Generating Discourse Inferences from Unscoped Episodic Logical Formulas

Gene Louis Kim, Benjamin Kane, Viet Duong, Muskaan Mendiratta,
Graeme McGuire, Sophie Sackstein, Georgiy Platonov, and **Lenhart Schubert**
University of Rochester
Department of Computer Science
gkim21,gplatono,schubert@cs.rochester.edu
bkane2,vduong,mmendira,gmcguir2,ssackste@u.rochester.edu

Abstract

Unscoped episodic logical form (ULF) is a semantic representation capturing the predicate-argument structure of English within the episodic logic formalism in relation to the syntactic structure, while leaving scope, word sense, and anaphora unresolved. We describe how ULF can be used to generate natural language inferences that are grounded in the semantic and syntactic structure through a small set of rules defined over interpretable predicates and transformations on ULFs. The semantic restrictions placed by ULF semantic types enables us to ensure that the inferred structures are semantically coherent while the nearness to syntax enables accurate mapping to English. We demonstrate these inferences on four classes of conversationally-oriented inferences in a mixed genre dataset with 68.5% precision from human judgments.

1 Introduction

ULF was recently introduced as a semantic representation that captures the core semantic structure within an expressive logical formalism while staying close enough to the surface language to annotate a dataset that can be used to train a parser (Kim and Schubert, 2019; Kim, 2019). Kim and Schubert (2019) focused on the descriptive power of ULF and its relation to its fully resolved counterpart, Episodic Logic (EL), but the combination of semantic and syntactic information encoded in ULFs should position it to enable certain structurally-driven inferences. In fact, Kim and Schubert (2019) mention some of these inferential classes that they expect ULF will support, but give no description of how to achieve this, nor a demonstration of it in practice.

ULF, being a pre-canonicalized semantic form, makes available many possible structures for similar semantic meanings, which leads to a challenge in formulating generalizable inferences. This pre-canonicalized nature of ULF, though structurally relatively intricate, has some advantages over fully canonicalized representations for use in natural language tasks. One is that it allows direct translation of intuitions about warranted textual inferences into inference rules (much as in Natural Logic). As well, the ability to accurately generate the English sentences corresponding to a ULF formula and choose how and when to modify the surface form allows a more natural interface with the end task. This feature allows us to evaluate inferences generated by ULF directly over English text rather than using an artificially structured interface, such as classification.

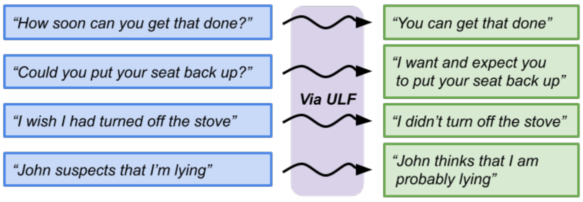
"How soon can you get that done?"
"Could you put your seat back up?"
"I wish I had turned off the stove"
"John suspects that I'm lying"
Via ULF
"You can get that done"
"I want and expect you to put your seat back up"
"I didn't turn off the stove"
"John thinks that I am probably lying"

Figure 1: Examples of the sorts of discourse inferences that we generate via ULFs.

We present a method of generating inferences from ULFs from a small set of interpretable inference rules by first defining general semantic predicates over ULF clauses and tree transformations that correspond to natural semantic operations in ULF. We then evaluate these on four of the five inferential classes presented by Kim and Schubert (2019) over a multi-genre dataset. The ULF structure allows us to incorporate a paraphrase-like rewrite module and then perform direct string comparisons of English generated from ULFs to human generated inferences. Human evaluations show that 68.5% of these generated inferences are acceptable and an error analysis of the system shows that many of the errors can be corrected

Proceedings of the First International Workshop on Designing Meaning Representations, pages 56–65
Florence, Italy, August 1st, 2019 ©2019 Association for Computational Linguistics

with some refinement to the inference rules and the ULF-to-English generation system.

