Abductive Reasoning for Continual Dialogue Understanding

Miroslav Janíček

German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany
mjanicek@dfki.de

Abstract. In this paper I present a continual context-sensitive abductive framework for understanding situated spoken natural dialogue. The framework builds up and refines a set of partial defeasible explanations of the spoken input, trying to infer the speaker's intention. These partial explanations are conditioned on the eventual verification of the knowledge gaps they contain. This verification is done by executing test actions, thereby going beyond the initial context. The approach is illustrated by an example set in the context of human-robot interaction.

1 Introduction

In task-oriented dialogues between two agents, such as between two humans or a human and a robot, there is more to dialogue than just understanding words. The agents need to understand what is being talked about, but it also needs to understand why it was told something. In other words, what the speaker *intends* the hearer to do with the information in the larger context of their joint activity.

Therefore, understanding language can be phrased as an *intention recognition* problem: given an utterance from the human, how do we find the intention behind it?

In this paper, I explore an idea inspired by the field of continual planning [4], by explicitly capturing the possible knowledge gaps in such an interpretation. The idea is based on the notion of *assertion*, an explicit test for the validity of a certain fact, going beyong the current context.

The structure of the paper is as follows. After briefly introducing the notion of intention recognition and abduction in the next section, I introduce the continual abductive reasoning mechanism in §3, and discuss it on an example in §4, before concluding with a short summary.

2 Background

The idea of expressing *understanding* in terms of intention recognition has been introduced by H. P. Grice [7,14]. In this paper, I build on Stone and Thomason's approach to the problem [17] who in turn extend the work done by Hobbs and others [8], and base their approach to intention recognition on *abductive reasoning.*

Abduction. Abduction is a method of explanatory logical reasoning introduced into modern logic by Charles Sanders Peirce [6]. Given a theory T, a rule $T \vdash A \rightarrow B$ and a fact B, abduction allows inferring A as an explanation of B. B can be deductively inferred from $A \cup T$. If $T \not\vdash A$, then we say that A is an *assumption*.

There may be many possible causes of B besides A. Abduction amounts to *guessing*; assuming that the premise is true, the conclusion holds too. To give a well-known example:

> Suppose we are given two rules saying "if the sprinkler is on, then the lawn is wet" and "if it rained, then the lawn is wet". Abductively inferring the causes for the fact that the lawn is wet then yields two possible explanations: the sprinkler is on, or it rained.

Obviously, as there may be many possible explanations for a fact, in practical applications there needs to be a mechanism for selecting the best one. This may be done by purely syntactic means (e.g. lengths of proofs), or semantically by assigning *weights* to abductive proofs and selecting either the least or most costly proof [16], or by assigning probabilities to proofs [12]. In that case, the most probable proof is also assumed to be the best explanation. Our approach combines both aspects.

Intention recognition. Abduction is a suitable mechanism to perform inferences on the pragmatic (discourse) level. For understanding, abduction can be used to infer the explanation *why* an agent said something, in other words the *intention* behind the utterance. Reversing the task, given an intention, we may infer the way *how* to express it [18]. Intentions can therefore serve as a middle representational layer and abduction as the inference mechanism by using which we either turn a realisation into an intention, or the other way around.

3 Approach

This paper extends the work of Stone and Thomason on context-sensitive language understanding by explicitly modelling the knowledge gaps that inevitably arise in such an effort due to uncertainty and partial observability. The approach is based on generating partial hypotheses for the explanation of the observed behaviour of other agents, under the assumption that the observed behaviour is intentional. These partial hypotheses are defeasible and conditioned on the validity (and eventual verification) of their assumptions.

In this section, I examine the an abductive reasoning system capable of representing knowledge gaps in the form of partial proofs, how such partial proofs can be generated and verified or falsified, and the semantic framework used in our system to capture linguistic meaning that the system then grounds in reality.

3.1 Partial Abductive Proofs

Our abductive inference mechanism is essentially Hobbs and Stickel's logic programming approach to weighted abduction [8,16] enhanced by a contextual aspect [1] with the weights in the system being assigned a probabilistic interpretation following Charniak and Shimony [5].

