the pipeline components and expose them as final annotations. An **Ontology Manager** will handle ontology-specific operations: loading the input TBox ontology, exposing intermediate ontology representations to the pipeline components, and populating the ABox from the annotation layer. Finally, a **Knowledge Graph (KG) Visualization Module** will produce the final graphical representation of the populated ontology as a Knowledge Graph.

The design of Stage 1 is partly inspired by BookNLP [7], an open source project which proved to be an important reference point for processing literary texts. EXTRAcT adopts the same general intuition that long-form narratives should be analyzed incrementally, and it also borrows the concepts of mentions and clusters as a bridge between textual evidence and stable narrative entities. BookNLP also exposed the relevance of LitBank [6,8] as a literary-domain dataset and pointed toward the adoption of an entity extraction model trained on it, *Maverick-mes-litbank* [43]. The conceptual difference between the two projects lies in the role of the annotations: in BookNLP they are a self-standing final output, while in EXTRAcT they are the building blocks of a symbolic structure intended for downstream use.

5 Stage 2, from Ontology to VEsNA-Pro

Stage 2, described in Figure 2, has been implemented in prototypical form.

Its output is organized into three file groups: an individual belief base for each character (**Beliefs** in Figure 2), an individual standardized VEsNA-Pro agent program per character (**Agents** in the picture), and a global declaration file that registers the generated agents within the VEsNA-Pro runtime (**VEsNA-Pro configuration** in the picture).

Given a character, its individual belief base aims to capture its point of view, derived by extracting the object-property assertions in which the character itself is the subject.

Unlike those belief bases, each agent program is generated from the same basic template. Its purpose is not to define complex agent behavior, but to demonstrate that the belief bases extracted from the ontology can be acted upon.

The configuration file registers each character as an agent within the runtime. It is also responsible for translating the OCEAN trait scores into corresponding runtime temperament parameters.

Stage 2 is currently in a more advanced implementation phase than Stage 1.

Stage 2 takes the ontology produced by Stage 1 — created by hand in a way coherent with what we expect from Stage 1 — and generates a VEsNA-compatible initialization package. The ontology is implemented in Turtle[4]. Stage 2's output is organized into a `beliefs/` directory, an `agt/` directory, and a shared `vesna.jcm` configuration file.

For each individual recognized as a character, the **VEsNA-Pro Export** module creates a belief base named `<book>_beliefs_<character>.asl` — where the .asl extension denotes an AgentSpeak(L) file — under `beliefs/`. It also creates a corresponding `<character>_agent.asl` file under `agt/`, which imports the belief base and contains the shared initial goal and AgentSpeak(L) plans. Finally, the exporter adds an agent declaration to `vesna.jcm`, where the character is initialized as a `vesna.VesnaAgent` and its OCEAN scores are encoded as VEsNA `temper` values.

More in details, the `<book>_beliefs_<character>.asl` file is built from the character-centred portion of the populated ontology. Each relevant Turtle assertion is converted into a Jason belief of the form `rdf(subject, predicate, object)`. For a character X, the individualized belief base is reconstructed by reproducing all outgoing graph edges that match `rdf(X, _, _)`. The file also contains the class hierarchy, together with the type and label of the character and of the entities reached through these relations. OCEAN properties are excluded because they are represented separately in the VEsNA-Pro runtime configuration.

The `<character>_agent.asl` file follows the same AgentSpeak(L) template for every character. It imports the corresponding belief file and defines a simple initial goal to demonstrate some minor goal functors to inspect the character's relations. Some of simple goals in the AgentSpeak(L) template are shown in table 1.

Finally, agents are declared in a section of the `vesna.jcm` file, witch associates the character with its generated AgentSpeak(L) program and instantiates it as a `vesna.VesnaAgent`. The exporter reads the five OCEAN properties attached to the character and converts their Stage 1 values from the interval $[0, 100]$ to the interval $[0, 1]$. These normalized values are inserted into the agent's VEsNA `temper`. Once all characters have been processed, their declarations are collected inside the shared `mas vesna` configuration.

[4] `https://www.w3.org/TR/turtle/`, accessed on November 10, 2026.

