Towards Convention-Based Game Strategies

Shuxian Pan[1,2][0000−0001−8984−6149] and Carles Sierra[1][0000−0003−0839−6233]

[1] Artificial Intelligence Research Institute (IIIA-CSIC), Campus de la UAB, Bellaterra, 08193, Barcelona, Spain
[2] School of Computing Technologies, RMIT University, Melbourne, Victoria 3001, Australia
{pshuxian,sierra}@iiia.csic.es
https://www.iiia.csic.es

Abstract. To effectively develop cooperative multiagent systems, we introduce an architecture that facilitates the agents' dynamic adoption of conventions. It expands an existing agent model's action selection architecture with a component that uses Natural Language Processing techniques. This component embeds conventions into agent interaction strategies to improve the predictability of other agents' actions if all agents adopt the same conventions in their strategies.

Keywords: Norm extraction · Multiagent systems · Cooperative AI.

1 Introduction

Conventions can be defined as recurrent behaviour patterns of human communities [2] that increase the predictability of interaction outcomes. In an AI context, conventions can coordinate agents' actions in multiagent systems and simplify the agents' decision-making machinery. In general, not all agents are necessarily willing or capable of adhering to the same conventions. However, in a cooperative multiagent system, we may assume that the agents will agree to adhere to the same conventions to improve their collective performance.

In the literature, different terms refer to these agent behaviour patterns, often used to determine whether a specific action is "correct" and sometimes represent typical behaviour in a society. The term *convention* is often related to patterns that result from an agreement among members of a given community or culture. The term *norm* is often associated with legal aspects of behaviour and contains rewards or sanctions. The general term *rule of behaviour* is also commonly used. In this paper, we will use the term *convention*, but our proposal could be applied to norms or rules, as they all share the same basic structure. If we consider the concepts of rules and conventions, the boundary between them in real life can be pretty vague since the conventions can easily be settled as rules with the agreement between the agents.

Previous research has demonstrated that conventions considerably enhance a multiagent system's overall performance. Conventions may have either an external or an internal origin. Machine learning methods, such as Reinforcement

Learning, have identified particular agents' communication patterns as *intrinsic* conventions [14, 20, 22, 40]. *Extrinsic* human social conventions are quite often imposed over human communities or multiagent systems [18, 23, 24, 43][3]. Conventions are frequently hand-coded in those previous works, representing a costly effort for engineers. Any modification to the system's conventions often requires manual checks of their soundness. Although previous work has introduced conventions into multiagent systems in an automated style [35], to our knowledge, no attempt has been made to develop an automatic NLP pipeline to embed conventions into a system. This paper introduces an architecture based upon [33] primarily concerned with processing natural language conventions. We believe it will facilitate the creation of convention-based multiagent systems and give users control over multiagent behaviour. To illustrate the components of the architecture, we will use the board game Hanabi. Nonetheless, we will also attempt to present the architecture, particularly the NLP component, in as general terms as possible. Our vision is to "program" agents by *declaring* in natural language the game rules and the strategic behaviour that the agent should show.

In short, the contributions of this paper are: 1) to propose a generalisation of the decision-making component of an existing architecture [33], 2) to discuss the use of an NLP pipeline for norm extraction, and 3) to explore the combination of ontologies and convention patterns to represent conventions formally. This paper is divided into four sections. Section 2 describes the research problem and our focus, Section 3 introduces the relevant background knowledge, Section 4 presents our preliminary proposal for the architecture, and Section 5 discusses future work. This paper is submitted as a short paper presenting ongoing work.

2 Problem

Our research objective is to study "how agents adapt their strategies to conventions in a cooperative multiagent system." To achieve this objective, we need to design an architecture with the following aspects/requirements:

- **Conventions**. Conventions are usually expressed in natural language and can be represented in a logical formalism [9, 16, 17]. Although conventions often lack the sanctioning aspect of (legal) norms, their structure is similar: both impose constraints on human or agent behaviour [1]. Thus, in our architecture, we focus on using NLP techniques for *norm extraction* (see Section 3.1) to process the conventions. We aim to automatically translate natural language conventions into a machine-readable representation for our agent model. Section 4.1 will outline this mechanism.
- **Knowledge Representation**. Agents must have a *model of the environment* to observe the actions of others and their consequences. To achieve

[3] Some of these works focus on *policies* rather than conventions. These two concepts are similar, although policies sometimes have a more probabilistic flavour [19]: there is the option that an agent probabilistically chooses an alternative to the action recommended by the policy when *exploring* the space.

their common goal, agents must also be able to understand, that is, find explanations for, the actions made by other agents. Hence, our agent model must include some *Theory of Mind* (ToM) representation [33] (see Section 3.2).
- **Reasoning**. The architecture of [33] includes a component to determine the agent's next action. This component contains a set of conventions to select an action from among several possibilities. We follow the same path in our architecture with some adaptations (see Section 4.4). For instance, decision rules in [33] contain priorities expressed with natural numbers. The higher the number, the higher the priority. Since [33] only shows examples requiring a few simple conventions, it is enough to make them hand-coded. However, for larger sets of conventions, possibly more complex, we will need (semi-)automated approaches for priority determination.

