

Fast Forward through Opportunistic Incremental Meaning Representation Construction

Petr Babkin

Rensselaer Polytechnic Institute
Troy, New York
babkip@rpi.edu

Sergei Nirenburg

Rensselaer Polytechnic Institute
Troy, New York
nirens@rpi.edu

Abstract

One of the challenges semantic parsers face involves upstream errors originating from pre-processing modules such as ASR and syntactic parsers, which undermine the end result from the get go. We report the work in progress on a novel incremental semantic parsing algorithm that supports simultaneous application of independent heuristics and facilitates the construction of partial but potentially actionable meaning representations to overcome this problem. Our contribution to this point is mainly theoretical. In future work we intend to evaluate the algorithm as part of a dialogue understanding system on state of the art benchmarks.

1 Introduction

The versatility of human language comprehension overshadows countless transient failures happening under the hood. At various points during a conversation we are bound to tolerate error and uncertainty: coming either from the speech itself (e.g., disfluencies, omissions) or resulting from our own mistakes and deficiencies as a hearer (e.g., misinterpretation of an ambiguous utterance due to incomplete knowledge, etc.). The entire process is an ever recurring cycle of gap filling, error recovery and proactive re-evaluation. Better yet, some of the arising issues we choose not to resolve completely — leaving room for underspecification and residual ambiguity, or abandoning altogether. Similarly, we are highly selective with respect to material to attend to, glancing over the bits deemed of only minor relevance or redundant.

We believe incrementality and opportunism to be key to achieving this type of behavior in a computational NLU system. Incremental construction of meaning representations allows for their contin-

uous refinement and revision with gradually accumulating evidence. In addition, it gives the system an opportunity to make meta-level decisions of whether to pursue analysis of further material and/or to a greater depth; or to satisfice with a partial meaning representation built so far — thereby reducing the amount of work and avoiding potentially problematic material.

By opportunism we refer to the ability to simultaneously engage all available sources of decision making heuristics (morphological, syntactic, semantic, and pragmatic knowledge) as soon as their individual preconditions are met. On one hand, this provides a degree of independence among the knowledge sources such that a failure of any one of them does not necessarily undermine the success of the process as a whole. On the other hand, opportunistic application of heuristics facilitates the construction of partial or underspecified meaning representations when their complete versions are infeasible to obtain.

As a step towards incorporating these principles in an NLU system, we present a novel opportunistic incremental algorithm for semantic parsing. The rest of the paper is organized as follows. In the following section, we will introduce our approach while providing necessary background. Next, we will work through an example derivation, highlighting some of the features of the algorithm. Then, in the discussion section, we will position this work in the context of related work in the field, as well as within our own research agenda. Finally, we will conclude with the progress to date and plans for the evaluation.

2 Approach

2.1 Framework

Our description will be grounded within the theoretico-computational framework of Ontological Semantics (OntoSem) — a cognitively inspired

and linguistically motivated knowledge-rich account of NLU and related cognitive processes (McShane and Nirenburg, 2016).

The goal of the algorithm we are presenting in this paper is to build meaning representations of text (TMRs) in an incremental and opportunistic fashion. As a source of building blocks for the TMRs we will use the OntoSem lexicon and ontology. The lexicon provides a bridge between surface forms (e.g., word lemmas, multi-word expressions, or phrasals) and their corresponding semantic types (called concepts), while the ontology encodes the relationships among concepts, such as case roles defined on them. In addition, OntoSem provides accounts of modality, reference, temporal relations, and speech acts, which become part of the search space for our algorithm. A meaning representation in OntoSem is denoted as a collection of numbered frames with key-value entries called slots. An example of such a TMR corresponding to the natural language utterance “Apply pressure to the wound!” is shown below:

```
REQUEST-ACTION-1
  agent HUMAN-1
  theme APPLY-1
  beneficiary HUMAN-2
APPLY-1
  agent HUMAN-2
  instrument PRESSURE-1
  theme WOUND-INJURY-1
```

The root of this TMR is an instance of the REQUEST-ACTION speech act evoked by the imperative mood of the utterance. The agent (requester) of this speech act is the implied speaker of the utterance. The beneficiary (or patient, in other semantic formalisms) is the implied addressee of this message. Next, the theme of the speech act (the action being requested) is an instance of the APPLY event defined in the ontology. The agent of the action requested is specified to be the same as the addressee. The instrument is specified to be an instance of PRESSURE. Finally, the theme (or target) of the requested action references some known instance of a WOUND-INJURY, either introduced earlier in text, or contained in the common ground of the interlocutors.