Goal functor	Description
`!known_targets : self(Self) & data(Self, label, SelfName)`	Returns the entities typed as characters that are connected to the agent through one of its outgoing relations.
`!who_do_you_care_for : self(Self) & data(Self, label, SelfName)`	Returns all entities for which the character has an outgoing `caresfor` belief, expressed as `rdf(Self, caresfor, Object)`.
`!who_do_you_oppose : self(Self) & data(Self, label, SelfName)`	Returns all entities for which the character has an outgoing `opposes` belief. The result is then expressed through a temper-annotated plan.

Table 1: Main front-facing goal functors invoked by the initial `!start` goal.

6 Stage 2, Initial Results

A Turtle ontology has been created from scratch, including some information on characters from the 'Peter and Wendy' novel by British novelist J. M. Barrie. Intentionally, we inserted conflicting information on who is enemy of whom. Indeed, we expect that Stage 1 may output such a conflicting information, given that characters may change their feeling towards other characters, and this may leed to multiple annotations. While this may sound odd in a static setting, it makes lot of sense if we consider the plot dynamics.

So far, we are not dealing with time. In other words, if a character X is in conflict with character Y in some part of the plot, but they become friends later on, we will have both 'friend' and 'enemy' relationships among them.

The correct way of dealing with this issue would be to associate a time interval to such relations, to make them coherent. Spatio-temporal knowledge representation is a vast and challenging research field [28,30,35]. Dealing with time is outside the scope of our current research. While we are aware of the limitations of considering relations static over time, our main objective is to complete the EXTRAcT implementation in its basic form.

Figures 3 and 4 show the runtime output produced by two narrative agents. `Hook` follows the standard AgentSpeak(L) template generated by Stage 2, while `Peter` uses an extended version containing additional goals discussed in the second demo below.

This first demo serves two purposes. First, it shows the basic goal functors described in Table 1 in action. It then how the OCEAN-derived temper profile can affect the plan selected for the same goal. In the present implementation, temper mainly introduces stylistic variation in how the same underlying information is presented. A more complete agent design could use temper annotations to select plans that perform different actions.

Figures 5 and 6 present the second demo, in which the custom agents `Peter` and `Wendy` share a simple interaction. After running some simple self-reporting goals, `Peter` sends `Wendy` a request for characters she considers adversaries.

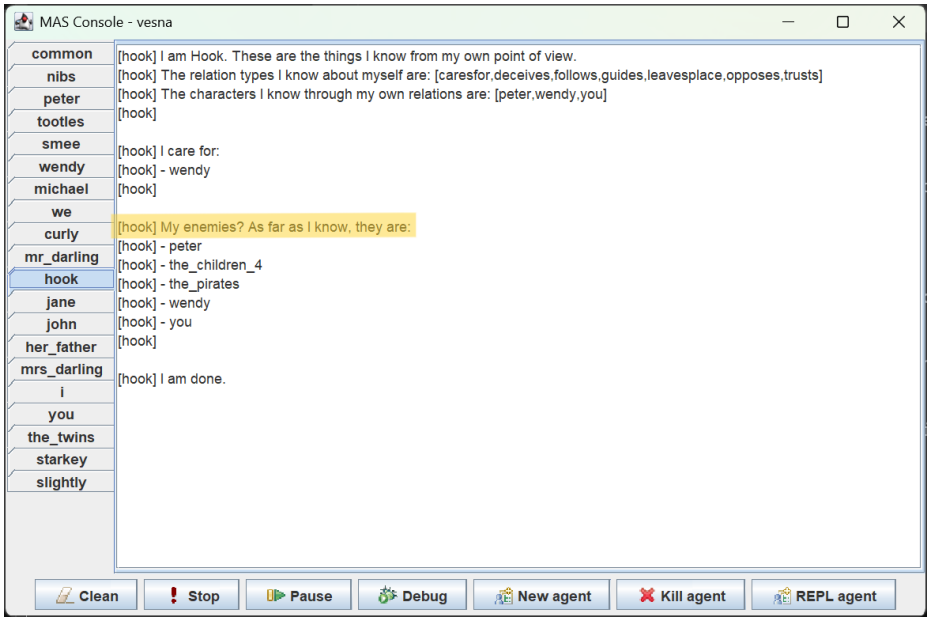

Fig. 3: A simple demo of basic VEsNA agents. In yellow is highlighted a simple example of how temper can influence agent behavior (Hook's point of view).

