A (Mostly) Symbolic System for Monotonic Inference with Unscoped Episodic Logical Forms

Gene Louis Kim[1], Mandar Juvekar[2], Junis Ekmekciu[3], Viet Duong[4], and Lenhart Schubert[5]
University of Rochester
Department of Computer Science
{gkim21[1],schubert[5]}@cs.rochester.edu
{mjuvekar[2],jekmekci[3],vduong[4]}@u.rochester.edu

Abstract

We implement the formalization of natural logic-like monotonic inference using Unscoped Episodic Logical Forms (ULFs) by Kim et al. (2020). We demonstrate this system's capacity to handle a variety of challenging semantic phenomena using the FraCaS dataset (Cooper et al., 1996).These results give empirical evidence for prior claims that ULF is an appropriate representation to mediate natural logic-like inferences.[1]

1 Introduction

A monotone function between partially ordered sets either preserves or inverts the ordering of argument values. More precisely, a function f is said to be upward monotone if $x \leq y$ implies $f(x) \leq f(y)$. Similarly, f is said to be downward monotone if $x \leq y$ implies $f(x) \geq f(y)$. If neither of these hold, f is said to be non-monotone. When used in the context of subset relations and entailment, monotonicity can be a tool for making natural language inferences. For instance, consider the second example in fig. 1. *Never* is downward monotone in entailment, since it flips the entailment ordering of (1) *I had a girlfriend taller than me before* entails (2) *I had a girlfriend before* to (2) ***I never*** *had a girlfriend before* entails (1) ***I never*** *had a girlfriend taller than me before*. Natural logic is an approach to generating natural language inferences based on syntactic structure and knowledge of the semantic properties of the lexical items and local constructions (Van Benthem et al., 1986; Sánchez-Valencia, 1991). An important fragment of natural logic is monotonicity calculus which operates using syntactic structure and the knowledge of polarity inducing elements and monotonicity relationships. Figure 1 shows the three basic cases of monotonicity inference, upward, downward, and non- monotone contexts leading to different entailment conditions.

Up (FraCaS)	*Some delegates (finished the survey on time)*▲ ⇒ *Some delegates finished the survey*
Down (MED)	*I never had a (girlfriend)*▼ *before* ⇒ *I never had a girlfriend taller than me before*
Non- (MED)	*Exactly 12 aliens read (magazines)*■ ⇏ *Exactly 12 aliens read (news magazines)*■

Figure 1: Upward (...)▲, downward (...)▼, and non-monotone (...)■ examples from the FraCaS and MED datasets.

Episodic Logic (EL) is an extended first-order logic designed to closely match the form and expressivity of natural language (Schubert, 2000). Unscoped Logical Form (ULF) is an underspecified form of EL. ULF completely specifies the semantic type structure of EL, but leaves scope, anaphora, and word sense unresolved (Kim and Schubert, 2019a). Kim and Schubert (2019b) proposed that ULF is suitable for five classes of inferences, namely monotonic inferences, inferences based on clause-taking verbs, inferences based on counterfactuals, inferences from questions, and inferences from requests. Kim et al. (2019) experimentally demonstrated the capacity of ULF to generate all of those classes of inferences except monotonic inference. Kim et al. (2020) presented a proof-based formalism for natural logic-like monotonic inference for ULF. They established a correspondence between their formalism and the natural logic treatment of Sánchez Valencia (1991), and showed that the formalism was capable of handling foundational natural logic inferences from the prior literature. We present an implementation of Kim et al.'s (2020) monotonic inference formalism and give empirical evidence for the feasibility of using ULFs as a basis for making natural logic-like

[1]The code is made available at https://github.com/genelkim/ulf-fracas.

inferences. Our system achieves a high precision on monotonicity problems using a small number of sound inference rules on a variety of inference cases. We thereby complete the work of Kim et al. (2019) in experimentally demonstrating that ULF is in fact capable of handling the five kinds of inference outlined by Kim and Schubert (2019b).

2 Background

Kim et al. (2019) demonstrated the capacity to use ULFs to generate inferences from clause-taking verbs, counterfactuals, questions, and requests while focusing on discourse-contexts that regularly give rise to these phenomena. They generated forward inferences from manually annotated ULFs using symbolic meta-axioms generalized to handle syntactic idiosyncrasies and achieved reasonable precision on a multi-genre dataset. Our work seeks to complement this by generating Natural Logic-like inferences from ULFs. Furthermore, we start our inferences from English using a symbolic transducer from English constituency parses and expand the scope of inferences to enable automatic evaluation on pre-constructed datasets.