Proof procedure. Formally, inference in our system makes use of four ingredients: *facts*, *rules*, *disjoint declarations* and *assumability functions*, collectively called the *abduction context*, and using these iteratively in order to derive proofs of an initial *goal*.

- Facts are modalised formulas of the form

  $$\mu : A$$

  where μ is a (possibly empty) sequence of contexts, and A is an atomic formula, possibly containing variables.
- Rules are modalised Horn clauses, i.e. formulas of the form

  $$(\mu_1 : A_1/t_1) \wedge ... \wedge (\mu_n : A_n/t_n) \rightarrow (\mu_H : H)$$

  where each of the $\mu_i : A_i$ and $\mu_H : H$ are modalised formulas. Each antecedent is annotated by t_i, which determines the way the antecedent is manipulated and is one of the following:
  - *true* – the antecedent has to be proven, i.e. either it is a fact, or a head of some rule;
  - *assumable(f)* – the antecedent is assumable under function f;
  - *assertion* – the antecedent is asserted, i.e. identifies a knowledge gap, conditioning the validity of the proof on it being proved in a subsequent reinterpretation (see below).
- Assumability functions are partial functions f, $f : \mathcal{F} \rightarrow \mathbb{R}_0^+$, where $\mathcal{F}$ is the set of modalised formulas, with the additional monotonicity property that if $F \in \text{dom}(f)$, then for all more specific (in terms of variable substitution) facts F', $F' \in \text{dom}(f)$ and $f(F) \leq f(F')$.
- A disjoint declaration is a statement of the form

  $$\text{disjoint}([\mu : A_1, ..., \mu : A_n])$$

  which specifies that at most one of the modalised formulas $\mu : A_i$ may be used in the proof. A_i and A_j cannot be unified for all $i \neq j$.

A *proof state* is a sequence of marked modalised formulas (called *queries* in this context)

$$Q_1[n_1], ...Q_m[n_m]$$

The markings n_i are one of the following:

Algorithm 1 (Nondeterministic) weighted abduction

```
$c$ = the abduction context
$L$ = the initial proof state

while $L$ contains a query marked as unsolved:
    $Q \leftarrow$ leftmost query in $L$ marked as unsolved
    choose a transformation rule $t$ so that APPLY-RULE$(t, Q, L, c)$ succeeds
    $L \leftarrow$ APPLY-RULE$(t, Q, L, c)$
```

- *unsolved(f)* – the query is yet to be proved, assumable under assumability
- *proved* – the query is proved or in the process of being proved;
- *assumed(f)* – the query is assumed under f;
- *asserted* – the query is asserted

The proof procedure starts from a single query marked as unsolved (called the *goal*), iteratively rewriting the proof state by manipulating the leftmost unsolved query Q_l. First, the query has to pass constraints imposed by disjoint declarations. If it does, it is either proved (using facts or rules), assumed under an assumability function, or eliminated if any of the queries to the right is unifiable with Q_l. In other words, each query is proved or assumed at most once.

The initial query Q is proved when there is no unsolved query in the proof state. The final proof state Π_Q is then the proof of Q. The proof procedure is schematised by Algorithm 1. Note that the proof procedure assures that the cost of the proofs are monotonic with respect to unification and application of rules, allowing for the use of efficient search strategies.

Knowledge gaps and assertions. Our extension of the "classical" logic-programming-based weighted abduction as proposed by Stickel and Hobbs lies in the extension of the proof procedure with the notion of *assertion* based on the work in continual automated planning [4], allowing the system to reason about information not present in the knowledge base, thereby addressing the need for reasoning under the open-world assumption.

In continual automated planning, assertions allow a planner to reason about information that is not known at the time of planning (for instance, planning for information gathering), an assertion is a construct specifiying a "promise" that the information in question will be resolved eventually.

By using a logic programming approach, we can use unbound variables in the asserted facts in order to reason not only about the fact that the given assertion will be a fact, but also under-specify its eventual arguments.

The proposed notion of an *assertion* for our abductive system is based on *test actions* $\langle F \rangle$ [2]. Baldoni et al. specify a test as a proof rule. In this rule, a goal F follows from a state $a_1, ..., a_n$ after steps $\langle F \rangle, p_1, ..., p_m$ if we can establish F on $a_1, ..., a_n$ with answer σ and this (also) holds in the final state resulting from executing $p_1, ..., p_m$.