`Wendy` returns several entities, which `Peter` records as new items in his knowledge base. He then compares `Wendy`'s opponents with his and infers that `Hook` is a common enemy. This example shows that the generated knowledge bases can support communication, belief acquisition, and a simple inference that is not directly stored in either agent's initial output.

7 Conclusions and Future Work

In this paper we presented EXTRAcT, a framework aimed at extracting temper and relations among agents from narrative text, with the ultimate goal of turning novels, theater plots, and scripts into playable multiagent representations compatible with VEsNA-Pro.

We described the two stages that compose EXTRAcT. Stage 1, still at the design phase, is conceived as a progressive annotation pipeline that extracts a knowledge graph from text and populates an ontology with information on characters' personality traits and relations. Stage 2, by contrast, has already been implemented as a working prototype: it compiles a populated ontology into belief bases, AgentSpeak(L) agent programs, and a VEsNA-Pro configuration file, one for each character identified in the input text.

To test the Stage 2 prototype in the absence of a working Stage 1, we manually built a Turtle ontology encoding information about the characters of *Peter*

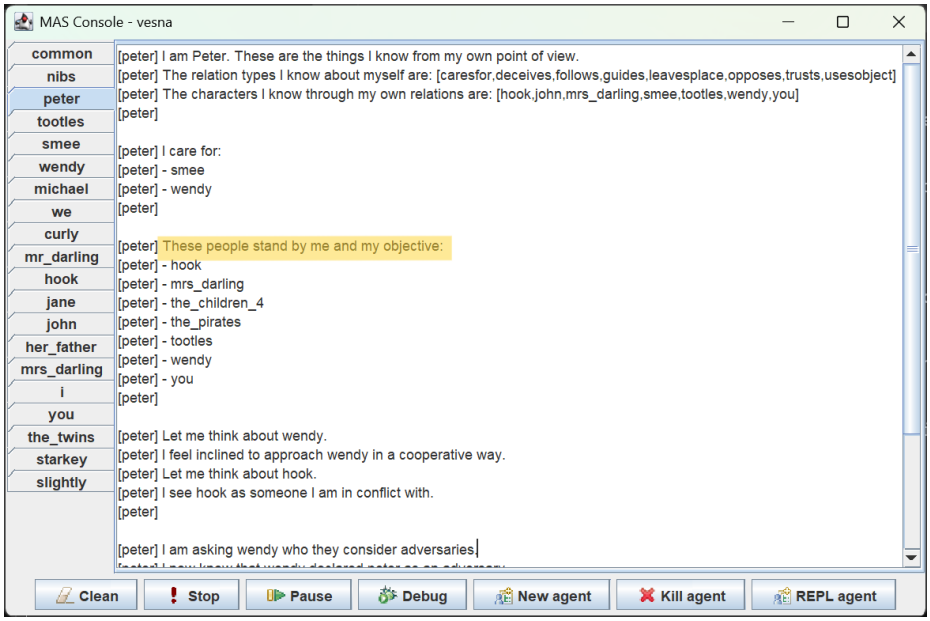

Fig. 4: Another simple demo, with Peter's point of view.

and Wendy, following the structural assumptions that Stage 1 is expected to satisfy. The resulting agents were instantiated and run inside VEsNA, showing that the generated belief bases and AgentSpeak(L) plans support basic self-reporting behavior, inter-agent communication, and simple inference over relations that were not explicitly stored in either agent's initial beliefs, such as the identification of a shared enemy. Albeit very simple, the demos illustrated how OCEAN-derived temper can influence which plan is selected in response to a given goal, even though in the current implementation this influence is limited to stylistic variation rather than a change of the underlying action performed.

These initial results are encouraging, but several limitations remain.

The main one is, clearly, the lack of a fully working implementation. Our short-term efforts are indeed devoted to completing Stage 1 and to designing a large set of experiments involving real written tales, novels, and theater plots.