2.1 Theoretical Inference Method

Kim et al. (2020) present a proof-based inference method which uses ULF as the base semantic representation. Polarities are computed respective to specific scopings of ULFs—in the form of scoped logical forms (SLFs)—then propagated back to the ULFs to enable inferences that are contingent on the polarity context. This method includes inference rules for ULFs that correspond directly to inference rules in Sánchez-Valencia's (1991) formulation of Natural Logic. The most notable inference rules are

Monotonicity (UMI)

$$\frac{\phi[P1^{\blacktriangle}],\ ((\text{every.d}\ P1)\ (\text{be.v}\ (=\ (\text{a.d}\ P2))))}{\phi[P2]}\,,$$

$$\frac{\phi[P2^{\blacktriangledown}],\ ((\text{every.d}\ P1)\ (\text{be.v}\ (=\ (\text{a.d}\ P2))))}{\phi[P1]}$$

Conversion (UCI)

$$\frac{((d1\ P)\ (\text{be.v}\ (=\ (d2\ Q))))}{((d1\ Q)\ (\text{be.v}\ (=\ (d2\ P))))}\quad \text{where } d1 \in \{\text{some.d, a.d, no.d}\} \text{ and } d2 \in \{\text{some.d, a.d}\}.$$

Polarity contexts that are necessitated by operators present in the formulas are omitted for clarity, e.g., every.d imposes a negative polarity on its restrictor and a positive polarity on its body. The remaining inference rules are *Polarity Marking* and *Negation Introduction/Elimination*.

Below is an simple inference example from the FraCaS dataset—the actual output of our system—which demonstrates a simple use of the UMI inference rule.[2] This example also shows some differences between our system and the original theoretical method presented by Kim et al. (2020). Namely, our UMI rules generalize to variants of *every A is a B* (in this case *all As are Bs*), our initial polarity marking method circumvents the need for SLFs (Sections 3.3 and 3.4), and we haves rules to generate monotonicity relations from intersective predicate modification (Section 3.4).

Inference Example (FraCaS Problem 24)

1.	((many.d (plur delegate.n)) ((past obtain.v) (k (interesting.a (plur result.n))) (adv-a (from.p (the.d survey.n)))))	Assumption
2.	((all.d (interesting.a (plur result.n))) ((pres be.v) (= (k (plur result.n)))))	Inter. modifier relation, 1.
3.	((many.d (plur delegate.n)$^{\blacksquare}$) ((past obtain.v) (k (interesting.a (plur result.n))$^{\blacktriangle}$) (adv-a (from.p (the.d survey.n)))))	Pol marking 1.
4.	((many.d (plur delegate.n)) ((past obtain.v) (k (plur result.n)) (adv-a (from.p (the.d survey.n)))))	UMI 2.,3.

For the syntactic conventions of ULF, such as the type-designating suffixes '.d', '.v', and '.n', see the descriptions provided by Kim and Schubert (2019b) or Kim et al. (2020).

2.2 Automated Monotonicity Inference

Building computational approaches to natural logic inference—distinct from general natural language inference—is an active area of research (Angeli and Manning, 2014; Tian et al., 2014; Mineshima et al., 2015; Abzianidze, 2016; Hu et al., 2019; Haruta et al., 2020). In order to evaluate our monotonicity-specific inference system fairly, we focus on the FraCaS dataset (Cooper et al., 1996) which carefully presents monotonicity-based entailments, for evaluation, and aim to show competence on monotonicity, rather than state-of-the-art (SOTA) performance. In our experiments (Section 5) we compare against a few notable systems that were previously evaluated on the same parts of the FraCaS dataset: Mineshima et al. (2015), Abzianidze (2016), Hu et al. (2019), and Haruta et al. (2020).

[2] Irrelevant polarity marking symbols are omitted for brevity and clarity.