2 Unscoped Episodic Logical Form

ULF is an underspecified variant of EL which captures the predicate-argument structure within the EL type-system while leaving operator scope, anaphora, and word sense unresolved (Kim and Schubert, 2019). All atoms in ULF, with the exception of certain logical functions, syntactic macros, and names are marked with an atomic type, which are written with suffixed tags: `.v`, `.n`, `.a`, `.p`, `.pro`, `.d`, etc. echoing the part-of-speech, such as verb, noun, adjective, preposition, pronoun, determiner, etc., respectively. Some of them contain further specifications as relevant to their entailments, e.g., `.adv-e` for locative or temporal adverbs (implying properties of events). These correspond to particular possible semantic denotations. For example, `.pro` is always an *entity*, `.p` is always a *binary predicate*, and `.v` is an *n-ary predicate*, where *n* can vary. ULF (and EL) uses type-shifting operators to retain type coherence while staying faithful to the semantic types. This is demonstrated in the following example.

(1) *Would you take Tom to Boston with you?*

```
(((pres would.aux-s) you.pro
  (take.v |Tom| (to.p-arg |Boston|)
   (adv-a (with.p you.pro)))) ?)
```

The type shifting operator `adv-a` is necessary in `(adv-a (with.p you.pro))` since this prepositional phrase is acting as a *predicate modifier* in (1), rather than as a predicate (e.g. *"My daughter is with you"*). Constituents in ULF are combined according to their bracketing and semantic types as ULF does not restrict operator ordering in most constructions.

In order to maintain word order and simplify the explicitly modeled structure, ULF includes syntactic *macros* and *relaxations*. ULF *macros* are marked explicitly and reorganize their arguments in a regular manner. For example, `sub` is a macro for moving topicalized constituents to their semantic positions—see the ULF in Figure 4 for an example. ULF *relaxations* are parts of ULFs that are not required to follow the strict operator-operand syntax because their exact position can be deduced otherwise. The subject-auxiliary inversion in (1) is an example of this.

2.1 Expected Inferences from ULF

Here we briefly describe the classes of inferences that Kim and Schubert (2019) propose could be generated with ULF. [1]

Inferences based on clause-taking verbs – For example, *"She managed to quit smoking"* entails that *"She quit smoking"* and *"John suspects that I am lying"* entails *"John believes that I am probably lying"*. Stratos et al. (2011) have demonstrated such inferences using fully resolved EL formulas.

Inferences based on counterfactuals – For example, *"I wish I hadn't forgotten to turn off the stove"* implicates that *the speaker had forgotten to turn off the stove*.

Inferences from questions – For example, *"How soon can you get that done?"* enables the inference that the addressee is able to get that done (in the foreseeable future), and that the questioner wants to know the expected time of completion, and expects that the addressee probably knows the answer, and will supply it.

Inferences from requests – For example, *"Could you put your seat back up a little?"* implies that the speaker wants the addressee to put their seat back up, and expects he or she will do so.

NLog (Natural Logic) inferences based on generalizations and specializations – For example, *"Every dog in the park chased after the squirrel"*, together with the knowledge that Spot was a dog at the park and that a squirrel is an animal entails that *Spot chased after an animal.*

A common feature among all of these inferences is that they are highly dependent on a combination of the predicate-argument and syntactic structures. Also, these are inferences that come naturally and spontaneously to speakers during conversation and are important for generating natural dialogues by setting up the appropriate conversational context.

[1] As ULFs do not resolve operator scope, anaphora, and word sense ambiguity, inferences generated with ULFs will retain these ambiguities. Therefore, the use of these inferences will either need to tolerate such ambiguities, or resolve them in a later step. Later resolution requires keeping track of context of formulas from which conclusions are drawn. For example, say we conclude from *"We know he lied"* that *"He lied"*. Resolving the referent of *"He"* requires the context of the original sentence, which likely disambiguates the person.

"Can somebody help me?" ⇒ *"I want somebody to help me."*

```
(((pres can.aux-v) somebody.pro          (i.pro ((pres want.v) somebody.pro
    (help.v me.pro)) ?)                      (to (help.v me.pro))))
```

Inference Rule

```
(∀a,t,v [[[a aux-indicating-request?] ∧ [t request-personal-pronoun?] ∧ [v verb?] ∧
          (((pres a) t v) ?)]
         → (i.pro ((pres want.v) t (to v)))])
```

Figure 2: An example of an inference rule for inferring an underlying desire from a request. Infixed notation in the inference rule is marked with square brackets for readability. Generalizations and variants of the rule for handling extraneous sentence modifiers, such as *please*, are omitted for clarity.