Because of the story dynamics, we expect that the extracted ontology will often contain (apparently) contradictory information. The experiments will be aimed at understanding under which conditions on the input text, EXTRAcT may produce significant and useful results.

Once these conditions will be understood, we will feed EXTRAcT with texts that meet them. Our long-term goal is to move from short demos toward scenarios that support emergent storytelling comparable to that of *City 20*, where agents derived from a script or novel interact with human players and with each other in a Virtual Reality environment. Achieving this objective would validate

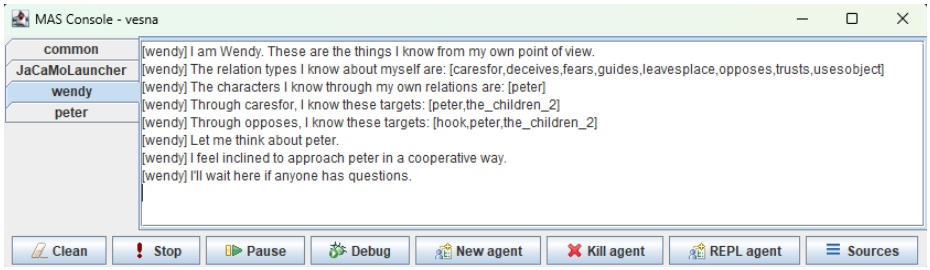

Fig. 5: A slightly more advanced demo, where two custom agent share a simple interaction. In yellow is highlighted a simple example of information exchange and inference (Wendy's point of view).

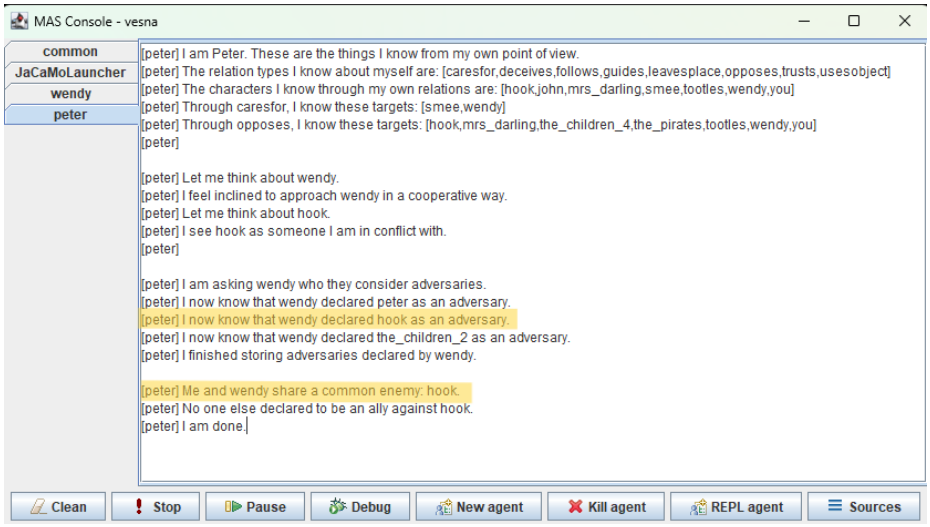

Fig. 6: Another more advanced demo, with Peter's point of view.

EXTRAcT not only as an extraction framework, but as a bridge between narrative material and interactive, agent-driven virtual worlds.

Disclaimer on the Use of Generative AI

EXTRAcT has been envisioned, designed, and implemented by the authors as part of their work on cognitive and emotional agents, and their implementation in VEsNA-Pro. Generative AI tools (Claude, in particular) have been only used for polishing and formatting the text.