An assertion is the transformation of a test into a partial proof which assumes the verification of the test, while at the same time conditioning the obtainability of the proof goal on the tested statements. $\mu : \langle D \rangle$ within a proof $\mathit{\Pi}[\langle D \rangle]$ to a goal C turns into $\mathit{\Pi}[D] \rightarrow C \wedge \mu : D$. Should $\mu : D$ not be verifiable, $\mathit{\Pi}$ is invalidated.

Probabilistic interpretation. In weighted abduction, weights assigned to assumed queries are used to calculate the overall proof cost. The proof with the lowest cost is the best explanation. However, weights are usually not assigned any semantics, and often a significant effort by the writer of the rule set is required to achieve expected results [8].

However, Charniak and Shimony [5] showed that by setting weights to $-\log$ of the prior probability of the query, the resulting proofs can be given probabilistic semantics.

Suppose that query Q_k can be assumed true with some probability $P(Q_k \text{ is true})$. Then if Q_k is assumable under assumability function f such that $f(Q_k) = -\log(P(Q_k \text{ is true}))$, and under the independence assumption, we can represent the overall probability of the proof $\mathit{\Pi} = Q_1[t_1], ..., Q_n[t_n]$ as

$$P(\mathit{\Pi}) = e^{\sum_{k=1}^{n} cost(Q_k)}$$

where

$$cost(Q_k) = \begin{cases} f(Q_k) & \text{if } m_i = assumed(f) \\ 0 & \text{otherwise} \end{cases}$$

The best explanation $\mathit{\Pi}_{best}$ of a query Q is then

$$\mathit{\Pi}_{best} = \underset{\mathit{\Pi} \text{ proof of } Q}{\arg\min} \; P(\mathit{\Pi})$$

Exact inference in such a system is NP-complete, and so is approximate inference given a threshold [5]. However, it is straightforward to give an anytime version of the algorithm – simply by performing iterative deepening depth-first search [13] and memorizing a list of most probable proofs found so far.

3.2 Generating Partial Hypotheses

For each goal G, a determinisation of Algorithm 1 returns a set of proofs H, with a total ordering on this set. Due to the use of assertions, some of these proofs may be partial, and their validity has to be verified. The presence of assertions in the proofs means that there is a knowledge gap, namely the truth value of the assertion. Each assertion thus specifies the need for performing a (test) action. This action might require the access to other knowledge bases than the abductive context, as in the case of resolving referring expressions, or an execution of a physical action.

Formally, given an initial goal G and context c, the abduction procedure produces a set H of hypotheses $c : \mathit{\Pi} \rightarrow C \wedge c_i : A_i$, where c_i is a sub-context in

Algorithm 2 (Nondeterministic) continual abduction

```
CONTINUAL-ABDUCTION($c, \Pi$):
    $c$ = context
    $\Pi$ = proof

    while $\Pi$ contains assertion $A$:
        $c' \leftarrow$ TEST-ACTION($c, A$)
        $H \leftarrow$ ABDUCE($c', A$)
        for all $\Pi' \in H$:
            CONTINUAL-ABDUCTION($c', \Pi'$)
    return $\Pi$
```

which where an assertion $A_i \in \Pi$ may be evaluated. Such proofs are thus both *partial* and *defeasible* — they may be both extended and discarded, depending on the evaluation of the assertions.

The set of possible hypotheses is continuously expanded until the best full proof is found. This process is defined in Algorithm 2.

The algorithm defines the search space in which it is possible to find the most probable proof of the initial goal G. The important point is, however, that it is just that — a definition. The actual implementation may keep track of the partial hypotheses it defines, and take the appropriate test actions when necessary, or postpone them indefinitely.

3.3 Representing Linguistic Meaning

For representing linguistic meaning in our system, we use the *Hybrid Logic Dependency Semantics* (HLDS), a hybrid logic [3] framework that provides the means for encoding a wide range of semantic information, including dependency relations between heads and dependents [15], tense and aspect [11], spatio-temporal structure, contextual reference, and information structure [10].