3 ULF Inference Rules

The inference rules that we define are tree transductions that respect the EL type system in both the antecedent and consequent clauses, ensuring semantic coherence in the concluded formulas. By using high-level predicates and transformations over ULF expressions, these are simple and interpretable at the top level. We use TTT (Purtee and Schubert, 2012) to define our tree-transductions rules as it provides a powerful and flexible way to declare tree transductions and supports custom predicate and mapping functions.

3.1 Named ULF Expression Predicates

The foundation of the interpretable predicates correspond to the ULF semantic types with syntactic features, e.g. `lex-pronoun?` which is true for any atom with a `.pro` suffix—a ULF pronoun. In line with TTT notation, we indicate predicates by ending the name with a question mark, `?`. These are defined over the possible compositions of ULF expressions which includes, for example, `verb?` and `tensed-verb?` that match arbitrary untensed and tensed verb phrases in ULF. This extends to all distinct ULF constituent types: `noun?`, `adv?`, `term?`, `plural-term?`, `sent?`, etc. We supplement these with predicates that correspond to patterns or enumerations of ULFs that correspond specifically to the inference task in question. For example, `aux-indicating-request?` is a predicate that is true for eight ULF auxiliary forms that correspond to a request.[2]

3.2 Named ULF Expression Transformations

High-level tree transformation rules which correspond to natural semantic modifications are also defined and named. These are defined for transformations where the indexical nature and looser syntactic constraints of ULF lead to non-trivial interactions with the syntactic structure. In other words, these rules are indexical and syntax-sensitive variants of simple EL inference rules. This includes rules such as `non-cf-vp!` which transforms a counterfactual verb phrase (VP) to the corresponding factual VP, `negate-vp!` which negates a VP, and `uninvert-sent!` which transforms an subject-auxiliary inverted sentence, e.g. a question, to the uninverted form. We indicate transformation rules by ending the name with an exclamantion mark, `!`. Here are a couple of examples of `negate-vp!` transformations for clarity.

(2) *left the house → did not leave the house*

```
  ((past leave.v) (the.d house.n))
→((past do.aux-s) not
    (leave.v (the.d house.n)))
```

(3) *had met before → had not met before*

```
  ((past perf) meet.v before.adv-e)
→((past perf) not (meet.v before.adv-e))
```

Examples (2) and (3) show that the way negation modifies a ULF verb phrase is dependent on the presence or absence of auxiliaries and aspectual operators (i.e. perfect and progressive aspect). And if this process results in a new head verb, the tense operator would need to be moved accordingly. In order to avoid directly managing these idiosyncratic syntactic phenomena in the inference rules, the VP negation is encapsulated into a single transformation rule.

3.3 Defining Inference Rules

The inferences rules are simple if-then relations defined over a structure where the predicates can appear in the antecedent and the named transformations can appear in the consequent. Figure 2 shows an inference rule for simple requests, written as a universal quantifier over ULF expressions. In practice, this rule is implemented using a TTT

[2] `can.aux-v`, `can.aux-s`, `will.aux-v`, `will.aux-s`, `would.aux-v`, `would.aux-s`, `could.aux-v`, and `could.aux-s`.

tree transduction rule. These rules can be formulated as EL meta-axioms (Morbini and Schubert, 2008) generalized with the named ULF expression predications and transformations to interface with the looser syntax of ULF and its representational idiosyncrasies inherited from English. Since the inferential categories we are exploring are a mixture of entailments, presuppositions, and implicatures their use in a general inference framework warrants additional management of projecting presuppositions and defusing implicatures.

4 Dataset Construction

We chose a variety of text sources for constructing this dataset to reduce genre-effects and provide good coverage of all the phenomena we are investigating. Some of these datasets include annotations, which we use only to identify sentence and token boundaries.

4.1 Data Sources

- **Tatoeba**

  The Tatoeba dataset[3] consists of crowd-sourced translations from a community-based educational platform. People can request the translation of a sentence from one language to another on the website and other members will provide the translation. Due to this pedagogical structure, the sentences are fluent, simple, and highly-varied. The English portion downloaded on May 18, 2017 contains 687,274 sentences.

- **Discourse Graphbank**

  The Discourse Graphbank (Wolf, 2005) is a discourse annotation corpus created from 135 newswire and WSJ texts. We use the discourse annotations to perform sentence delimiting. This dataset is on the order of several thousand sentences.