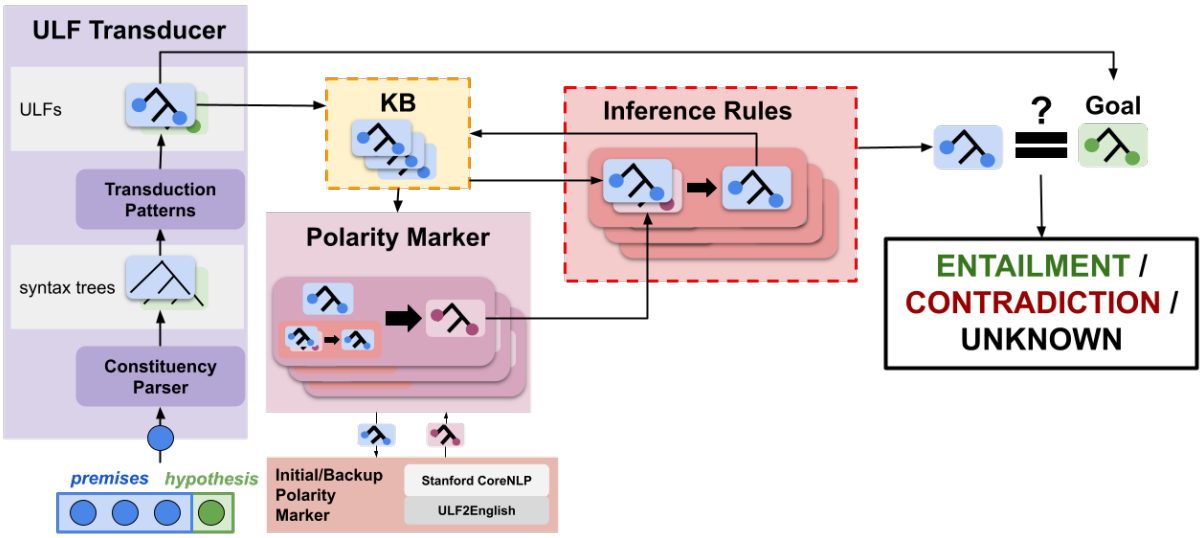

Figure 2: A diagram of the inference system component dependencies.

Mineshima et al. (2015) and Abzianidze (2016) extend first-order lambda logical forms with higher-order terms (e.g most, many, half of, etc.) and augment first-order inference with rules geared towards those terms. Haruta et al. (2020) achieve SOTA performance by employing degree and event semantics to approximate key higher-order logic features presented in different linguistic phenomena. Hu et al. (2019) differs from the others by running directly on the natural language text, with a combinatory categorical grammar (CCG)-based monotonicity labeling system.

Our approach most resembles Hu et al.'s (2019) system because our logical form closely matches the form and expressiveness of natural language, which enables monotonic reasoning using a relatively compact set of inference rules and we also use an auxiliary representation to obtain monotonicity labels. Our goal is not to achieve the SOTA performance on this dataset; rather the SOTA system results are provided to contextualize our system's performance with respect to the wider research efforts on this dataset.

3 System Description

Our inference system starts with a set of premise sentences and a hypothesis sentence in English which are automatically converted to ULF and then used to determine an *entailment*, *contradiction*, or *unknown* relationship between the premises and the hypothesis through a forward inference search from the premises. The inference process is modeled after the theoretical framework described by Kim et al. (2020) which uses SLFs for identifying the polarity operator scopes and computing the global polarity context of each sub-expression. These polarities are then mapped back to the corresponding ULFs which are used as the basis for the inference rules.

We simplify Kim et al.'s (2020) framework in two ways. First, we do not include the scoping possibilities in the proof process. That is, we compute a single SLF for each ULF and assume that it is the correct scoping. Second, we introduce variations of the monotonicity and conversion inference rules that correspond to ULF macros and specific syntax. These reduce two steps of inference (expanding the macro and then applying the inference rule) to a single step. Both of these simplifications are introduced to reduce the search space and speed up the inference process.

Here we describe an example of the second simplification. We directly extract the monotonicity relation from the complex expression containing the nominal predicate with its premodifiers and postmodifiers (i.e., all but the determiner or kind-forming operator of the term derived from a noun phrase). In postnominal modification, the operands of the n+preds macro—combining a nominal predicate with postmodifying predicates—are used directly for inference, without expansion of the macro construct to a conjunctive lambda predicate. For example, for a postmodified noun, as in the phrase *a dog in the park*, we directly extract the entailments *every dog in the park is a dog* and *every dog in the park is in the park*, rather than first

converting the phrase to *something that is a dog and that is in the park* and indirectly computing the rules via explicit predicate intersection. For an intersective prenominal modifier, as in the phrase *a happy dog*, we directly extract the entailments *every happy dog is a dog* and *every happy dog is happy*, rather than first converting the phrase to *something that is happy and that is a dog* first.

The inference system has the following high-level components:

- a heuristic-based inference search function
- a constituency parse to ULF tree transducer
- a global polarity marking function
- inference rules with polarity propagators
- external knowledge resources

Figure 2 shows a diagram of the component dependencies. While most of the inference system is symbolic, the initial constituency parses and initial polarity marking—used for ULF transduction and scope selection, respectively—are computed using NN and ML methods. Furthermore, the ML-based polarity marking is used when the symbolic polarity propagation methods fail or take too long.