2.2 Algorithm

A TMR can be represented as a (possibly rooted) graph $G = (V, E)$, where the set of vertices V represents semantic instances and the set of edges E

denotes case role linkings among them. We can thus formulate our problem generally as that of producing a pair (V, E) for an input utterance u as a sequence of tokens $\langle u_1, u_2, \dots, u_n \rangle$, so as to maximize some score: $\arg \max_{(V, E)} \text{score}(V, E)$.

The algorithm operates in two phases: the forward pass generates candidate solutions and the backward pass extracts the best scoring solution. The forward pass of the algorithm proceeds iteratively: by performing a series of incremental operations: *next* and *link*, which we define in order. The basic data structure used is an abstract set S of items p_0 through p_n .

next is defined as an incremental operation consuming one or more tokens from the input utterance $\langle u_1, u_2, \dots, u_n \rangle$ and returning zero or more items p to be added to the set S .

link operation is defined as accepting a set S_i and returning a new set S_{i+1} and a bipartite graph defined by $(S_i, S_{i+1}, L_{i+1,i})$, where $L_{i+1,i}$ is a set of edges between the two sets-partitions.

The following snippet provides a high level description of the execution of the forward pass of the algorithm.

```
function EXPAND( $\langle u_1, u_2, \dots, u_n \rangle$ )
   $S_i \leftarrow \emptyset$ 
  initialize lattice  $C$  with tuple  $\langle S_i, \emptyset, \emptyset \rangle$ 
  while NO-ACTIONABLE-TMR( $C$ ) do
    if MORE-INPUT-NEEDED( $C$ ) then
       $S_i \leftarrow S_i \cup \text{NEXT}(\langle u_1, u_2, \dots, u_n \rangle)$ 
    else
       $C \leftarrow C \cup \{\text{LINK}(S_i)\}$ 
       $S_i \leftarrow S_{i+1}$ 
      if FIXED-POINT( $C$ ) then
        return  $\emptyset$ 
      end if
    end if
  end while
  return  $C$ 
end function
```

Expand in the name of the function refers to the fact that the lattice data structure used to store the candidate solutions is repeatedly expanded by the two incremental operations we just introduced. The *next* operation expands the width of the lattice by adding new elements to the set S_i , which can be thought of as a “layer” of the lattice. The *link* operation, expands the depth of the lattice by creating a new “layer” and connecting it to the elements of the previous layer. Intuitively, the process can be visualized diagonally as alternating

horizontal and vertical expansions. The entire lattice can be thought of as a metaphorical “loom”, on which meaning representations are “woven”. The depth of the lattice is the depth of nesting in the corresponding meaning representation tree with a linear upper bound. In practice, however, we expect it to approach $O(\log|U|)$ as most of the nesting occurs in multiple-argument instances.

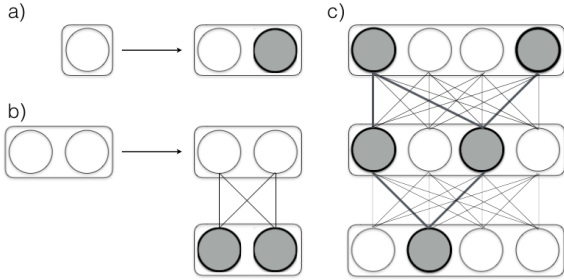


Figure 1: A schematic depiction of the incremental operations *next* (a), *link* (b), and the resulting lattice containing a solution (c).

The following conditionals are employed to control the execution flow of the main algorithm:

1. NO-ACTIONABLE-TMR captures high-level desiderata for execution continuation. In the current implementation the algorithm simply requires that the TMR is headed by an event and that its as well as its constituents’ core roles (e.g., agent, theme) are filled; and that it spans all components of the input. Work is ongoing to operationalize pragmatic considerations for actionability, including whether a partial TMR is coherent with preceding discourse or is consistent with a known script, that could warrant early halting.

2. MORE-INPUT-NEEDED is triggered when there are no components to link despite the TMR being non-actionable.

3. FIXED-POINT basically means that further iteration does not produce any novel interpretations.

The subsequent solution extraction phase amounts to simply extracting TMR candidates from the lattice via depth-first traversal and ranking them by the cumulative scores of constituent nodes and edges. It should be noted that this procedure operates on a considerably reduced subset of the original search space as it effectively chooses among a limited set of viable candidates produced during the forward pass. Going a level deeper in our description, we will now turn to the implementation details of the incremental operations *next* and *link*.