- **Project Gutenberg**

  Project Gutenberg[4] is an online repository of texts with expired copyright. We downloaded the top 100 most popular books from the 30 days prior to February 26, 2018. We then ignored books that have non-standard writing styles: poems, plays, archaic texts, instructional books, textbooks, and dictionaries. This collection totals to 578,650 sentences.

- **UIUC Question Classification**

  The UIUC Question Classification dataset (Li and Roth, 2002) consists of questions from the TREC question answering competition. It covers a wide range of question structures on a wide variety of topics, but focuses on factoid questions. This dataset consists of 15,452 questions.

4.2 Pattern-based Filtering

As the phenomena that we want to focus on are relatively infrequent, we wrote filtering patterns to reduce the number of human annotations needed to get a sufficient dataset for evaluation. Requests, for example, occur once in roughly every 100 to 1000 sentences, depending on the genre. The filtering is performed by first sentence-delimiting and tokenizing the source texts then matching these tokenized sentences over linguistically augmented regular expression patterns. The filtering patterns are designed for near-full recall of the targeted sentence types by retaining sentences that superficially look like they could be of those types.

The sentence-delimiters and tokenizers are hand-built for each dataset for a couple of reasons. First, general purpose models are likely to fail systematically on our multi-genre dataset and relatively infrequent phenomena, leading to unintended changes in the dataset distribution. Second, the datasets have common patterns and existing annotations which can be exploited in a hand-built system. For example, the Discourse Graphbank follows the ends of sentences with a newline and in the Tatoeba and UIUC datasets each line is a sentence. The transparency of the rules also have the benefit that we can interpretably fix errors in their performance in the future.

These filtering patterns are written in augmented regex patterns. Figure 3 shows two such augmented regex patterns for plain and inverted if-then counterfactual constructions. The regexes are augmented with *tags* written in angle brackets, e.g. `<begin?>`. These tags refer to regex fragments that are reusable and conceptually coherent to people. `<begin?>` matches either the beginning of the string or space separated from previous text. `<mid>` matches words that are padded with spaces on the sides (i.e. separate tokens from what's defined next to it) and `<mid?>` is a variant that allows just a space as well. `<past>` and `<ppart>` are alternative lists of past tense and past participle verb forms. `<futr>` is an alternatives list of different

[3] https://tatoeba.org/eng/

[4] https://www.gutenberg.org

Basic if-then `"<begin?>(if|If)<mid>(was|were|had|<past>|<ppart>)<mid?>(<futr>) .+"`

If I thought this would make it difficult for the family , I would n't do it , " he said . – Discourse Graphbank

Inverted if-then `"<begin?>(<futr>)<mid>if<mid>(was|were|had|<past>|<ppart>) .+"`

Tom would n't have married Mary if he 'd known she had spent time in prison . – Tatoeba

Figure 3: Example shorthand regex patterns (Section 4.2) for filtering candidate sentences with matching sentences.

conjugations of *"will"*. Tags for closed classes of words and shorthands for common non-word patterns were hand-curated. Tags for open classes such as `<past>` and `<ppart>` are generated from the XTAG morphological database (Doran et al., 1994) with minor edits during the development process.

4.3 Sentence selection

After performing filtering, we still have far too many sentences to feasibly annotate, so we build a balanced set of 800 sentences split evenly among the four sentence types we filtered for, *clause-taking verbs*, *counterfactuals*, *requests*, and *questions*. For each sentence type, we select the sentence round-robin between the four datasets to balance out the genres. Some types of sentences appear more that 200 times in this sampling because some sentences pass multiple filters. For example, *"Could you open the door?"* passes both the *request* and *question* filters.

4.4 Inference Annotation

As we discuss in Section 7, evaluating automated inferences effectively is a major challenge. Every sentence leads to many inferences at various levels of discourse, certainty, and context-dependence. This is exacerbated by the ability to paraphrase the inferred statements. By limiting ourselves to inferences of particular general structures, we are able to elicit natural responses from people that are restricted to the particular phenomena that we are interested in investigating.

The annotations are separated into the same four categories as the filtering: *clause-taking verbs*, *counterfactuals*, *questions*, and *requests*. The annotator is first asked to select the structural inference pattern that holds for the given sentence and write down the corresponding inferred sentence. For example, say there is the sentence *"If I were rich, I would own a boat"*. The annotator would select an inference template along the lines of `(if <x> were <pred>, <x> would <q>)` → `(<x> is not <pred>)` and write down the inference "I am not rich". This way we can get a fluent inference, but push the annotator to think about the inferences structurally. The annotators are additionally instructed to keep the inference as fluent as possible, preserve the original sentence as much as possible, and keep the perspective of the speaker of the sentence. We also included an option for annotators to add new rules, to extend the dataset into categories we did not anticipate. This category will be referred to as *Other*.