3.1 Search Process

Our inference process is guided by a simple heuristic forward search. Algorithm 1 describes this process in detail. In order to retain completeness while using fast and naive heuristic functions, the search process alternates between heuristic guided search and breadth-first search every several steps. This is a generic search process, where h is a heuristic function which estimates the distance from some formula, x, to the goal formula, ϵ is a small positive number which is used to gives preference formulas reached earlier in the search process in cases of ties, and c is the number of search steps in a row that the search process uses heuristic search before switching to BGS and vice-versa. Section 4 specifies the values of these parameters that we use in our experiments.

3.2 ULF Transducer

The ULF transducer converts constituency parses into ULFs with a series of simple correspondences from the phrase structure and POS tags to ULF expressions. This technique is the same as those used in the initial stages of prior transduction-based EL parsers (Schubert, 2002; Schubert and Tong, 2003; Gordon and Schubert, 2010; Schubert, 2014), but modified for Kim and Schubert's (2019b) modern

Algorithm 1 Heuristic search. Inference rules map a set ULF premises to a set of ULF inferences.

Inputs: Φ, a set of premises; ψ, a goal ULF; h, a heuristic function; M, a search depth limit.
Outputs: The entailment classification.
Global Constants: U, a list of unary rules; B, a list of binary rules; ε, a small positive number; c, a step count for search method change.
Procedure:
Initialize $n \leftarrow 0$, KB $\leftarrow \Phi$.
Initialize $Q_{\text{h}} \leftarrow$ empty priority queue.
Initialize $Q_{\text{bfs}} \leftarrow$ empty basic queue.
Initialize $Q \leftarrow Q_{\text{h}}$.
Initialize $Q_{\text{other}} \leftarrow Q_{\text{bfs}}$.
loop
  If $n > M$ or $Q = \varnothing$, **return** UNKNOWN.
  If $\psi \in$ KB, **return** ENTAILMENT.
  If $\neg\psi \in$ KB, **return** CONTRADICTION.
  $\nu \leftarrow Q.\text{pop}()$.
  $t_{\text{unary}} \leftarrow U \times \nu$.
  $t_{\text{binary}} \leftarrow B \times ((\nu \times \nu) \cup (\nu \times \text{KB}) \cup (\text{KB} \times \nu))$.
  Push all results x of computing the tuples in t_{unary} and t_{binary} that are not contained in KB to Q_{h} with key $h(x) + n\varepsilon$ and Q_{bfs}.
  KB $\leftarrow$ KB $\cup\, \nu$.
  $n \leftarrow n + 1$.
  if $n \mod c = 0$ **then**
    tmp $\leftarrow Q$.
    $Q \leftarrow Q_{\text{other}}$.
    $Q_{\text{other}} \leftarrow$ tmp.
  end if
end loop

ULF specification. Some transduction rules add type assumptions that are not necessarily true, but are unlikely to affect the monotonicity inferences. For example, ULF makes a semantic distinction between event modifiers (e.g. *today*) and proposition modifiers (e.g. *surprisingly*) which is not relevant for monotonicity inferences. If the parser fails to eliminate one of these options, it assumes that it is an event modifier.[3]

We use the Berkeley neural parser (Kitaev and Klein, 2018) to get the constituency trees.[4] A neural network-based ULF parser has recently become

[3]The transduction rules are written in a combination of the tree-to-tree transduction language (Purtee and Schubert, 2012) and a simplified variant.

[4]The version 0.2.0 release and the `benepar_en3` model available at https://github.com/nikitakit/self-attentive-parser/.

available (Kim et al., 2021), but we opted not to use it because sentences in monotonicity datasets tend to be fairly short and follow written English syntax. Kim et al.'s (2021) parser is more robust to language length and variety. However, for our evaluation datasets we found a symbolic transduction to be more reliable. Additionally, our symbolic transductions have more predictable and regular errors. This allows monotonicity inferences to succeed even with minor errors.

3.3 Polarity Marking

We delegate the initial polarity marking problem to a component of the Natlog and NaturalLi systems (MacCartney and Manning, 2008; Angeli and Manning, 2014) which runs over raw English text.[5] We then align the polarities of each token to the corresponding ULF sub-expression. Rather than using the actual English premises and hypothesis we use the output of the ULF2English system (Kim et al., 2019) so that we can use its subroutines to assist in subexpression alignment.

This alignment is then used to select the scoping by finding the possible SLF that minimizes the number of polarity discrepancies between the NatLog polarity labels and the labels inferred from the scoping and a manually curated list of negative polarity operators. The inference rules propagate the polarities so this is only performed on the input sentences (Section 3.4). During the inference process, this polarity marking is reserved as a fallback in cases where polarity propagation via inference rules fails or takes too long.