We will use the game Hanabi as a testbed. Hanabi is a cooperative board game for two to five players that will serve as an illustrative example for this paper. The game's goal is to build card stacks in a specific order. There are five distinct colour stacks, each containing up to five cards that must be played in order from 1 to 5; the more cards correctly played, the better the final score. The players cannot view their cards, only those of their fellow players. The game actions are: providing a hint on a card held by another player (so-called "clue a card", saying its colour or number), discarding a card, and playing a card. There is a series of conventions that complement these rules. For instance, H-Group Conventions[4] are conventions organised and published by Hanabi players. Table 1 shows some of them. This paper will mainly use the two conventions labelled "Chop" in Table 1 as examples. See the following illustration of the use of a convention.
Example: Alice has no clued cards. Bob has cards in the first, third and fifth slots clued. Considering only the game rules, players could play or discard any card. However, when following the "Chop" conventions, Alice *should* discard her *chop* card in her fifth slot, and Bob should discard the card in his fourth slot.

Table 1. Some conventions extracted from the H-Group Conventions.

Labels	Conventions
Chop	The right-most unclued card in a player's hand is called their "chop" card. When a player needs to discard, they should discard their chop card.
Types of Clues	Players are only allowed to give two types of clues: a Play Clue (meaning to play the focused card) and a Save Clue (meaning to save the focused card for later).
Play Clues	Play Clues can be given with either a colour or number clue.
Save Clues	Save Clues can only be given to chop cards.

[4] https://hanabi.github.io

3 Background

3.1 Norm Extraction

This paper's primary focus is on conventions. Hence, how to process conventions is the critical part of the architecture we want to discuss in detail in this paper. We will adapt existing state-of-the-art norm extraction techniques. Norm extraction is a sub-task of natural language processing that involves recognising and extracting norm structures from natural language text using (semi-)automatic approaches. Most past research has been conducted in the context of norm extraction from legal documents. Even though the definitions of norms and conventions are slightly different, from an NLP perspective, the differences are such that we can apply legal norm extraction techniques to conventions. However, we know that certain procedures were developed to address specific concerns of legal norms that may not be needed to process conventions. Unlike legal texts, the structure and semantics of conventions, particularly those for Hanabi, can be pretty simple and limited.

Recent norm extraction techniques and a general overview were evaluated in [13]. Norm extraction, like other NLP tasks, usually begins with text preprocessing. Several existing NLP toolkits and pipelines (e.g. NLTK[5], or Stanza[6]) provide automated preprocessing techniques, including tokenisation, removal of stop words and punctuation, and lemmatisation. In addition, some particular syntactic structures of the input text need to be modified, such as lists of items with enumerations, colons and numerous references which contains punctuation and alpha-numeric characters (prevalent in legal texts), to avoid failure of the sentence processing [46]. After this preprocessing, subsequent steps consist of parsing and/or tagging, the standard techniques in NLP. Some studies applied pre-trained general parsers, such as the Stanford parser[7] [11, 41, 46], or the Berkeley parser[8] [41]. Based on various grammar systems, parsers generate the grammar tree structure of the sentence over words (dependency) or phrases (constituency). Conversely, Tagging is the annotation process that can attach syntactic, semantic, or logical features to words. The tagging and parsing processes can be performed simultaneously by the same tool or by tagging before parsing. For instance, we can automatically annotate words with part-of-speech (POS) tags before the parsing process starts. Some specific annotations, such as deontic information of legal norms, can only be done manually. Additional knowledge sources, such as Word Net [11] and Wikipedia [25, 37], were explored to supplement the semantic representation. In [41], the method was more sophisticated, containing a task-specific dictionary and vectorisation of sentences. Once annotated data is collected, machine learning or symbolic methodologies can incorporate norms into an AI system. One example is using the tax code as training data for a complicated multi-layer convolutional neural network (CNN)

[5] https://www.nltk.org
[6] https://stanfordnlp.github.io/stanza/index.html
[7] https://nlp.stanford.edu/software/lex-parser.shtml
[8] https://github.com/slavpetrov/berkeleyparser

to classify sentences into several deontic categories [32]. Another example is training a classifier based on the syntactic/semantic features in the norm sentences to extract specific elements from them [15]. Symbolic applications include NL2KR [16] and the Candc and Boxer tool chain [11]. Although their processes are not identical, their primary goal is to produce formal representations of input sentences adopting Combinatory Categorial Grammar (CCG). However, neither tool is maintained, so users should anticipate compatibility issues when applying them. As for the evaluation, although [13] created the gold standard for semantic parsing in the legal domain, the size of this test set is limited. Like the other test sets for legal norm extraction, they have to be annotated by experts. For metrics, recall, precision, and accuracy were commonly used [6, 12, 36, 41].