References

1. Agiollo, A., Omicini, A.: Integrating machine learning into Belief-Desire-Intention agents: Current advances and open challenges (2025), https://arxiv.org/abs/2510.20641
2. Amalvy, A., Labatut, V., Dufour, R.: Data augmentation for robust character detection in fantasy novels. In: Rochat, Y., Métrailler, C., Piotrowski, M. (eds.) Proceedings of the Workshop on Computational Methods in the Humanities 2022, Lausanne, Switzerland, June 9-10, 2022. CEUR Workshop Proceedings, vol. 3602, pp. 23–32. CEUR-WS.org (2022), https://ceur-ws.org/Vol-3602/paper2.pdf
3. Amalvy, A., Labatut, V., Dufour, R.: Renard: A modular pipeline for extracting character networks from narrative texts. CoRR **abs/2407.02284** (2024). https://doi.org/10.48550/ARXIV.2407.02284
4. Amalvy, A., Labatut, V., Dufour, R.: The role of natural language processing tasks in automatic literary character network construction. In: Rambow, O., Wanner, L., Apidianaki, M., Al-Khalifa, H., Eugenio, B.D., Schockaert, S. (eds.) Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025. pp. 8462–8473. Association for Computational Linguistics (2025), https://aclanthology.org/2025.coling-main.566/
5. Aruanno, B., Bordegoni, M., Carulli, M., Colombo, G., Rossoni, M., Kwok, T.H., Lim, T., Medellin-Castillo, H.I., Ramani, K., Regazzoni, D., et al.: Extended reality in industry and healthcare: Current trends and future perspectives. Journal of Computing and Information Science in Engineering **25**(12), 120806 (2025). https://doi.org/https://doi.org/10.1115/1.4070203
6. Bamman, D.: Born literary natural language processing. In: Tilton, L., Mimno, D., Johnson, J.M. (eds.) Computational Humanities. Debates in the Digital Humanities, University of Minnesota Press, Minneapolis, Minnesota (2024), https://dhdebates.gc.cuny.edu/read/computational-humanities-5c64bbab-d7ca-41be-8f87-f26117a9a20f/section/7c98d16f-ef21-4201-ad00-24b4e3ba6be1
7. Bamman, D.: BookNLP: A natural language processing pipeline for books. https://github.com/booknlp/booknlp (2026), Software repository, Accessed on November 10, 2026
8. Bamman, D., Lewke, O., Mansoor, A.: An annotated dataset of coreference in English literature. In: Proceedings of the Twelfth Language Resources and Evaluation Conference. pp. 44–54. European Language Resources Association, Marseille, France (May 2020), https://aclanthology.org/2020.lrec-1.6/
9. Bates, J.: The role of emotion in believable agents. Commun. ACM **37**(7), 122–125 (1994). https://doi.org/10.1145/176789.176803
10. Boissier, O., Bordini, R.H., Hübner, J.F., Ricci, A., Santi, A.: Multi-agent oriented programming with JaCaMo. Science of Computer Programming **78**(6), 747–761 (Jun 2013). https://doi.org/10.1016/j.scico.2011.10.004
11. Bordini, R., Hübner, J., Wooldridge, M.: Programming Multi-Agent Systems in AgentSpeak Using Jason, vol. 8. John Wiley & Sons, Ltd, United Kingdom (10 2007). https://doi.org/10.1002/9780470061848
12. Bourgois, A., Barré, J., Seminck, O., Poibeau, T.: Toward an ontological representation of fictional characters. Computational Humanities Research **2**, e6 (2026). https://doi.org/10.1017/chr.2026.10025
13. Damiano, R., Lombardo, V., Pizzo, A.: The ontology of drama. Appl. Ontology **14**(1), 79–118 (2019). https://doi.org/10.3233/AO-190204, https://doi.org/10.3233/AO-190204