The *next* operation translates the next word token (or a group of tokens) from the input utterance into corresponding semantic type that are then combined together into a TMR. It is currently realized as a basic lexical look-up by lemma, with greedy matching for multi-word expressions and non-semantically loaded tokens are skipped. For polysemous words, semantic representations for all variants are considered. While the On-toSem lexicon specifies a battery of syntactic and morphological constraints imposed on each word sense (e.g., syntactic dependencies they are expected to have — cf. McShane and Nirenburg (2016) for an in-depth overview), their application is deferred until the scoring stage as word sense disambiguation is pursued jointly, as part of meaning representation construction by the algorithm, rather than as a pre-processing stage.

The *link* operation is the core TMR-building routine of the algorithm.

function LINK(S_i)

$S_{i+1} \leftarrow S_i; L_{i+1,i} \leftarrow \emptyset$

for all argument taking p_j in S_{i+1} **do**

$R_j \leftarrow \text{CASE-ROLES}(p_j)$

for all F in ALIGN(S_i, R_j) **do**

$F \Leftrightarrow \{p_k \in S_i \mid \text{SAT}(p_k, r_{j,k} \in R_j)\}$

yield $\langle j, k, r_{j,k}, \text{SCORE}(p_j, p_k, r_{j,k}) \rangle$

end for

end for

return $(S_i, S_{i+1}, L_{i+1,i})$

end function

On each application, it considers a superposition of TMR fragments of depth 1 formed by the elements of the two adjacent partitions S_i and S_{i+1} . The set $L_{i+1,i}^i$ is used to store edges connecting the elements of the two sets. For each argument-taking instance p_j in S_{i+1} , the function attempts to find alignments from roles R_j to instances $F \subset S_i$ that would satisfy their semantic constraints.

For each such alignment, the function yields the resulting edges as tuples containing the end point indices along with the corresponding case role label and link score. In addition to the already mentioned ranking of candidate meaning representations during the solution extraction phase, link scores can also be used to drive beam search during the forward pass to prune the search space on the fly. Scores are currently assigned based on a) how closely the filler matches the role’s type constraint (inverse ontological distance) and b) whether the role matches the expected syntactic

dependency specified for it in the lexicon. A large part of future work will involve incorporation of pragmatic heuristics including those based on co-reference, coherence (e.g., prefer the role agent if a filler in question took on the that role in a series of events in preceding discourse), and knowledge of scripts, all of which we hypothesize especially crucial in situations with unreliable syntax.

The computational complexity of this operation is a little harder to estimate since it depends on variable numbers of word senses and corresponding case roles. Treating them as constant factors gives us quadratic worst-case complexity in the length of the utterance. A more accurate estimate can be obtained but the key point is that the overlapping representation of hypotheses with iterative deepening eliminates the branching factor of exhaustive permutation and results in polynomial rather than exponential order of complexity.

2.3 A Worked Example

We will now proceed with the derivation of the meaning representation for the example sentence.

Although the OntoSem lexicon specifies more fine grain senses of the words used, to keep the example simple, we will only consider a subset. Our description will follow the derivation in Table 1.

1. The algorithm initializes with the empty sets of semantic instances S and case roles links L .
2. Both NO-ACTIONABLE-TMR and MORE-INPUT-NEEDED conditions are triggered, invoking the *next* operation. The first token “Apply” has an instrumental sense e.g., “apply paint to the wall”, but to introduce some degree of ambiguity we will also consider a phrasal sense as in “he needs to apply himself”, having a meaning of the modality of type EFFORT.
3. At the next step, the conditions are triggered again since neither of the current head event candidates have their roles specified. The next token “pressure” instantiates two possible interpretations: literal, physical PRESSURE, and figurative i.e. “to be pressured to do something” translating into a modality of type POTENTIAL.
4. The currently disjoint TMR components (p_1, p_2, p_3, p_4) can now be connected via case