3.2 Theory of Mind

Theory of Mind (ToM) is an ability acquired through social interaction. To comprehend others' actions, humans need to create models of the beliefs of others. This ability can be further nested. For instance, not just the beliefs that agent i holds about the beliefs of j, but also the beliefs that agent i holds about the beliefs of j about the beliefs of k and so on. The former example is a first-order ToM statement, and the latter a second-order ToM statement. Although a higher order implies a deeper degree of comprehension, the rise in complexity will usually offset any gains [44, 45], so it is vital to consider and control the depth. [33] offers a thorough introduction to the previous research. [10] introduced the (potential) application of ToM in AI but also indicated that many existing approaches neglected or over-simplified the mental states of agents, which is critical for the human mind and their mental process. In [33], ToM focuses on deriving explicit beliefs, so the mental states are not involved. For our architecture, it will be the same. Our agents require this capacity to predict the actions of others and act upon that prediction.

3.3 Hanabi

Hanabi has been proposed as a challenging game[9] to explore the limits of machine learning or rule-based systems [7, 21, 30, 40, 42, 43]. For example, a new gameplay setting named *other-play* [42] (implemented from [22]), or an adversarial mechanism to self-play [43]. In these works, the agents either followed *different* conventions or played with human players. We do not discuss them in this article since, as stated previously, we focus our work on agents that play together and adhere to the same conventions. Thus, there is no need to address potential convention conflicts.

4 Architecture

Figure 1 illustrates the complete architecture we propose. [33] serves as the inspiration. In our approach, a cooperative agent receives as input: information

[9] A detailed literature review is provided by [4].

about the environment, messages from other agents, and a list of conventions to be followed when making decisions. A series of modules process the input so the action selection component can determine the action to take. These modules are divided into two blocks: "NLP" and "Agent Decision-making". The following subsections describe the modules in detail.

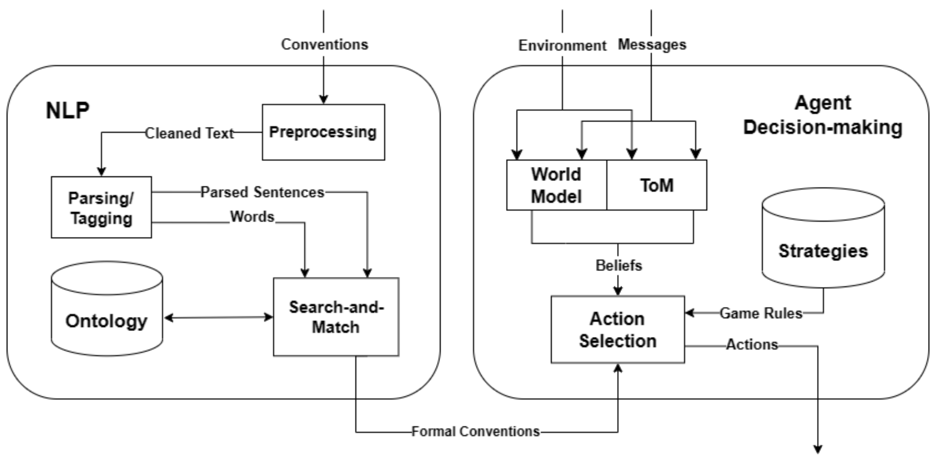

Fig. 1. The architecture for an agent in the convention-based cooperative multiagent system. Note that the "NLP" block works offline, meaning the processing will be done only once before the agent decision-making process occurs.

In the "NLP" block, we find the modules constituting the pipeline for processing natural language conventions and generating their formal representations. In the "Agent Decision-making" block, there are modules concerned with the agent's knowledge of the world (World Model) and the module building explanations for the actions of others (Theory of Mind — ToM). These two modules contain the agent's beliefs and methods to update them.

4.1 NLP

The "NLP" block's goal is to translate natural language conventions into a machine-readable formalism so that the agent can adapt its strategies. As stated in Section 3.1, adapting an off-the-shelf translation system is not feasible as all previous systems we are aware of are not maintained anymore [11, 16]. Therefore, we must adapt some of the methods and ideas of these systems to develop our processing pipeline. This pipeline includes a preprocessing step, a parser (with a tagging/annotating step), a database for ontologies, and a novel algorithm that generates the formal representation of conventions.

The "Preprocessing" module, as discussed in 3.1, will contain the techniques that must be applied for a general norm extraction task. For instance, H-Group

Conventions are published in HTML format, so the text needs to be extracted from the HTML code for those conventions as input. We can either manually extract them or apply existing web content extraction tools such as GOOSE[10], which was reported with the best performance for English newspaper text [3][11].