14. Dell'Acqua, P., Costantini, S.: Empathetic human-agent interaction via emotional behavior trees. Intelligenza Artificiale **17**(1), 89–100 (2023). https://doi.org/10.3233/IA-230014
15. Dennett, D.C.: The Intentional Stance. MIT Press (1987)
16. Ferrando, A., Briola, D., Collier, R.W., Mascardi, V.: Agency and generation: Friends or enemies? In: EUMAS (1). Lecture Notes in Computer Science, vol. 16258, pp. 142–162. Springer (2025). https://doi.org/10.1007/978-3-032-22817-8_9
17. Ferrando, A., Gatti, A., Mascardi, V.: MEDiTATe: a first step of a journey from BDI to neuroscience, and back. In: EMAS. Lecture Notes in Computer Science, vol. 16407, pp. 117–140. Springer (2025). https://doi.org/10.1007/978-3-032-18011-7_8
18. Ferrando, A., Gatti, A., Mascardi, V.: Oops, I heard that! Situated communication with locality-aware KQML. In: Rodriguez, S., Feng, L., Müller, J.P. (eds.) Engineering Multi-Agent Systems - 13th International Workshop, EMAS 2025, Detroit, MI, USA, May 19-20, 2025, Revised Selected Papers. Lecture Notes in Computer Science, vol. 16407, pp. 157–176. Springer (2025). https://doi.org/10.1007/978-3-032-18011-7_10
19. Ferrando, A., Gatti, A., Mascardi, V.: Integrating Virtual Reality, Chatbots, and BDI Agents: VEsNA Goes Fast!, pp. 289–322. Springer Nature Switzerland, Cham (2026). https://doi.org/10.1007/978-3-032-01082-7_11
20. Gatti, A., Casale, F., Mascardi, V., Stucchi, A., Ferrando, A.: VEsNA-Pro: Exploiting BDI agents with propensities for emergent narrative. In: Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems. p. 967–976. AAMAS '26, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2026). https://doi.org/10.65109/PDPF4024
21. Gatti, A., Ferrando, A., Mascardi, V.: Situated agents in action: Extending VEsNA with spatial and grabbable artifacts. In: PAAMS. Lecture Notes in Computer Science, vol. 16031, pp. 348–353. Springer (2025). https://doi.org/10.1007/978-3-032-07638-0_29
22. Gatti, A., Mascardi, V., Ferrando, A.: ChatBDI: Think BDI, talk LLM. In: AAMAS. pp. 2541–2543. International Foundation for Autonomous Agents and Multiagent Systems / ACM (2025), https://dl.acm.org/doi/10.5555/3709347.3743930
23. Gatti, A., Mascardi, V., Ferrando, A.: Let me talk to you! Natural language interaction between humans and BDI agents via ChatBDI. In: ECAI. Frontiers in Artificial Intelligence and Applications, vol. 413, pp. 3646 – 3654. IOS Press (2025). https://doi.org/10.3233/FAIA251242
24. Godot foundation: Godot web site (2026), https://godotengine.org/, Accessed on November 10, 2026
25. Goldberg, L.: An alternative "description of personality": The big-five factor structure. Journal of Personality and Social Psychology **59**, 1216–1229 (12 1990). https://doi.org/10.1037/0022-3514.59.6.1216
26. Jones, P., Quinn, E., Koskinen, J.H.: Measuring centrality in film narratives using dynamic character interaction networks. Soc. Networks **63**, 21–37 (2020). https://doi.org/10.1016/J.SOCNET.2020.03.003
27. Lampropoulos, G., Anastasiadis, T., Garzón, J.: Augmented reality and virtual reality in exergaming. Future Internet **17**(8), 332 (2025). https://doi.org/10.3390/fi17080332