#	Op	u_z	$\Delta S_i, \Delta L_{i+1}^i$	C
1	<i>init</i>	\emptyset		\emptyset
2	<i>next</i>	Apply	p_1 :APPLY, p_2 :EFFORT	p_1 p_2
3-4	<i>next</i> <i>link</i>	pressure	p_3 :PRESSURE, p_4 :POTENTIAL l_4^1, l_4^2, l_2^4 :scope, l_1^3 : instrument, theme	p_1 p_2 p_3 p_4 p_1 p_2 p_3 p_4
5-6	<i>next</i> <i>link</i>	wound	p_5 :WOUND-INJURY l_1^5 :theme	p_1 p_2 p_3 p_4 p_5 p_1 p_2 p_3 p_4 p_5
7-8	<i>next</i> <i>link</i>	!	p_6 :REQUEST-ACTION l_6^1, l_6^2, l_6^4 :theme	p_1 p_2 p_3 p_4 p_5 p_6 p_1 p_2 p_3 p_4 p_5 p_6
9	<i>link</i>	n/a	$L_3^2 \leftarrow L_2^1 \setminus \{l_1^*\}$ $L_2^1 \leftarrow L_2^1 \setminus \{l_6^*\}$	p_1 p_2 p_3 p_4 p_5 p_6 p_1 p_2 p_3 p_4 p_5 p_6 p_1 p_2 p_3 p_4 p_5 p_6
10	<i>extract</i>	n/a	p_6 :REQUEST-ACTION scope p_1 :APPLY instrument theme p_3 :PRESSURE p_5 :WOUND-INJURY	p_1 p_2 p_3 p_4 p_5 p_6 p_1 p_2 p_3 p_4 p_5 p_6 p_1 p_2 p_3 p_4 p_5 p_6

Table 1: Example TMR derivation. The columns respectively indicate: operation performed, current token, instantiated concept or linked semantic role, and the visual representation of the solution lattice.

roles, which triggers the *link* operation, resulting in four compound TMR fragments: (APPLY instrument/theme PRESSURE), (EFFORT scope POTENTIAL), (POTENTIAL scope APPLY), (POTENTIAL scope EFFORT). The first interpretation has a role ambiguity to be resolved during the solution extraction phrase.

5. Neither of the TMR fragments are complete, still having unspecified roles, thereby triggering the *next* operation. The following token “wound” translates into p_8 :WOUND-INJURY concept¹.

6. During the next iteration of linking p_8 :WOUND-INJURY is employed as a theme of p_1 :APPLY resulting in the (APPLY (instrument PRESSURE) (theme WOUND-INJURY)) TMR fragment.

7. The last token “!” confirms the imperative mood of the utterance, signaling the instantiation of the REQUEST-ACTION speech act.

8. The three TMR heads are then linked to the theme slot of of the REQUEST-ACTION instance.

9. Since none of the current fragments alone spans all of the components of the input despite their aggregate coverage, they need to be combined together through the creation of a new layer producing nested TMR fragments. Certain links have been pruned as not leading to novel solutions.

10. The termination condition is fulfilled as there now exists a complete TMR candidate accounting for all of the lexemes observed in the input. It is shown in the table both as a tree and as a highlighted fragment of the solution lattice.

3 Discussion & Related Work

Syntax plays a formidable role in the understanding of meaning. In a number of theories of semantics, important aspects such as case role alignment are determined primarily based on some syntactico-semantic interface such as in Lexical Functional Grammars (Ackerman, 1995), Combinatory Categorical Grammars (Steedman and Baldridge, 2011), or others. Some approaches have even gone as far as to cast the entire problem of semantic parsing as that of decorating a dependency parse (May, 2016). However, such tight coupling of semantics with syntax has its downsides. First, linguistic variation in the expression of the same meaning needs to be accounted for explicitly (e.g., by creating dedicated lexical senses

¹Although a verbal sense is also conceivable, we will assume it has been ruled out by morphology.

or production rules to cover all possible realizations). Second and more importantly, the feasibility of a semantic interpretation becomes conditional on the well-formedness of the language input and the correctness of the corresponding syntactic parse. This requirement seems unnecessarily strong considering a) human outstanding ability to make sense of ungrammatical and fragmented speech (Kong et al., 2014) and b) still considerably high error rates of ASR transcription and parsing (Roark et al., 2006).

The algorithm we present, by contrast, operates directly over the meaning representation space, while relying on heuristics of arbitrary provenance (syntax, reference, coherence, etc.) in its scoring mechanism for disambiguation and to steer the search process towards the promising region of the search space. This allows it to focus on exploring conceptually plausible interpretations while being less sensitive to upstream noise. The algorithm is opportunistic as the overlapping lattice representation allows it to simultaneously pursue multiple hypotheses, while the explicit evaluation of the candidates is deferred until a promising solution is found.

Incremental semantic parsing has come back to the forefront of NLU tasks (Peldszus and Schlangen, 2012), (Zhou et al., 2016). Accounting for incrementality helps to make dialogues more human-like by accommodating sporadic asynchronous feedback as well as corrections, wedged-in questions, etc. (Skantze and Schlangen, 2009). In addition, incrementality of meaning representation construction coupled with sensory grounding has been shown to help prune the search space and reduce ambiguity during semantic parsing (Kruijff et al., 2007). The presented algorithm is incremental as it proceeds via a series of operations gradually extending and deepening the lattice structure containing candidate solutions. The algorithm constantly re-evaluates its partial solutions and potentially produces an actionable TMR without exhaustive search over the entire input.