The annotations were performed by members of our research group, including some of the authors. These were completed before starting the development of the inference system. There is the possibility of development being skewed by knowledge of the annotated data, but we expect this factor to be quite small since the core inference system was built by only a couple of the annotators and the bulk of this development was done several months after completion of the annotations. The annotations totaled 698 inferences from 406 sentences.[5]

5 Evaluation

We developed the inference rules based on a set of 40 sentences randomly sampled from the annotated dataset. The correctness of these inferences is evaluated both through an automatic evaluation over the whole dataset and a human evaluation of a sample of the inferences. Both evaluations are done directly over English sentences by automatically translating the ULF inferences to English sentences. The automatic evaluation also involves a ULF rewriting module to handle semantically equivalent inference variants. All of these components are fine-tuned on the 40 sentence devset. In all of the experiments we start with human ULF annotations as a reliable ULF parser is not yet available.[6]

[5] This is half of the original 800 sampled sentence after filtering sentences that had duplicates due to dataset artifacts we failed to notice at the sentence selections stage and sentences that could not be annotated given the current ULF guidelines.

[6] Kim (2019) reports some promising preliminary results on parsing ULFs.

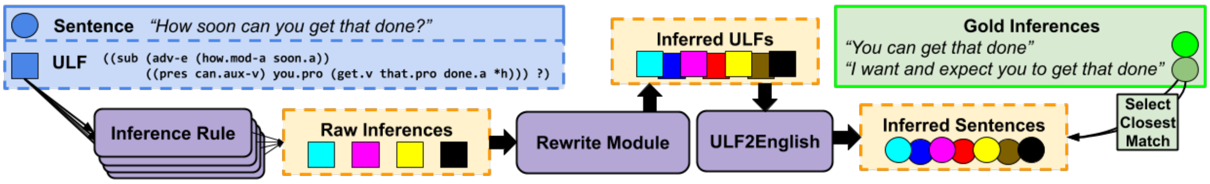
Sentence “How soon can you get that done?”
ULF ((sub (adv-e (how.mod-a soon.a))
((pres can.aux-v) you.pro (get.v that.pro done.a *h))) ?)
Inference Rule
Raw Inferences
Rewrite Module
Inferred ULFs
ULF2English
Inferred Sentences
Gold Inferences
“You can get that done”
“I want and expect you to get that done”
Select Closest Match

Figure 4: A diagram of the automatic ULF inference evaluation pipeline.

5.1 ULF to English

The ULF-to-English translation is done in a simple multi-stage process of

1. Analyzing the ULF type of each clause,
2. Incorporating morphological inflections based on the type analysis,
3. Filtering out purely logical operators, and
4. Mapping logical symbols to surface form counterparts.

The closeness of ULF to syntax and its preservation of most word-ordering makes hand-building a robust function for this reasonably simple. The verb conjugations and noun pluralizations are performed using the `pattern-en` python package. The code for mapping ULFs to English is available at `https://github.com/genelkim/ulf2english`.

5.2 Rewriting Rules

The rewriting rules capture alternative ways to represent the same sentence without changing the meaning. This includes clausal restructuring (e.g. *“I expect that you come here”* to *“I expect you to come here”* or *“I expect you come here”*), merging inferences (e.g. *“I want you to get that done”* and *“I expect you to get that done”* to *“I want and expect you to get that done”*) and others of this sort that are extremely unlikely to change the meaning of the sentence.

5.3 Automatic Inference Evaluation

A diagram of the automatic evaluation pipeline is presented in Figure 4. The pipeline for a given source sentence and ULF proceeds as follows:

1. Use the inference rules (Section 3) to generate a set of raw inferences from the source ULF.
2. Generate a complete set of possible realizations of the inferred ULFs by rewriting the raw inferences into possible structural variations (Section 5.2).
3. Translate inferred ULFs into English to get a set of inferred sentences (Section 5.1).
4. For each human inference elicited from the current source sentence, find the system-inferred sentence that has the smallest edit distance.
5. Report recall over human inferences with a max edit distance threshold.[7]

We use an edit distance threshold of 3. This allows minor English generation errors such as verb conjugations and pluralizations, but does not allow simple negation insertion/deletion (a difference of a space-separated *“not”* token). Table 1 lists the results of this evaluation. The numerical values are fairly low, but this may be expected given the evaluation procedure. A trivial baseline such as most frequent devset inference or copying the source sentence would lead to a score of 0 or very close to 0 as these are very unlikely to be within a 3-character edit from the inferences in the dataset.