HLDS uses hybrid logic to capture dependency complexity in a model-theoretic relational structure, using ontological sorting to capture categorial aspects of linguistic meaning, and naturally capture (co-)reference by explicitly using *nominals* in the representation.

Generally speaking, HLDS represents an expression's linguistic meaning as a conjunction of modalised terms, anchored by the nominal that identifies the head's proposition:

$$@_{h:\mathsf{sort}_h} \; (\mathbf{prop}_h \wedge \langle \mathsf{R}_i \rangle \; (d_i : \mathsf{sort}_{d_i} \wedge \mathbf{dep}_i))$$

Here, the head proposition nominal is h. $\mathbf{prop}_h$ represents the *elementary predication* of the nominal h. The dependency relations (such as Agent, Patient, Subject, etc.) are modelled as modal relations $\langle R_i \rangle$, with the dependent being identified by a nominal d_i. Features attached to a nominal (e.g. $\langle \mathsf{Num} \rangle$ $\langle \mathsf{Quantification} \rangle$, etc.) are specified in the same way.

Figure 1 gives an example of HLDS representation (logical form) of the sentence "Take the mug". The logical form has three nominals, $event_1$, $agent_1$ and $thing_1$ that form a dependency structure: $event_1$ is the the head of dependency relations Actor (the dependent being $agent_1$), $\mathsf{Patient}$ ($thing_1$), and $\mathsf{Subject}$ ($agent_1$). Each nominal has an ontological sort (illustrated on $event_1$, the sort is $\mathsf{action\text{-}non\text{-}motion}$) a proposition (**take**), and features (Mood).

$$\begin{array}{l}
@_{event_1:\mathsf{action\text{-}non\text{-}motion}}(\mathbf{take} \wedge \\
\qquad\qquad \langle \mathsf{Mood} \rangle\ \mathbf{imp} \wedge \\
\qquad\qquad \langle \mathsf{Actor} \rangle\ agent_1 : \mathsf{entity} \wedge \\
\qquad\qquad \langle \mathsf{Patient} \rangle\ thing_1 : \mathsf{thing} \wedge \mathbf{mug} \wedge \\
\qquad\qquad\qquad \langle \mathsf{Delimitation} \rangle\ \mathbf{unique} \wedge \\
\qquad\qquad\qquad \langle \mathsf{Num} \rangle\ \mathbf{sg} \wedge \\
\qquad\qquad\qquad \langle \mathsf{Quantification} \rangle\ \mathbf{specific})) \wedge \\
\qquad\qquad \langle \mathsf{Subject} \rangle\ (agent_1 : \mathsf{entity} \wedge \mathbf{addressee}))
\end{array}$$

Fig. 1. HLDS semantics for the utterance "Take the mug"

$$\begin{array}{l}
\mathrm{sort}(\mathrm{event}_1, \mathrm{action\text{-}non\text{-}motion}), \\
\mathrm{prop}(\mathrm{event}_1, \mathrm{take}), \\
\mathrm{feat}(\mathrm{event}_1, \mathrm{mood}, \mathrm{imp}), \\
\mathrm{rel}(\mathrm{event}_1, \mathrm{actor}, \mathrm{agent}_1), \\
\mathrm{sort}(\mathrm{agent}_1, \mathrm{entity}), \\
\mathrm{prop}(\mathrm{agent}_1, \mathrm{addressee}), \\
\mathrm{rel}(\mathrm{event}_1, \mathrm{patient}, \mathrm{thing}_1), \\
\mathrm{feat}(\mathrm{thing}_1, \mathrm{delimitation}, \mathrm{unique}), \\
\mathrm{feat}(\mathrm{thing}_1, \mathrm{num}, \mathrm{sg}), \\
\mathrm{feat}(\mathrm{thing}_1, \mathrm{quantification}, \mathrm{specific})
\end{array}$$

Fig. 2. The translation of the hybrid logic formula in Figure 1 into abduction facts

Every logical form in HLDS, being a formula in hybrid logic, can be decomposed into a set of facts in the abductive context corresponding to its minimal Kripke model. The resulting set of abduction facts obtained by decomposing the logical form in Figure 1 is shown by Figure 2.