Possible SLFs are computed by generating every possible scope configuration while accounting for island constraints. We roughly model scope island constraints with the following rule: *Scoping operators cannot scope outside of ancestors that are ULF type-shifters*. This rule handles complex modifiers (which are shifted from predicates to modifiers) and reified clausal complements (e.g., *I believe that everyone thinks.*) and is implemented trivially with the ULF type system. This is an approximation of the full range of actual island constraints, which come in various classes and with nuances that are still under active investigation in linguistics research. Our rule tends to be stricter than actual scope island constraints leading to some losses in expressive capacity, such as exceptions to commonly accepted island constraints (Barker, 2021) and the *de dicto* / *de re* distinction for clausally-embedded indefinite quantifiers (Donnellan, 1966; Burge, 1977).[6] However, this is only a limitation of our implementation of scoping and polarity propagation. A more nuanced treatment of available scopings can be accommodated by the underlying theoretical inference framework (Kim et al., 2020).

3.4 Inference Rules

All of our inference rules fall under one of four categories.

1. *Monotonicity Substitution*

   This is the core monotonicity inference. Given the premise *Every A is a B*, *B* is substituted for *A* in positive polarity contexts and *A* is substituted for *B* in negative polarity contexts. In order to reduce the proof lengths, we suppress ULF macro expansion rules and extract monotonicity relations directly from macro instances.

2. *Conversion*

   Some A is a B $\Leftrightarrow$ *Some B is an A*

3. *Conservativity*

   δ *As are Bs* $\Leftrightarrow$ δ *As are As that/who are Bs*, where δ is a determiner. This is a category of inferences in the FraCaS dataset and a commonly used inference step for introducing and eliminating relative clauses in simple quantified expressions.

4. *Equivalences*

   This includes equivalent determiner substitutions (e.g., *Every dog is happy* $\Leftrightarrow$ *All dogs are happy*) and predicate synonym substitutions (e.g., *I saw the accident* $\Leftrightarrow$ *I witnessed the accident*).

We have 9 total inference rules when accounting for specializations for macros—though some of these inference rules themselves include several distinct transduction patterns to account for minor syntactic variations.

In order to identify whether a modification is intersective, we use the non-subsective adjective list by Nayak et al. (2014) expanded to words in the WordNet (Miller, 1995) synsets.

[5] This is available through the Natural Logic component of Stanford CoreNLP.

[6] For example, the referential reading of *someone* in the sentence *I know that someone lied* is not available if the indefinite quantifier is not allowed to take wide scope over the sentence.

Polarity Propagation For computational efficiency, each inference rule has a corresponding polarity propagation function. The polarity propagation function takes the premise ULF formulas, their polarity markings, and the conclusion and computes the polarity marking of the conclusion.

As a concrete example, consider the polarity propagation function for the UMI inference rule with the premises and conclusions based on FraCaS problem 24 described in section 2.1. The premises are steps 1 (*many delegates obtained interesting results from the survery*) and 2 (*all interesting results are results*) in the inference example and the conclusion is step 4 (*many delegates obtained results from the survey*). The polarity marking[7] for step 1 is step 3 of the proof and the polarity marking for step 2 is (all.d (interesting.a (plur result.n))▼ ((pres be.v) (= (k (plur result.n)▲)))).

The propagation function identifies that (plur result.n)▲ is the polarized version of the subexpression that substituted for (interesting.a (plur result.n)) in the step 1 premise. Thus, most polarity markings are transferred over from the step 1 polarity marking except the marking for (plur result.n) in the substituted subexpression. This leads to the following polarity marking of the conclusion.[8]

((many.d (plur delegate.n)■)
((past obtain.v) (k (plur result.n)▲)
(adv-a (from.p (the.d survey.n)))))

Most of these propagation functions can be implemented efficiently without accessing the corresponding SLFs because the inference context eliminates the possibility of polarity operators interacting outside of the localized expression substitution due to scope island constraints (Fodor and Sag, 1982; Park, 1995; Ruys and Winter, 2011; Barker, 2015). For example, the conversion rule substitutes two nominal predicates for each other in sentences with the main verb *be*, an indefinitely quantified subject, and a nominal subject complement. In this case, any quantifier embedded within either nominal predicate is constrained by the Complex NP Constraint (Ross, 1967).

A notable exception is the monotonicity substitution of determiners: the polarity propagation function must have access to the SLFs and cannot be implemented as efficiently because the new determiner may induce different polarities in its restrictor and body than the replaced determiner.[9] Properly computing the global polarity from this requires access to the quantifier scopes.