28. Mantle, M., Batsakis, S., Antoniou, G.: Large scale distributed spatio-temporal reasoning using real-world knowledge graphs. Knowl. Based Syst. **163**, 214–226 (2019). https://doi.org/10.1016/J.KNOSYS.2018.08.035
29. Michel, G., Epure, E.V., Hennequin, R., Cerisara, C., Lapata, M.: Graphlit: Learning text-enriched dynamic character network representations for literary study. CoRR **abs/2605.28643** (2026). https://doi.org/10.48550/ARXIV.2605.28643
30. Nipurakis, T., Chatzinikolaou, S., Vassiliou, G., Papadakis, N.: Spatial and temporal knowledge representation: Ontological foundations, semantic web standards. Electronics **15**(8) (2026). https://doi.org/10.3390/electronics15081590
31. Ortega-Ochoa, E., Arguedas, M., Daradoumis, T.: Empathic pedagogical conversational agents: a systematic literature review. British Journal of Educational Technology **55**(3), 886–909 (2024). https://doi.org/10.1111/bjet.13413
32. Peters, H., Cerf, M., Matz, S.C.: Large language models can infer personality from free-form user interactions. CoRR **abs/2405.13052** (2024). https://doi.org/10.48550/ARXIV.2405.13052
33. Peters, H., Matz, S.C.: Large language models can infer psychological dispositions of social media users. PNAS Nexus **3**(6) (06 2024). https://doi.org/10.1093/pnasnexus/pgae231
34. Pianzola, F., Cheng, L., Pannach, F., Yang, X., Scotti, L.: The GOLEM ontology for narrative and fiction. Humanities **14**(10) (2025). https://doi.org/10.3390/h14100193, https://www.mdpi.com/2076-0787/14/10/193
35. Plamper, P., Köpcke, H., Groß, A.: A survey on spatio-temporal knowledge graph models. CoRR **abs/2512.16487** (2025). https://doi.org/10.48550/ARXIV.2512.16487
36. Rao, A.S.: AgentSpeak(L): BDI agents speak out in a logical computable language. In: 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Eindhoven, The Netherlands, January 22-25, 1996. Lecture Notes in Computer Science, vol. 1038, pp. 42–55. Springer (1996). https://doi.org/10.1007/BFb0031845
37. Rao, A.S., Georgeff, M.P.: Deliberation and its role in the formation of intentions. In: UAI. pp. 300–307. Morgan Kaufmann (1991). https://doi.org/10.1016/B978-1-55860-203-8.50043-7
38. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a BDI-architecture. In: KR. pp. 473–484. Morgan Kaufmann (1991), https://dl.acm.org/doi/10.5555/3087158.3087205
39. Rao, A.S., Georgeff, M.P.: BDI agents: From theory to practice. In: Proceedings of the First International Conference on Multiagent Systems, June 12-14, 1995, San Francisco, California, USA. pp. 312–319. The MIT Press (1995)
40. Reilly, W.S.N.: Believable Social and Emotional Agents. Technical report, Carnegie Mellon University, Pittsburgh, PA (May 1996)
41. Saeteros, D., Gallardo-Pujol, D., Ortiz-Martínez, D.: Text speaks louder: Insights into personality from natural language processing. PLOS ONE **20**(6), 1–32 (06 2025). https://doi.org/10.1371/journal.pone.0323096
42. Sanjeewa, R., Iyer, R., Apputhurai, P., Wickramasinghe, N., Meyer, D.: Empathic conversational agent platform designs and their evaluation in the context of mental health: systematic review. JMIR Mental Health **11**, e58974 (2024). https://doi.org/10.2196/58974
43. SapienzaNLP: Maverick mes LitBank. https://huggingface.co/sapienzanlp/maverick-mes-litbank (2024), Hugging Face model card. Official weights for Maverick-mes trained on LitBank and based on DeBERTa-large. Accessed on November 10, 2026

44. Shatokhin, O., Dzedzickis, A., Pečiulienė, M., Bučinskas, V.: Extended reality: types and applications. Applied Sciences **15**(6), 3282 (2025). https://doi.org/10.3390/app15063282
45. Unity technologies: Unity web site (2026), https://unity.com/, Accessed on November 10, 2026
46. Untold Games: City 20 press news. https://lulop.com/it_IT/post/show/290340/untold-games-parametric-urban-.html, Accessed on November 10, 2026
47. Untold Games: Untold games official website. https://untoldgames.com, Accessed on November 10, 2026
48. Velásquez, J.D.: Modeling emotions and other motivations in synthetic agents. In: AAAI/IAAI. pp. 10–15. AAAI Press / The MIT Press (1997), https://dl.acm.org/doi/10.5555/1867406.1867408
49. Yu, M., Li, J., Yao, S., Pang, W., Zhou, X., Zhou, X., Meng, F., Zhou, J.: Personality understanding of fictional characters during book reading. In: Rogers, A., Boyd-Graber, J.L., Okazaki, N. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023. pp. 14784–14802. Association for Computational Linguistics (2023). https://doi.org/10.18653/V1/2023.ACL-LONG.826
50. Zall, R., Kangavari, M.R.: Comparative analytical survey on cognitive agents with emotional intelligence. Cognitive Computation **14**(4), 1223–1246 (2022). https://doi.org/10.1007/s12559-022-10007-5
51. Zall, R., Kheyrkhah, A., Cambria, E., Naseri, Z., Kangavari, M.R.: Intelligent agents with emotional intelligence: Current trends, challenges, and future prospects (2026), https://arxiv.org/abs/2511.20657