5.4 Human Inference Evaluation

The human inference evaluation was performed over 127 raw ULF inferences. This was built out of 100 randomly sampled inferences with the addition of every counterfactual and clause-taking inference as they are not as common. Each inference was then translated to English, then presented alongside the source sentence to 3 to 4 independent human judges. The judges evaluated correctness of the discourse inference and the grammaticality of the output sentence. Table 2 presents the results of this. 87 of the 127 inferences were marked as correct by a majority of judges and only 21 were marked as incorrect by a majority of judges, for the remaining 19 inferences judges either disagreed completely or a majority judged it as context-dependent. 99 of the 127 inferences were judged grammatical by a majority of judges, which demonstrates the efficacy of the ULF-to-

[7] We do not report precision over automatic inferences because missing inferences are common in our dataset. This could be alleviated in the future by explicitly splitting the inference elicitation task into smaller subtasks and/or incorporating a reviewing stage where initial inferences are reviewed, corrected, and possibly added to by a second person.

	cf	cls	req	q	oth	all
Recall	1/13 (8%)	1/33 (3%)	33/97 (34%)	69/316 (22%)	7/130 (5%)	112/662 (18%)

Table 1: Results of automatic inference evaluation described in Section 5.3. **cf** stands for counterfactual inferences, **cls** for clause-taking, **req** for request, **q** for question, **oth** for other.

	cf	cls	req	q-pre	q-act	oth	all
Correct*	11/27	2/5	17/19	13/21	31/39	13/16	68.5%
Incorrect*	9/27	3/5	0/19	3/21	3/39	3/16	16.5%
Context*	7/27	0/5	2/19	5/21	5/39	0/16	15.0%
Grammar	20/27	1/5	19/19	12/21	33/39	14/16	78.0%

Table 2: Results of majority human evaluation of system generated inferences. Evaluation on 127 inferences with from the test set by 3 or 4 people per inference. **Correctness is evaluated on whether the sentence is a reasonable inference in conversation, allowing for some awkwardness in phrasing. Context, means the correctness is highly context-dependent.* The inference type labels in the header row are the same as in Table 1 except for the addition of breaking down questions to **q-pre** for question presuppositions and **q-act** for question act inferences.

English translation system.[8] The system seems to struggle most with counterfactual and clause-taking inferences.

5.5 Evaluation of Rewriting Rules

In order to verify that the rewriting rules in fact preserve the semantic meanings, we gathered a sample of 100 system-inferred sentences that were closest to a gold inference (step 4 in Section 5.3). Each inferred sentence is judged as whether it is a valid rewrite of one or more of the raw inferences. A valid rewrite does not introduce new semantic information. 91 out of the 100 were judged as valid by a majority of three human judges. As such, the rewriting system is not abusively over-generating sentences that are semantically different and match to gold inferences, increasing the recall score.

6 Analysis and Discussion

The human inference evaluation (Section 5.4) showed that the system struggles most with counterfactual and clause-taking verb constructions. This is largely because the sampling procedures for these constructions are not as effective, leading to fewer positive examples in our dataset. In turn, our development set of 40 sentences only included a handful of examples of each inference, so the rules remained brittle after adjusting to the development set. In fact, two of the three incorrect clause-taking verb inferences are a result of a simple mistake of allowing arbitrary terms rather than only reified sentences and verbs in the antecedent.

Some of the automatic inferences were impossible to handle using our inference rules because of disagreements among human elicited inferences on what circumstances warrant particular inferences and how precisely an inference should be expressed. For example, the distinction between the presence or absence of the word “probably” is best handled with a separate confidence metric. In conversations, the distinction between highly probable statements and simply true statements is blurred. One could choose to include or omit “probably" for statements where the possibility of the plain sentence being false is small. Still, we would not want to add this as a rewriting rule since strictly speaking, such hedges do affect the meaning. Similarly, human elicited inferences disagreed on whether requests warrant a question act inference (e.g. *“Could you open the door?”* → *“You know whether you could open the door”*). We opted to avoid generating these inferences in building our rules, which significantly affected the recall score in the automatic evaluation.