4 Experimental Setup

In our experiments we allow a maximum of 50 inference steps and use a leaf label F1 heuristic (LL-F1) which alternates with breadth-first search (BFS) every 5 inference steps. LL-F1 computes the F1 score between the leaf labels of the new formula and the goal formula, ignoring order, but preserving repetitions. This is turned into a cost ranging 0-to-1 by subtracting it from 1.

The FraCaS dataset is a set of entailment questions related to specific semantic phenomena that were curated by semanticists (Cooper et al., 1996). It contains 346 problems, of which 12 do not have well-defined answers. We focus on the most relevant section of the FraCaS dataset, section 1: Generalized Quantifiers (GQs). This is also the largest section, making up almost a quarter of the dataset. Due to the small size of the FraCaS dataset and the challenging phenomena it contains, prior research has trained and tested models on the same problems, focusing on the capacity of their systems to perform such inferences, rather than their competence in learning and generalizing to a larger scale. This aligns nicely with our goal to demonstrate the capacity to use ULFs as the basis for monotonic inferences, rather than present a system to compete with the state-of-the-art on entailment tasks.

5 Results

Our experiments show that our system is able to precisely cover a variety of semantic phenomena and constructions, but, as expected from a demonstration system, does not achieve the robustness of SOTA entailment systems.

Table 2 shows the confusion matrix of our system on the FraCaS dataset. Our system shows very high precision (it is never incorrect when it makes a definitive conclusion—not UNK) because of the soundness of our inference rules. While our system fails to correctly identify any contradictions, this is not an inherent limitation of the system. It was simply the case that parser errors led to the inability to match the inferred negated formula with the hypothesis in the 5 problems that have contradiction labels.

[7] Omitting irrelevant polarities.

[8] Again, omitting irrelevant polarities.

[9] For example, in positive contexts, *the* may be replaced with *a*, as in, *I saw the dog* ⇒ *I saw a dog*. *The* imposes a flat entailment context on its restrictor whereas *a* imposes a positive entailment context which warrants a fresh computation of the global polarity markings.

Section	Accuracy %													
	Single-premise				Multi-premise				Overall					
	BL	Ours	MN	LP	BL	Ours	MN	LP	BL	Ours	MN	LP	HU	HR
1 GQs	45	73	82	93	57	67	73	93	50	70	78	93	88	99

Table 1: FraCaS performance of our system (Ours) compared against a majority class (ENT) baseline (BL) and several notable RTE systems: MN (Mineshima et al., 2015), LP (Abzianidze, 2016), HU (Hu et al., 2019), and HR (Haruta et al., 2020). Hu et al. (2019) and Haruta et al. (2020) only report the overall accuracy of their systems.

Gold\Pred.	ENT	CON	UNK
ENT	**22**	0	15
CON	0	**0**	5
UNK	0	0	**32**

Table 2: Confusion matrix on the FraCaS dataset.

In Table 1, the accuracy of our system is compared to the majority class baseline and other natural logic systems that focus on monotonicity and FraCaS inferences. According to the table, a variety of methods prove effective at monotonic reasoning over a variety of linguistic phenomena. LP and HR perform notably well and both rely on CCG parses for obtaining the representation and theorem provers for managing inferences. Although our system falls short of the performance of SOTA systems on FraCaS, we still perform noticeably better than the majority class baseline. Investigating the error cases of our system makes clear that the shortfalls of our system are not inherent in the theoretical approach—rather they are due to syntactic and inference cases that were not addressed in this exploratory inference system.

The polarity propagation system used the fallback system (the polarity marking component of the Natlog system) in 42 out of the 3,109 (1.3%) total polarity propagation calls made in the GQs section of the FraCaS evaluation.

5.1 Qualitative Analysis

Figure 3 shows three distinct success and three distinct failure cases of our system. First looking at the successes, example 18 is a multi-premise entailment problem which requires conservativity inference and multiple UMIapplications in both positive and negative contexts. Example 59 has two distinct components—first the determiner *a few* must be generalized to *at least a few*, second *female* must be recognized as an intersective modifier and removed to generalize the nominal predicate in positive polarity context. Example 60 again has the intersective modifier *female*, but must not trigger an inference because of the negative polarity context.