In the "Parsing/Tagging" module, existing parsers like Standford Parser can be used to retrieve the semantic and syntactic features from the conventions. We can assume that the parsing output will neither contain a vast vocabulary nor a complex sentence structure. The terms in the conventions will thus refer to the limited ontology of a particular domain, e.g. cards, colours, or numbers in Hanabi, so the vocabulary is naturally tiny. Similarly, conventions are relatively simple rules to be understandable by the public, e.g. conventions in Table 1. Differently from legal norms, the conventions we aim at are thus straightforward. Given this simplicity, we consider first-order logic expressive enough to formalise conventions. More concretely, [34] proposed a representation language called "Agent Situation Language" (ASL) used to represent the rules of games. We will explore using this language, or an extension of it, to represent conventions, as conventions have a similar expressive power to game rules. We might explore using Jason as the interpreter of ASL since ASL is similar to Agentspeak [38], a programming language used to represent some Hanabi conventions in [33].

"Ontology" is a ontology database, which works closely with the "Search-and-Match" module for formal conventions' generation. A simpler mechanism of "Search-and-Match" would be to analyse the frequent words from the input and their semantic and syntactic features, which need no ontology. For example, when considering the tokens extracted from a convention, the verb phrases (`VP`) are naturally mapped into predicates, and noun phrases (`NP`) are naturally mapped into predicate arguments. Therefore, a simple rule can be: from a sentence `S = NP V ADJ` generate `V(NP, ADJ)`. For instance, from "The card colour is red", we obtain the predicate instance `is(card_colour, red)`. A more complex rule can combine a constituency tree with a dependency tree. This combination will help determine which words should be placed together as predicates and arguments. A similar method is proposed in [28]: they wrote the predicate in a slightly different way to our proposal, as `dependency(governor, dependent)`, and created categories for different semantic rules. Nevertheless, ontologies are much richer structures that allow us to represent complex knowledge within a particular domain [5, 8, 31]. Thus, if there is an existing ontology in the particular domain of the conventions or a reference ontology, we can use it to determine the meaning of the words[12]. Applying an existing ontology can also reduce the work of creating formal representations for conventions from scratch. Back to the sentence "The card colour is red", instead of creating a rule to capture the

[10] https://github.com/goose3/goose3

[11] Another tool with the best performance in [3] was Newspaper. Unlike GOOSE, it was primarily designed for newspaper texts and cannot extract structured text.

[12] If no ontology exists, then we can generate one from the text using parsing and concepts/relation extraction rules, which consider semantic and syntactic features of the words [26].

features, we can search the ontology for concepts like "card" and "colour" and use the known relation between them, that is, cards have colours, and red is a common colour for cards. With these concepts and relations, we can formalise this sentence more efficiently. The intuitive meaning behind the name "Search-and-Match" is that the algorithm will use the input (words and tags in parsed sentences) as keywords to search the same (or similar) concepts in the "Ontology" database, match the keywords with the concepts, decide which relations can be found between the keywords, and generate the formal representation of the convention based on these relations.

However, some utterances, words or phrases represent features requiring formalising domain-dependent solid knowledge. This knowledge can be expressed as logical formulae patterns that include ontological elements from "Ontology", and some of the features extracted from the text of the convention. The "Search-and-Match" module will first consult the "Ontology" with those utterances, words or phrases from its input, and try to match them with ontology concepts based on their semantic similarity (e.g. the semantic features they share or the similarity between their semantic features). If the existing ontology in the database does not contain these (domain-specific) concepts, this module should be able to update or enrich the ontology. For instance, consider the expression "right-most card" in Table 1. It is a relative concept: its meaning may refer to either a physical position (e.g. fifth slot in Hanabi) or a "chop" (see the example in Section 2). The ontology should be able to help select the second (logical) meaning as the actual meaning of the expression "right-most card". Here is the example for *discard chop (right-most unclued) card* code that the ontology should provide:

```
if   convention_concept(right_most_card) and
     next_option([discard(_,_,X),discard(_,_,Y),discard(_,_,Z)]) and
     clued(_,_,Z) and ~clued(_,_,Y) and ~clued(_,_,X) and X < Y
then next_action(discard(_,_,Y))
```

That in English would read like "if the convention text mentions the word right_most_card, the strategic component doubts about discarding one of three cards, and only one of them is clued, then the card to discard will be the one of the non-clued two that is in the right-most position."

Such schemes can be grouped into structures representing the semantic similarity of the concepts. For instance, the code patterns for "right-most" and "left-most" concepts can be placed close to each other, possibly hanging from "position", even if "position" does not appear in the conventions. For example, see the pattern associated with the concept of "position". It contains a variable `OP` as a placeholder for an operator that can be later instantiated once we univocally determine the concrete position expressed in the convention:

```
[position(OP) for convention([position, place, location])]:=
if   next_option([discard(_,_,X),discard(_,_,Y),discard(_,_,Z)]) and
     clued(_,_,Z) and ~clued(_,_,Y) and ~clued(_,_,X) and X OP Y
then next_action(discard(_,_,Y))
```

If a word in the convention (e.g. "right_most" or "right_place") is found by the ontology as related with the concept "right_most_card", the ontology can then provide the instance **position(<)**, or **position(>)** instead if the concept "left_most_card" is found. Such structure can be expressed in the language for formal conventions as:

```
[right_most_card(<) is_a position(OP)
          for convention([right_most, on_the_right, right_place])]
```

A complex ontology structure might have programming patterns for some of its concepts but not necessarily for all. If no programming patterns can be obtained from the ontology, a request for an update may be issued.