The ULF-to-English generation system is remarkably accurate given its fairly simple pipeline approach and given that this is the first real use of this system. 78% grammaticality shows room for improvement and a cursory review of the errors show that there are some ULF macros that still need handling and that verb conjugations need to be made more robust.

Given these results, improvements to the filtering system for counterfactual and clause-taking verb constructions, gathering a larger dataset with a more robust collection procedure, and another set of experiments with the larger dataset would be valuable next steps in more precisely measuring the use of ULF in generating discourse inferences.

7 Related Work

Inference demonstrations have been performed in the past for various semantic representations, showing their respective strengths. Discourse

[8] Some inferences marked as ungrammatical were also marked as correct, indicating that the ULF-to-English failures can be minor enough to be easily understood.

Representation Structures and Minimal Recursion Semantics (MRS) can both be mapped to FOL and run on FOL theorem provers (Kamp and Reyle, 1993; Copestake et al., 2005). MRS has been successfully used for the task of recognizing textual entailment (RTE) (Lien and Kouylekov, 2015). Similarly, EL has been shown successful in generating FOL inferences (Morbini and Schubert, 2009) and self-aware metareasoning (Morbini and Schubert, 2011). Abstract Meaning Representation (Banarescu et al., 2013) focuses on event structure, resolution of anaphora, and word senses rather than logical inference and has been demonstrated to support event extraction and summarization (Rao et al., 2017; Wang et al., 2017; Dohare et al., 2017). TRIPS LF (Allen, 1994; Manshadi et al., 2008) is an unscoped modal logic directly integrated with a lexical ontology and has been used for dialogue and biomedical event extraction (Perera et al., 2018; Allen et al., 2015). Distributional representations have been shown to be very effective for RTE, such as in the SNLI and MultiNLI datasets (Bowman et al., 2015; Williams et al., 2018). These datasets are much larger than previous RTE datasets and both provide classification tasks supporting the use of an implicit distributional representation in a neural network system. The discourse inferences we demonstrated with ULFs, which require access to some syntactic information, as well our evaluations based on reliable English generation, are a challenge to all of the semantic representations discussed, because of their relative remoteness from syntax.

In the realm of evaluation methods, our work has similarities with the TAC KBP slot-filling task, which defines specific types of information that the system is meant to extract from the text without knowledge of the possible correct answers (Ellis et al., 2015). But TAC KBP focuses on restricted types of factoids, whereas our evaluation focuses on structure-based sentential inferences. In recent years inference evaluations have typically been posed as either a classification tasks similar to RTE (Bowman et al., 2015; Williams et al., 2018) or multiple-choice question answering (Clark et al., 2018). This knowledge of possible alternatives allows systems to avoid modeling inferences explicitly and to exploit statistical artifacts. The inference model trained on the ATOMIC commonsense dataset was evaluated without providing a set of possible choices by using BLEU (Sap et al., 2019). Though BLEU scores tend to correlate with correct inferences in practice, using it as a metric of evaluation is fraught with danger. Small changes that dramatically alter the meaning of a sentence (e.g., negation) are not reflected in the BLEU scores, and for structurally oriented inferences, incorrect inferences are likely to have misleadingly high scores.

8 Conclusions

We presented the first known method of generating inferences from ULF and an evaluation of inferences, focusing on discourse inferences. We also presented a method of collecting human elicitations of restricted categories of structural inferences, allowing a novel forward inference evaluation. We used these elicited inferences to automatically evaluate the generated inferences with promising results. Human judgments on a sample of generated inferences showed that 68.5% of the inferences are reasonable discourse inferences, 16.5% were unreasonable, and 15% were context-dependent or had disagreements between judges. Our experiments also demonstrate some of the advantages of using a semantic representation closer to the syntactic form such as ULF—reliable translation to English and access to syntactic signals—though this comes at the cost of a more complicated interface with the semantic patterns. There are clear areas of future work on improving the human elicitation collection and the implementation of the inference system. A larger and more refined dataset of inference elicitations will likely allow the development of a robust inference system on the discourse inference categories in question.