HLDS only represents the meaning as derived from the linguistic realisation of the utterance and does not evaluate the state of affairs denoted by it. This sets the framework apart from semantic formalisms such as DRT [9]. The grounding in reality is partly provided by the continual abductive framework by generating and validating (or ruling out) partial abductive hypotheses as more information is added to the system.

4 Example

Let us examine the mechanism in an example. Suppose that a human user is dealing with a household robot capable of manipulating objects (picking them up, putting them down). The robot and the human are both looking at a table with a represented by the term “mug_1”, and the human wants the robot to pick up the mug.

The human’s utterance,

“Take the mug.”

is analysed in terms of HLDS (see Figure 1), and its translation is made part of the abductive context c.

Suppose that proving the following goal in the context c

$$\text{uttered}(\text{human}, \text{robot}, \text{event}_1)$$

yields the following (best) proof, displayed with markings following §3.1:

$$\begin{array}{rlr}
\text{uttered}(\text{human}, \text{robot}, \text{event}_1) & [\textit{proved}] & (1) \\
\hline
\text{prop}(\text{event}_1, \text{take}) & [\textit{proved}] & (2) \\
\text{intends}(\text{event}_1, \text{human}, I) & [\textit{assumed}(p = 0.9)] & (3) \\
\text{rel}(\text{event}_1, \text{patient}, \text{thing}_1) & [\textit{proved}] & (4) \\
\text{refers\_to}(\text{thing}_1, X) & [\textit{asserted}] & (5) \\
\text{pre}(I, \text{object}(X)) & [\textit{asserted}] & (6) \\
\text{post}(I, \text{state}(\text{is-holding}(\text{robot}, X))) & [\textit{assumed}(p = 0.7)] & (7)
\end{array}$$

The proof is an explanation of the event (1) in terms of a partially specified intention I (3), defined by its pre- and post-condition. The precondition is the existence of an entity X (6), and the postcondition (7) is the state in which the robot is holding the entity X. The proof appeals to the logical form of the utterance (2, 4).

In the proof, atoms (3) and (7) are assumed under a constant assumability function that assigns them probability 0.9 and 0.7 respectively. This means that given our knowledge base, such a sentence expresses an intention of the human with prior probability 0.9, and that with prior probability 0.7, this intention has something to do with the robot physically taking holding some object (as opposed to uses such as “take a picture”).

The atoms (5) and (6) are marked as *asserted*. These assertions identify the knowledge gaps in the interpretation – the interpretation is only valid if they are verified using test actions.

Suppose that the assertion (5) is tested first. This amounts to resolving the referring expression, headed by the nominal $\textit{thing}_1$ in the logical form. The action is performed, giving rise to a new abduction context c', in which the abduction context is updated by specifying a reference resolution function r, yielding the following two hypotheses:

$$\begin{array}{rl}
\text{refers\_to}(\text{thing}_1, \text{mug}_1) & [\textit{proved}] \\
\hline
\text{ref}(\text{thing}_1, \text{mug}_1) & [\textit{assumed}(r)]
\end{array}$$

(i.e. p_1 refers to the mug under assumability function r),

$$\frac{\text{refers\_to}(\text{thing}_1, X)\ [\textit{proved}]}{\text{unknown-referent}(\text{thing}_1, X)\ [\textit{assumed}(p = \textit{0.4})]}$$

i.e. the reference was not resolved. This accounts for the possibility of misunderstanding, where the human might be referring to an object that is not part of common ground, and the reference thus cannot be resolved.

Now that the assertion (5) has been tested, the system can check the assertion (6). Depending on the value of the assumability function r above, it might first perform the test action for existence in the former (if $r > 0.4$), or the latter context (if $r < 0.4$), or in a random order (if $r = 0.4$).

In the former context, where the reference has been resolved to the mug, the robot might ask “Did you mean I should take *this* object?” (pointing at the mug, testing the hypothesis

$$\text{pre}(I, \text{object}(\text{mug}_1))$$

In the latter case, it might ask “Which object did you mean?”, prompting the human to give an answer that would ultimately become the proof of the test action for

$$\text{pre}(I, \text{object}(X))$$

in the proof above.