When importing an existing domain-independent ontology, we must particularise it to the context of an application, e.g. Hanabi, adding programming patterns. For instance, from a node in an ontology like "position," we can add as leaves more domain-specific concepts ("left_most_card" and "right_most_card") and associate them with the rules shown above.

In short, the "Search-and-Match" module performs two operations. On the one hand, it updates ontologies by adding concepts appearing in the conventions under analysis and associating formal convention programming patterns with them. This ontology can be enriched (e.g. by training a long short-term memory neural network for updating ontologies when new concepts are introduced [39]). On the other hand, it instantiates the parameters of existing formal convention programming schemes with particular values coming from the annotated words from the NLP conventions analysis.

4.2 World Model

The "World Model" is the module that represents knowledge about the environment. It contains Precepts, Domain-related Clauses, and Impossibility Clauses, all representing different kinds of information.

Factual information (facts) about the environment is represented as literals. The initialisation of an agent's set of beliefs can thus be obtained from observing the environment. In our running example, these facts include the colour and number of cards other players hold. Though each agent does not have explicit knowledge of its cards, they can make inferences about the cards' colour or number. The system's possible states and actions are limited since there are just two types of clauses (about colour and number). "Domain-related" clauses indicate relationships among literals and specify more sophisticated characteristics; for example, a "playable" card must have a number which is one unit above the number on the card of the same colour that is on the stack on the table (e.g. red 3 is a "playable" card when there is a red 2 on the top of the red stack). "Impossibility clauses" express the circumstances where two literals cannot both be true. For example, the impossibility clause for "two different cards cannot occupy the same slot" will have its condition become true when more than one card is assigned to the same slot.

4.3 ToM

The "ToM" module represents the beliefs that one agent has about the beliefs of others. It combines Theory of Mind clauses and Abducible Clauses. Theory of Mind clauses explicitly represent agents' beliefs about facts of the environment. In contrast, Abducible clauses represent the possible beliefs they might have about facts of the environment.

The Theory of Mind Clauses are based on belief chains (see Section 3.2). Beliefs are encoded as literals of the type **believes(Ag, F)** in [33], which are true when the observer believes that an observee **Ag** is aware of a fact **F**. In our context, the concept of *fact* is equivalent to *belief* in the general definition of ToM because the observation of actions and the state of the environment are the only things we plan to use. As the environment is not fully observable (e.g. player's cards are hidden from them), beliefs do not necessarily correspond to reality. This is so because agents will update beliefs via querying specific ToM clauses based on abduction, and abduction does not necessarily provide truthful consequences. A relevant source of belief updates is those abductive consequences that an agent i derives from the beliefs it holds about the beliefs of another agent j on i, that is, on itself. In that case, an agent becomes an observer of itself through the eyes of another agent, observer and observee simultaneously.

Example: Alice clues one of Bod's cards, telling him its colour is red. When doing so, Alice can infer how Bob might interpret that piece of knowledge in terms of Bob's beliefs about Alice's beliefs leading to her telling Bob the colour of the card. For instance, Bob may infer that Alice is giving him a save clue (see Table 1) not to discard the card since this card is his current chop card.

Similarly to [33], we will leave out of our architecture any mechanism to determine beliefs about actions to be taken by other agents. We do so because of the high complexity of this kind of reasoning. However, we think a ToM representation with such capability would improve our agent architecture. We will consider it as future work.

Abducible Clauses complement the Theory of Mind Clauses. They add *potential* beliefs to the knowledge base as long as they do not contradict any preexisting beliefs. Note that these clauses are domain-specific.

Example: Alice is currently holding the belief *I have a red or blue card in the third slot*. If Alice's abduction mechanism produces *I have a red card in the third slot*, which is not contradictory to the current belief, she may (defeasibly) infer that the card's colour is red and act accordingly.

4.4 Action Selection

The `SelectAction` function in [33] relies on *Action Selection Clauses* written in AgentSpeak. These clauses represent the actions the agents might take and the beliefs they need to hold to take these actions. All the clauses also contain priority information encoded as a natural number. In [33], the game rules were manually coded as an environment in Java, while some conventions were manually programmed as Action Selection clauses. In the same work, the `SelectAction` function implements hard-coded strategies and takes Action Selection Clauses as input. After ranking the Action Selection Clauses based on their priorities, the function checks the clauses in order starting from the highest priority ones. If the clause's body is true according to the beliefs the agent is holding, and the potentially abducible beliefs, then the action suggested by the clause is selected; the remaining clauses are not considered. The game rules further verify the feasibility of this selected action. If the verification fails, the agent will take a default action (which can be defined by the developer).