Now taking a look at the failures, example 25 requires the recognition of *in major national newspapers* as an adjunct that may be dropped for a more general meaning. Our system parses the premise incorrectly—specifically "results published in major national newspapers" is parsed as a single kind-of-event[10] argument rather than an argument and an adjunct. This is can be addressed by an improvement of the ULF parser, e.g., an expansion of the verb subcategorization frames known by the ULF transduction rules. Example 48 requires the introduction of the phrase *a lot of* in negative polarity context, since it acts as a specializing modifier. Our system does not recognize *a lot of* as specializing modifier and this sort of multi-word idiosyncratic syntactic construction for a specializing modifier needs to be addressed specifically in the grammar. Finally, example 76 is a reversal of the intersective modifier *female* that was in the successful example 59. Because we use a forward inference framework, the proof-system does not have access to the modifier *female*. This could be handled by extraction of necessary intersective monotonicity rules from the hypothesis or more generally keeping a lexicon of intersective modifiers—though, the latter approach would be less efficient if implemented naively.

6 Conclusion

We have presented a simple implementation of forward monotonic inference starting with English sentences and using ULFs as the representational basis. Our system shows a high degree of precision on a variety of monotonicity phenomena,

[10] A kind-of-event is a type in the domain of discourse in EL semantics corresponding to generic events. For example, in the sentence "The news reporting on a missing kitten was unexpected", *unexpected* is a predicate over the kind-of-event "The news reporting on a missing kitten". This is distinct from similar EL types of events, which are particular instances, and propositions, which are statements that may be true or false.

Successes

ID	Correct inference
18	*Every European has the right to live in Europe*; *Every European is a person*; *Every* (*person who has the right to live in Europe*)$^{\blacktriangledown}$ *can travel freely within Europe* ⇒ *Every European can travel freely within Europe*
59	(*A few*)$^{\blacktriangle}$ (*female committee members*)$^{\blacktriangle}$ *are from Scandinavia* ⇒ *At least a few committee members are from Scandinavia*
60	*Few* (*female committee members*)$^{\blacktriangledown}$ *are from southern Europe* ⇏ *Few committee members are from southern Europe*

Failures

ID	Correct inference
25	*Several delegates* (*got the results published in major national newspapers*)$^{\blacktriangle}$ ⇒ *Several delegates got the results published*
48	*At most ten commissioners spend* (*time*)$^{\blacktriangledown}$ *at home* ⇒ *At most ten commissioners spend a lot of time at home*
76	*Few* (*committee members*)$^{\blacktriangledown}$ *are from southern Europe* ⇒ *Few female committee members are from southern Europe*

Figure 3: Several examples of inference successes and failures. The relevant polarity contexts for the final (or last two if not overlapping) inference step is marked with (...)$^{\blacktriangle}$ or (...)$^{\blacktriangledown}$ and the relevant spans are underlined in the premises and hypothesis. Our inference system predicated UNK for each of the failure examples.

empirically confirming the final class of inferences that Kim and Schubert (2019b) proposed would be supported by ULF alongside a suite of pragmatics-oriented inference capabilities of ULF described in Kim et al. (2019). The present effort is a feasibility demonstration and further engineering, expanding the coverage, is needed to create a system competitive with the state-of-the-art.

The specifics of our demonstration system and the results point to many possible avenues of improvement. Beyond direct improvements to the ULF parser, operator scoping, and inference rules to cover more constructions, the proof-search process can be expanded to explicitly include alternate parsing and scoping choices thereby enabling proper exploration of ambiguous constructions. For example, each particular English-to-ULF parse and each scoping choice leading to a distinct SLF can be formulated as an inference rule that can be explored. The inference rules can also be made more flexible by implementing the RI-1 and RI-2 rules that Kim et al. (2020) describe as a generalization of UMI. The direct access to syntactic structure from ULF leaves room for a much more sophisticated treatment of linguistic constraints (notably island constraints as discussed in section 3.4) and the logical type structure makes ULF theoretically capable of inferences from disjunctive conclusions; e.g., *Alice has a dog or a cat*, given that *Alice has a furry pet* and *Furry pets are either dogs or cats*. Finally, in the vein of merging ML/DL and symbolic approaches, ULF can be reliably translated back into English (Kim et al., 2019) so that ML/DL approaches that work over raw English text can be accessed and used in conjunction with the symbolic rules. In fact, the polarity marking component of our system (section 3.3) is precisely an example of such a bridging of methods.

7 Acknowledgments

This work was supported by NSF EAGER grant NSF IIS-1908595, DARPA CwC subcontract W911NF-15-1-0542, and a Sproull Graduate Fellowship from the University of Rochester. We are grateful to the anonymous reviewers for their helpful feedback.

References

Lasha Abzianidze. 2016. Natural solution to fracas entailment problems. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 64–74.