5 Conclusion and Future Work

This paper presents an abductive framework for natural language understanding that is based on abductive reasoning over partial hypotheses. The framework is set within the process of intention recognition.

The abductive framework is contextually-enhanced version of a logic programming approach to weighted abduction with a probabilistic semantics assigned to the weights. Our extension of this framework is in the introduction of the notion of *assertion*, which is essentially a requirement for a future test to verify or falsify the proposition, i.e. to fill a knowledge gap about the validity of the proposition. The hypotheses are therefore defeasible in the sense that the falsification of their assertions leads to a retraction and adoption of an initially less likely alternative.

By explicitly reasoning about these knowledge gaps, the system is able to go beyond the current context and knowledge base, addressing the need for reasoning under the open-world assumption.

Future research in this area will include a more informed interface to the decision-making processes involved in the selection of the hypotheses to test, and the stability of the partial hypotheses.

6 Acknowledgments

The work reported in this paper has been supported by the EU ICT FP7 projects CogX, "Cognitive Systems that Self-Understand and Self-Extend" (project no. 215181) and NIFTi, "Natural Human-Robot Collaboration in Dynamic Environments" (project no. 247870).

I would like to thank Geert-Jan Kruijff for insightful and inspiring discussions, and the reviewers for their comments.

References

1. Baldoni, M., Giordano, L., Martelli, A.: A modal extension of logic programming: Modularity, beliefs and hypothetical reasoning. J. Log. Comp. 8(5), 597–635 (1998)
2. Baldoni, M., Giordano, L., Martelli, A., Patti, V.: A modal programming language for representing complex actions. In: Proc. of DYNAMICS'98. pp. 1–15 (1998)
3. Blackburn, P.: Representation, reasoning, and relational structures: a hybrid logic manifesto. Logic Journal of the IGPL 8(3), 339–625 (2000)
4. Brenner, M., Nebel, B.: Continual planning and acting in dynamic multiagent environments. Journal of Autonomous Agents and Multiagent Systems (2008)
5. Charniak, E., Shimony, S.E.: Probabilistic semantics for cost based abduction. In: AAAI-90 proceedings (1990)
6. Fann, K.T.: Peirce's Theory of Abduction. Mouton, The Hague, The Netherlands (1970)
7. Grice, H.P.: Meaning. The Philosophical Review 66(3), 377–388 (1957)
8. Hobbs, J.R., Stickel, M.E., Appelt, D., Martin, P.: Interpretation as abduction. Tech. Rep. 499, AI Center, SRI International, Menlo Park, CA, USA (Dec 1990)
9. Kamp, H., Reyle, U.: From Discourse to Logic. Kluwer, Dordrecht, The Netherlands (1993)
10. Kruijff, G.J.M.: A Categorial-Modal Logical Architecture of Informativity: Dependency Grammar Logic & Information Structure. Ph.D. thesis, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic (2001)
11. Moens, M., Steedman, M.: Temporal ontology and temporal reference. Computational Linguistics 14(2), 15–28 (1988)
12. Poole, D.: Probabilistic Horn abduction and Bayesian networks. Artificial Intelligence 64(1), 81–129 (1993)
13. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall, Upper Saddle River, NJ, USA, second edn. (2003)
14. Schiffer, S.R.: Meaning. Oxford, Clarendon Press (1972)
15. Sgall, P., Hajičová, E., Panevová, J.: The Meaning of the Sentence in Its Semantic and Pragmatic Aspects. Reidel Publishing Company, Dordrecht, The Netherlands and Academia, Prague, Czechoslovakia (1986)
16. Stickel, M.E.: A Prolog-like inference system for computing minimum-cost abductive explanations in natural-language interpretation. Tech. Rep. 451, AI Center, SRI International, Menlo Park, CA, USA (Sep 1988)
17. Stone, M., Thomason, R.H.: Context in abductive interpretation. In: Proceedings of EDILOG 2002: 6th workshop on the semantics and pragmatics of dialogue (2002)
18. Stone, M., Thomason, R.H.: Coordinating understanding and generation in an abductive approach to interpretation. In: Proceedings of DIABRUCK 2003: 7th workshop on the semantics and pragmatics of dialogue (2003)