Our architecture will modify the hard-coded strategies described and implemented in the `SelectAction` function by a customisable component. Our "Action Selection" module will receive three kinds of input: the rules of the game from the "Strategies" database written in ASL, the Action Selection Clauses in [33] rewritten in ASL, and the other formal conventions, also written in ASL, from the "Search-and-Match" module. Note that the SelectAction function written in ASL might have a different structure than the one written in AgentSpeak.

5 Discussion and Future Work

This short paper presents the initial ideas for an NLP-based agent architecture capable of processing conventions expressed in natural language. We have illustrated the architecture using examples from the card game Hanabi. Our next objective is to implement an agent following this architecture and putting it to work playing with other agents. We will check if our NLP correctly interprets the conventions when our agent plays with other agents that have the conventions hardwired in their strategy.

First, We will adapt and extend the model in [33] with additional modules for NLP. There will likely be some modifications to the original model, such as for the `SelectAction` function. The game rules and some conventions were manually coded in the original architecture. Thus, that implementation will be our testbed against which we will test the correct workings of our NLP component. We will also consider modifying the language ASL proposed in [34]. For instance, as discussed in Section 2, manual annotation of rule priorities may not be the best solution. We will work with real-world conventions to determine whether an alternative approach for ranking is required. In addition, apart from the already mentioned H-Group Conventions, other sources of conventions might be used. An example can be conventions generated from non-natural language data (e.g. records of game playing [18, 23]). We are planning to implement an existing dataset pairing natural language sentences with their first-order logic representation [27, 29]. Our objective is to have a pairing between conventions and their formal representation. However, as it is unlikely that we can have a large set of such pairings, because they require a lot of manual work and the number of conventions is not very large, we would like to explore transfer learning techniques over the pairings in [27, 29]. We also plan to create a more general version of our architecture, adaptable to any convention-based cooperative scenario. Also, we plan to extend the architecture to formalise norms that restrict agents' behaviour, for instance, by limiting the set of available actions.

One of the challenges we are meeting is the preprocessing of the conventions. Even though an engineer can create specific patterns to rewrite sentences into simplified English, we want to automate the procedure using existing tools. Submitting prompts to large-language models for generating simplified sentences or a rephrasing of the sentence can be one of the possible solutions. Another challenge is the reliability of the module's output, whether it represents the convention precisely enough for the system to process and for the agents to follow. In this case, we need a mechanism to generate several formal representations as candidates and select them based on their performance in the system.

Although there are many appealing research topics in the study of conventions. For instance, how to model reaction mechanisms to deal with agents that break conventions, norm emergence [35], or the dynamics of conventions during gameplaying we will limit the scope of our research to the topics mentioned in the previous paragraphs.

Acknowledgements This research is conducted under the REDI Program, a project that has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no. 101034328. This paper reflects only the author's view and the Research Executive Agency is not responsible for any use that may be made of the information it contains. This research also receives support from the VALAWAI project (Horizon #101070930) and the VAE project (Grant no. TED2021-131295 B-C31) funded by MCIN/AEI/10.13039/501100011033.

References

1. Ågotnes, T., Van Der Hoek, W., Rodríguez-Aguilar, J.A., Sierra, C., Wooldridge, M.J.: On the logic of normative systems. In: IJCAI. vol. 7, pp. 1175–1180 (2007)
2. Balke, T., da Costa Pereira, C., Dignum, F., Lorini, E., Rotolo, A., Vasconcelos, W., Villata, S.: Norms in MAS: Definitions and Related Concepts. In: Andrighetto, G., Governatori, G., Noriega, P., van der Torre, L.W.N. (eds.) Normative Multi-Agent Systems, Dagstuhl Follow-Ups, vol. 4, pp. 1–31. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2013). https://doi.org/10.4230/DFU.Vol4.12111.1, http://drops.dagstuhl.de/opus/volltexte/2013/3998
3. Barbaresi, A., Lejeune, G.: Out-of-the-box and into the ditch? multilingual evaluation of generic text extraction tools. In: Language Resources and Evaluation Conference (LREC 2020). pp. 5–13 (2020)
4. Bard, N., Foerster, J.N., Chandar, S., Burch, N., Lanctot, M., Song, H.F., Parisotto, E., Dumoulin, V., Moitra, S., Hughes, E., Dunning, I., Mourad, S., Larochelle, H., Bellemare, M.G., Bowling, M.: The hanabi challenge: A new frontier for ai research. Artificial Intelligence **280** (3 2020). https://doi.org/10.1016/j.artint.2019.103216

5. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. ScientificAmerican.com (05 2001)
6. Boella, G., Di Caro, L., Robaldo, L.: Semantic relation extraction from legislative text using generalized syntactic dependencies and support vector machines. In: Theory, Practice, and Applications of Rules on the Web: 7th International Symposium, RuleML 2013. pp. 218–225. Springer (2013)
7. Canaan, R., Gao, X., Togelius, J., Nealen, A., Menzel, S.: Generating and adapting to diverse ad-hoc partners in hanabi. IEEE Transactions on Games (2022)
8. Consortium, W.W.W.: Owl 2 web ontology language document overview (second edition), https://www.w3.org/TR/owl2-overview/
9. da Costa Pereira, C., Tettamanzi, A.G., Villata, S., Liao, B., Malerba, A., Rotolo, A., van Der Torre, L.: Handling norms in multi-agent system by means of formal argumentation. Journal of Applied Logics-IfCoLoG Journal of Logics and their Applications **4**(9), 1–35 (2017)
10. Cuzzolin, F., Morelli, A., Cirstea, B., Sahakian, B.J.: Knowing me, knowing you: theory of mind in ai. Psychological medicine **50**(7), 1057–1061 (2020)
11. Dragoni, M., Villata, S., Rizzi, W., Governatori, G.: Combining nlp approaches for rule extraction from legal documents. In: 1st Workshop on MIning and REasoning with Legal texts (MIREL 2016) (2016)
12. Ferraro, G., Lam, H.P.: Nlp techniques for normative mining. FLAP **8**(4), 941–974 (2021)
13. Ferraro, G., Lam, H.P., Tosatto, S.C., Olivieri, F., Islam, M.B., van Beest, N., Governatori, G.: Automatic extraction of legal norms: Evaluation of natural language processing tools. In: Sakamoto, M., Okazaki, N., Mineshima, K., Satoh, K. (eds.) New Frontiers in Artificial Intelligence. pp. 64–81. Springer International Publishing (2020)
14. Foerster, J., Song, F., Hughes, E., Burch, N., Dunning, I., Whiteson, S., Botvinick, M., Bowling, M.: Bayesian action decoder for deep multi-agent reinforcement learning. In: Proceedings of the 36th International Conference on Machine Learning. vol. 97, pp. 1942–1951. PMLR (2019)
15. Gao, X., Singh, M.P.: Extracting normative relationships from business contracts. In: AAMAS. pp. 101–108. Citeseer (2014)
16. Gaur, S., Vo, N.H., Kashihara, K., Baral, C.: Translating simple legal text to formal representations. In: New Frontiers in Artificial Intelligence: JSAI-isAI 2014 Workshops. pp. 259–273. Springer (2015)
17. Governatori, G., Rotolo, A.: A conceptually rich model of business process compliance. In: Proceedings of the Seventh Asia-Pacific Conference on Conceptual Modelling-Volume 110. pp. 3–12. Citeseer (2010)
18. Gray, J., Lerer, A., Bakhtin, A., Brown, N.: Human-level performance in no-press diplomacy via equilibrium search (2020), arXiv preprint arXiv:2010.02923
19. Harsanyi, J.C.: Games with randomly disturbed payoffs: A new rationale for mixed-strategy equilibrium points. International journal of game theory **2**(1), 1–23 (1973)
20. Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., Silver, D.: Rainbow: Combining improvements in deep reinforcement learning. Proceedings of the AAAI Conference on Artificial Intelligence **32**(1) (2018). https://doi.org/10.1609/aaai.v32i1.11796, https://ojs.aaai.org/index.php/AAAI/article/view/11796
21. Hu, H., Lerer, A., Cui, B., Pineda, L., Brown, N., Foerster, J.: Off-belief learning. In: Proceedings of the 38th International Conference on Machine Learning. vol. 139, pp. 4369–4379 (2021)