The design of the algorithm was influenced by the classical blackboard architecture (Carver and Lesser, 1992). Blackboard architectures offer significant benefits: they develop solutions incrementally by aggregating evidence; employ different sources of knowledge simultaneously; and entertain multiple competing or cooperating lines of

reasoning concurrently. Some implementation details of our algorithm were inspired by the well-known Viterbi algorithm: e.g., forward and backward passes, solution scoring (Viterbi, 1967); as well as the relaxed planning graph heuristic from autonomous planning: namely, the lattice representation of partial solutions and the repeated simultaneous application of operators (Blum and Furst, 1995).

4 Conclusion & Future Work

The opportunistic and incremental algorithm we presented in this paper has the potential to improve the flexibility and robustness of meaning representation construction in the face of ASR noise, parsing errors, and unexpected input. This capability is essential for NLU results to approach true human capability (Peldszus and Schlangen, 2012). We implemented the algorithm and integrated it with non-trivial subsets of the current OntoSem’s domain-general lexicon (9354 out of 17198 senses) and ontology (1708 concept realizations out of 9052 concepts) (McShane et al., 2016). Once fully integrated and tuned, we plan to formally evaluate the performance of the algorithm on a portion of the SemEval 2016 AMR parsing shared task dataset (May, 2016) while measuring the impact of parsing errors on the end TMR quality. In addition, it would be interesting to empirically quantify the utility of our proposed pragmatic heuristics in the domain of a task-oriented dialogue such as the Dialogue State Tracking Challenge (Williams et al., 2013).

Acknowledgements

This research was supported in part by Grant N00014-16-1-2118 from the U.S. Office of Naval Research and by the Cognitive and Immersive Systems Laboratory collaboration between Rensselaer Polytechnic Institute and IBM Research. We would also like to thank the three anonymous reviewers for their valuable feedback.

References

- Farrell Ackerman. 1995. Lexical functional grammar. In Jef Verschuren, Jan-Ola Östman, and Jan Blommaert, editors, *Handbook of Pragmatics*, pages 346–350. John Benjamins, Amsterdam.
- Avrim L. Blum and Merrick L. Furst. 1995. Fast planning through planning graph analysis. *Artificial Intelligence*, 90(1):1636–1642.
- Norman Carver and Victor Lesser. 1992. The Evolution of Blackboard Control Architectures.
- Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archana Bhatia, Chris Dyer, and Noah A. Smith. 2014. A Dependency Parser for Tweets.
- Geert-Jan Kruijff, Pierre Lison, Trevor Benjamin, Henrik Jacobsson, and N. Hawes. 2007. Incremental, multi-level processing for comprehending situated dialogue in human-robot interaction. In *Language and Robots: Proceedings from the Symposium (LangRo’2007)*. University of Aveiro, 12.
- Jonathan May. 2016. SemEval-2016 Task 8: Meaning Representation Parsing.
- Marjorie McShane and Sergei Nirenburg. 2016. Decisions for semantic analysis in cognitive systems.
- Marjorie McShane, Sergei Nirenburg, and Stephen Beale. 2016. Language understanding with ontological semantics.
- Andreas Peldszus and David Schlangen. 2012. Incremental Construction of Robust but Deep Semantic Representations for Use in Responsive Dialogue Systems. In Eva Hajiou, editor, *Proceedings of the Coling Workshop on Advances in Discourse Analysis and its Computational Aspects*.
- Brian Roark, Mary Harper, Eugene Charniak, Bonnie Dorr, Mark Johnson, Jeremy G. Kahn, Yang Liu, Mari Ostendorf, John Hale, Anna Krasnyanskaya, Matthew Lease, Izhak Shafran, Matthew Snover, Robin Stewart, and Lisa Yung. 2006. Sparseval: Evaluation metrics for parsing speech.
- Gabriel Skantze and David Schlangen. 2009. Incremental dialogue processing in a micro-domain. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL ’09, pages 745–753, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Mark Steedman and Jason Baldrige, 2011. *Combinatory Categorical Grammar*, pages 181–224. Wiley-Blackwell.
- Andrew Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, April.
- Jason Williams, Antoine Raux, Deepak Ramach, and Alan Black. 2013. The dialog state tracking challenge. In *In Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue SIGDIAL*.
- Junsheng Zhou, Feiyu Xu, Hans Uszkoreit, Weiguang Qu, Ran Li, and Yanhui Gu. 2016. AMR parsing with an incremental joint model. In *EMNLP*.