Gabor Angeli and Christopher D. Manning. 2014. NaturalLI: Natural logic inference for common sense reasoning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 534–545, Doha, Qatar. Association for Computational Linguistics.

Chris Barker. 2015. Scope. In Shalom Lappin and Chris Fox, editors, *Handbook of Contemporary Semantics*, 2 edition, chapter 2, pages 40–76. Wiley Blackwell.

Chris Barker. 2021. Rethinking scope islands. *Linguistic Inquiry*, pages 1–55.

Tyler Burge. 1977. Belief de re. *The Journal of Philosophy*, 74(6):338–362.

Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Johan Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, and Steve Pulman. 1996. Using the framework. Technical Report LRE 62-051 D-16, The FraCaS Consortium.

Keith S Donnellan. 1966. Reference and definite descriptions. *The philosophical review*, 75(3):281–304.

J. Fodor and I. Sag. 1982. Referential and quantificational indefinites. *Linguistics and Philosophy*, 5:355–398.

Jonathan Gordon and Lenhart Schubert. 2010. Quantificational sharpening of commonsense knowledge. In *Proceedings of the AAAI 2010 Fall Symposium on Commonsense Knowledge*.

Izumi Haruta, Koji Mineshima, and Daisuke Bekki. 2020. Combining event semantics and degree semantics for natural language inference. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1758–1764, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Hai Hu, Qi Chen, and Larry Moss. 2019. Natural language inference with monotonicity. In *Proceedings of the 13th International Conference on Computational Semantics - Short Papers*, pages 8–15, Gothenburg, Sweden. Association for Computational Linguistics.

Gene Kim, Benjamin Kane, Viet Duong, Muskaan Mendiratta, Graeme McGuire, Sophie Sackstein, Georgiy Platonov, and Lenhart Schubert. 2019. Generating discourse inferences from unscoped episodic logical formulas. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 56–65, Florence, Italy. Association for Computational Linguistics.

Gene Kim and Lenhart Schubert. 2019a. 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, Viet Duong, Xin Lu, and Lenhart Schubert. 2021. A transition-based parser for unscoped episodic logical forms.

Gene Louis Kim, Mandar Juvekar, and Lenhart Schubert. 2020. Monotonic inference for underspecified episodic logic. In *Proceedings of the Workshop Natural Logic Meets Machine Learning*.

Gene Louis Kim and Lenhart Schubert. 2019b. A type-coherent, expressive representation as an initial step to language understanding. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 13–30, Gothenburg, Sweden. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Bill MacCartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 521–528, Manchester, UK. Coling 2008 Organizing Committee.

George A. Miller. 1995. WordNet: A lexical database for english. *Communications of the ACM*, 38(11):39–41.

Koji Mineshima, Pascual Martínez-Gómez, Yusuke Miyao, and Daisuke Bekki. 2015. Higher-order logical inference with compositional semantics. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2055–2061.

Neha Nayak, Mark Kowarsky, Gabor Angeli, and Christopher D Manning. 2014. A dictionary of non-subsective adjectives. Technical Report CSTR 2014-04, Department of Computer Science, Stanford University.

Jong C. Park. 1995. Quantifier scope and constituency. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 205–212, Cambridge, Massachusetts, USA. 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.

John Robert Ross. 1967. *Constraints on Variables in Syntax*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Eddy G Ruys and Yoad Winter. 2011. Quantifier scope in formal linguistics. In *Handbook of philosophical logic*, pages 159–225. Springer.

Victor Sánchez Valencia. 1991. Categorial grammar and natural logic. ILTI Prepublication: Logic, Philosophy and Linguistics (LP) Series.

Victor Sánchez-Valencia. 1991. *Studies on Natural Logic and Categorial Grammar*. Ph.D. thesis, University of Amsterdam.

Lenhart Schubert. 2002. Can we derive general world knowledge from texts? In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pages 94–97, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lenhart Schubert. 2014. From treebank parses to episodic logic and commonsense inference. In *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, pages 55–60, Baltimore, MD. Association for Computational Linguistics.

Lenhart Schubert and Matthew Tong. 2003. Extracting and evaluating general world knowledge from the brown corpus. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning*, pages 7–13.

Lenhart K. Schubert. 2000. The situations we talk about. In Jack Minker, editor, *Logic-based Artificial Intelligence*, pages 407–439. Kluwer Academic Publishers, Norwell, MA, USA.

Ran Tian, Yusuke Miyao, and Takuya Matsuzaki. 2014. Logical inference on dependency-based compositional semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 79–89, Baltimore, Maryland. Association for Computational Linguistics.

Johan Van Benthem et al. 1986. *Essays in Logical Semantics*. Springer.