22. Hu, H., Lerer, A., Peysakhovich, A., Foerster, J.: “other-play” for zero-shot coordination. In: Proceedings of the 37th International Conference on Machine Learning. vol. 119, pp. 4399–4410 (2020)
23. Hu, H., Wu, D.J., Lerer, A., Foerster, J., Brown, N.: Human-ai coordination via human-regularized search and learning (2022), arXiv preprint arXiv:2210.05125
24. Jacob, A.P., Wu, D.J., Farina, G., Lerer, A., Hu, H., Bakhtin, A., Andreas, J., Brown, N.: Modeling strong and human-like gameplay with kl-regularized search. In: Proceedings of the 39th International Conference on Machine Learning. vol. 162, pp. 9695–9728 (2022)
25. Kaptein, R., Serdyukov, P., De Vries, A., Kamps, J.: Entity ranking using wikipedia as a pivot. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management. p. 69–78. CIKM ’10, Association for Computing Machinery, New York, NY, USA (2010). https://doi.org/10.1145/1871437.1871451
26. Kumar, N., Kumar, M., Singh, M.: Automated ontology generation from a plain text using statistical and nlp techniques. International Journal of System Assurance Engineering and Management **7**, 282–293 (2016)
27. Levkovskyi, O., Li, W.: Generating predicate logic expressions from natural language. In: SoutheastCon 2021. pp. 1–8. IEEE (2021)
28. Longo, C.F., Longo, F., Santoro, C.: Caspar: Towards decision making helpers agents for iot, based on natural language and first order logic reasoning. Engineering Applications of Artificial Intelligence **104**, 104269 (2021)
29. Lu, X., Liu, J., Gu, Z., Tong, H., Xie, C., Huang, J., Xiao, Y., Wang, W.: Parsing natural language into propositional and first-order logic with dual reinforcement learning. In: Proceedings of the 29th International Conference on Computational Linguistics. pp. 5419–5431 (2022)
30. Lupu, A., Cui, B., Hu, H., Foerster, J.: Trajectory diversity for zero-shot coordination. In: Proceedings of the 38th International Conference on Machine Learning. vol. 139, pp. 7204–7213 (2021)
31. McGuinness, D.L., Van Harmelen, F., et al.: Owl web ontology language overview, https://www.w3.org/TR/2004/REC-owl-features-20040210/
32. Michel, M., Djurica, D., Mendling, J.: Identification of decision rules from legislative documents using machine learning and natural language processing. In: Proceedings of the 55th Hawaii International Conference on System Sciences. pp. 6247–6256 (2022)
33. Montes, N., Osman, N., Sierra, C.: Combining theory of mind and abduction for cooperation under imperfect information. In: Baumeister, D., Rothe, J. (eds.) Multi-Agent Systems. pp. 294–311. Springer International Publishing, Cham (2022)
34. Montes, N., Osman, N., Sierra, C.: A computational model of ostrom’s institutional analysis and development framework. Artificial Intelligence **311**, 103756 (2022). https://doi.org/https://doi.org/10.1016/j.artint.2022.103756
35. Morales, J., Lopez-Sanchez, M., Rodriguez-Aguilar, J.A., Vasconcelos, W., Wooldridge, M.: Online automated synthesis of compact normative systems. ACM Transactions on Autonomous and Adaptive Systems (TAAS) **10**(1), 1–33 (2015)
36. Olson, T., Forbus, K.: Learning norms via natural language teachings. In: Proceedings of the Ninth Annual Conference on Advances in Cognitive Systems (2021)
37. Pehcevski, J., Vercoustre, A.M., Thom, J.A.: Exploiting locality of wikipedia links in entity ranking. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) Advances in Information Retrieval. pp. 258–269. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)

38. Rao, A.S.: Agentspeak(l): Bdi agents speak out in a logical computable language. In: Van de Velde, W., Perram, J.W. (eds.) Agents Breaking Away. pp. 42–55. Springer Berlin Heidelberg, Berlin, Heidelberg (1996)
39. Sanagavarapu, L.M., Iyer, V., Reddy, R.: A deep learning approach for ontology enrichment from unstructured text. arXiv preprint arXiv:2112.08554 (2021)
40. Shih, A., Sawhney, A., Kondic, J., Ermon, S., Sadigh, D.: On the critical role of conventions in adaptive human-ai collaboration. In: 9th International Conference on Learning Representations (2021). https://doi.org/10.48550/arXiv.2104.02871
41. Sleimi, A., Sannier, N., Sabetzadeh, M., Briand, L., Dann, J.: Automated extraction of semantic legal metadata using natural language processing. In: 2018 IEEE 26th International Requirements Engineering Conference (RE). pp. 124–135 (2018). https://doi.org/10.1109/RE.2018.00022
42. Treutlein, J., Dennis, M., Oesterheld, C., Foerster, J.: A new formalism, method and open issues for zero-shot coordination. In: Proceedings of the 38th International Conference on Machine Learning. vol. 139, pp. 10413–10423 (2021)
43. Tucker, M., Zhou, Y., Shah, J.: Adversarially guided self-play for adopting social conventions. CoRR **abs/2001.05994** (2020), https://arxiv.org/abs/2001.05994
44. de Weerd, H., Verbrugge, R., Verheij, B.: Higher-order social cognition in the game of rock-paper-scissors: A simulation study. In: Bonanno, G., van Ditmarsch, H., van der Hoek, W. (eds.) Proceedings of the 10th Conference on Logic and the Foundations of Game and Decision Theory. pp. 218–232 (2012)
45. de Weerd, H., Verheij, B.: The advantage of higher-order theory of mind in the game of limited bidding. In: van Eijck, J., Verbrugge, R. (eds.) CEUR Workshop Proceedings. vol. 751, pp. 149–164 (2011)
46. Wyner, A., Peters, W.: On rule extraction from regulations. Frontiers in Artificial Intelligence and Applications **235**, 113–122 (2011). https://doi.org/10.3233/978-1-60750-981-3-113