References

James Allen, Will de Beaumont, Lucian Galescu, and Choh Man Teng. 2015. Complex event extraction using DRUM. In *Proceedings of BioNLP 15*, pages 1–11, Beijing, China. Association for Computational Linguistics.

J.F. Allen. 1994. *Natural Language Understanding*, second edition. Benjamin Cummings, Redwood City, CA, USA.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with*

Discourse, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal Recursion Semantics: An introduction. *Research on Language and Computation*, 3(2):281–332.

Shibhansh Dohare, Harish Karnick, and Vivek Gupta. 2017. Text summarization using abstract meaning representation.

Christy Doran, Dania Egedi, Beth Ann Hockey, B. Srinivas, and Martin Zaidel. 1994. Xtag system - a wide coverage grammar for english. In *COLING 1994 Volume 2: The 15th International Conference on Computational Linguistics*.

Joe Ellis, Jeremy Getman, Dana Fore, Neil Kuster, Zhiyi Song, Ann Bies, and Stephanie Strassel. 2015. Overview of linguistic resources for the TAC KBP 2015 evaluations: Methodologies and results. In *Proceedings of the Eighth Text Analysis Conference (TAC 2015)*.

Hans Kamp and Uwe Reyle. 1993. *From discourse to logic*. Kluwer, Dortrecht.

Gene Kim and Lenhart Schubert. 2019. A type-coherent, expressive representation as an initial step to language understanding. In *Proceedings of the 13th International Conference on Computational Semantics*, Gothenburg, Sweden. Association for Computational Linguistics.

Gene Louis Kim. 2019. Towards parsing unscoped episodic logical forms with a cache transition parser. In *the Poster Abstracts of the Proceedings of the 32nd International Conference of the Florida Artificial Intelligence Research Society*.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

Elisabeth Lien and Milen Kouylekov. 2015. Semantic parsing for textual entailment. In *Proceedings of the 14th International Conference on Parsing Technologies*, pages 40–49, Bilbao, Spain. Association for Computational Linguistics.

Mehdi Hafezi Manshadi, James Allen, and Mary Swift. 2008. Toward a universal underspecified semantic representation. In *13th Conference on Formal Grammar (FG 2008), Hamburg, Germany*.

Fabrizio Morbini and Lenhart Schubert. 2008. Metareasoning as an integral part of commonsense and autocognitive reasoning. In *AAAI-08 Workshop on Metareasoning*.

Fabrizio Morbini and Lenhart Schubert. 2009. Evaluation of Epilog: A reasoner for Episodic Logic. In *Proceedings of the Ninth International Symposium on Logical Formalizations of Commonsense Reasoning*, Toronto, Canada.

Fabrizio Morbini and Lenhart Schubert. 2011. Metareasoning as an Integral Part of Commonsense and Autocognitive Reasoning. In Michael T. Cox and Anita Raja, editors, *Metareasoning: Thinking about thinking*. MIT Press.

Ian Perera, James Allen, Choh Man Teng, and Lucian Galescu. 2018. A situated dialogue system for learning structural concepts in blocks world. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 89–98, Melbourne, Australia. Association for Computational Linguistics.

Adam Purtee and Lenhart Schubert. 2012. TTT: A tree transduction language for syntactic and semantic processing. In *Proceedings of the Workshop on Applications of Tree Automata Techniques in Natural Language Processing*, ATANLP '12, pages 21–30, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sudha Rao, Daniel Marcu, Kevin Knight, and Hal Daumé III. 2017. Biomedical event extraction using abstract meaning representation. In *BioNLP 2017*, pages 126–135, Vancouver, Canada,. Association for Computational Linguistics.

Maarten Sap, Ronan LeBras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. ATOMIC: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*. AAAI Press.

Karl Stratos, Lenhart K. Schubert, and Jonathan Gordon. 2011. Episodic Logic: Natural Logic + reasoning. In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development (KEOD)*.

Yanshan Wang, Sijia Liu, Majid Rastegar-Mojarad, Liwei Wang, Feichen Shen, Fei Liu, and Hongfang Liu. 2017. Dependency and amr embeddings for drug-drug interaction extraction from biomedical literature. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics*, ACM-BCB '17, pages 36–43, New York, NY, USA. ACM.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Florian Wolf. 2005. *Coherence in natural language : data structures and applications*. Ph.D. thesis, Massachusetts Institute of Technology, Dept. of Brain and Cognitive Sciences.