Combining Natural Logic and Shallow Reasoning for Question Answering

Gabor Angeli
Stanford University
Stanford, CA 94305

Neha Nayak
Stanford University
Stanford, CA 94305

Christopher D. Manning
Stanford University
Stanford, CA 94305

{angeli, nayakne, manning}@cs.stanford.edu

Abstract

Broad domain question answering is often difficult in the absence of structured knowledge bases, and can benefit from shallow lexical methods (broad coverage) and logical reasoning (high precision). We propose an approach for incorporating both of these signals in a unified framework based on natural logic. We extend the breadth of inferences afforded by natural logic to include relational entailment (e.g., *buy* $\rightarrow$ *own*) and meronymy (e.g., a person born in a city is born the city's country). Furthermore, we train an *evaluation function* – akin to gameplaying – to evaluate the expected truth of candidate premises on the fly. We evaluate our approach on answering multiple choice science questions, achieving the best published results on the dataset.

1 Introduction

Question answering is an important task in NLP, and becomes both more important and more difficult when the answers are not supported by hand-curated knowledge bases. In these cases, viewing question answering as textual entailment over a very large premise set can offer a means of generalizing reliably to open domain questions.

A natural approach to textual entailment is to treat it as a logical entailment problem. However, this high-precision approach is not feasible in cases where a formal proof is difficult or impossible. For example, consider the following hypothesis (H) and its supporting premise (P) for the question *Which part of a plant produces the seeds?*:

P: *Ovaries are the female part of the flower, which produces eggs that are needed for making seeds.*

H: *A flower produces the seeds.*

This requires a relatively large amount of inference: the most natural atomic fact in the sentence is that ovaries produce eggs. These inferences are feasible in a limited domain, but become difficult the more open-domain reasoning they require. In contrast, even a simple lexical overlap classifier could correctly predict the entailment (see, e.g., MacCartney (2009)). In fact, such a bag-of-words entailment model has been shown to be surprisingly effective on the Recognizing Textual Entailment (RTE) challenges (MacCartney, 2009). On the other hand, such methods are also notorious for ignoring even trivial cases of nonentailment that are easy for natural logic, e.g., recognizing negation in the example below:

P: *Eating candy for dinner is an example of a poor health habit.*

H: *Eating candy is an example of a good health habit.*

We present an approach to leverage the benefits of both methods. Natural logic – a proof theory over the syntax of natural language – offers a framework for logical inference which is already familiar to lexical methods. As an inference system searches for a valid premise, the candidates it explores can be evaluated on their similarity to a premise by a conventional lexical classifier.

We therefore extend a natural logic inference engine in two key ways: first, we handle relational entailment and meronymy, increasing the total number of inferences that can be made. We further implement an *evaluation function* which quickly provides an estimate for how likely a candidate premise is to be supported by the knowledge base, without running the full search. This can then more easily match a known premise despite still not matching exactly.

We present the following contributions: (1) we extend the classes of inferences NaturalLI can perform on real-world sentences by incorporating re-

lational entailment and meronymy, and by operating over dependency trees; (2) we augment NaturalLI with an evaluation function to provide an estimate of entailment for any query; and (3) we run our system over the Aristo science questions corpus, achieving the best published numbers.

2 Background

We briefly review natural logic and NaturalLI – the existing inference engine we use. Much of this paper will extend this system, with additional inferences (Section 3) and a soft lexical classifier (Section 4).

2.1 Natural Logic

Natural logic is a formal proof theory that aims to capture a subset of logical inferences by appealing directly to the structure of language, without needing either an abstract logical language (e.g., Markov Logic Networks; Richardson and Domingos (2006)) or denotations (e.g., semantic parsing; Liang and Potts (2015)). We use the logic introduced by the NatLog system (MacCartney and Manning, 2007; 2008; 2009), which was in turn based on earlier theoretical work on Monotonicity Calculus (van Benthem, 1986; Sánchez Valencia, 1991). We adopt the precise semantics of Icard and Moss (2014); we refer the reader to this paper for a more thorough introduction to the formalism.

At a high level, natural logic proofs operate by mutating spans of text to ensure that the mutated sentence follows from the original – each step is much like a syllogistic inference. Each mutation in the proof follows three steps:

1. An atomic lexical relation is induced by either inserting, deleting or mutating a span in the sentence. For example, in Figure 1, mutating *The* to *No* induces the $\curlywedge$ relation; mutating *cat* to *carnivore* induces the $\sqsubseteq$ relation. The relations $\equiv$ and $\sqsubseteq$ are variants of entailment; $\curlywedge$ and $\Downarrow$ are variants of negation.
2. This lexical relation between words is projected up to yield a relation between sentences, based on the *polarity* of the token. For instance, *The cat eats animals* $\sqsubseteq$ *some carnivores eat animals*. We explain this in more detail below.
3. These sentence level relations are *joined* together to produce a relation between a premise, and a hypothesis multiple mutations away. For example in Figure 1, if we join $\sqsubseteq$, $\equiv$, $\sqsubseteq$, and $\curlywedge$, we get negation ($\Downarrow$).

The notion of *projecting* a relation from a lexical item to a sentence is important to understand.[1] To illustrate, *cat* $\sqsubseteq$ *animal*, and *some cat meows* $\sqsubseteq$ *some animal meows* (recall, $\sqsubseteq$ denotes entailment), but *no cat barks* $\not\sqsubseteq$ *no animal barks*. Despite differing by the same lexical relation, the sentence-level relation is different in the two cases.

We appeal to two important concepts: *monotonicity* – a property of arguments to natural language operators; and *polarity* – a property of tokens. From the example above, *some* is monotone in its first argument (i.e., *cat* or *animal*), and *no* is antitone in its first argument. This means that the first argument to *some* is allowed to mutate up the specified hierarchy (e.g., hypernymy), whereas the first argument to *no* is allowed to mutate down.

Polarity is a property of tokens in a sentence determined by the operators acting on it. All lexical items have *upward* polarity by default; monotone operators – like *some*, *several*, or *a few* – preserve polarity. Antitone operators – like *no*, *not*, and *all* (in its first argument) – reverse polarity. For example, *mice* in ***no*** *cats eat mice* has downward polarity, whereas *mice* in ***no*** *cats do**n't*** *eat mice* has upward polarity (it is in the scope of two downward monotone operators).

As a final note, although we refer to the monotonicity calculus described above as *natural logic*, this formalism is only one of many possible natural logics. For example, McAllester and Givan (1992) introduce a syntax for first order logic which they call *Montagovian syntax*. This syntax has two key advantages: first, the "quantifier-free" version of the syntax (roughly equivalent to the monotonicity calculus we use) is computationally efficient while still handling limited quantification. Second, the syntax more closely mirrors that of natural language.

2.2 NaturalLI

We build our extensions within the framework of NaturalLI, introduced by Angeli and Manning (2014). NaturalLI casts inference as a search problem: given a hypothesis and an arbitrarily large corpus of text, it searches through the space of lex-

[1] For clarity we describe a simplified semantics here; NaturalLI implements the semantics described in Icard and Moss (2014).

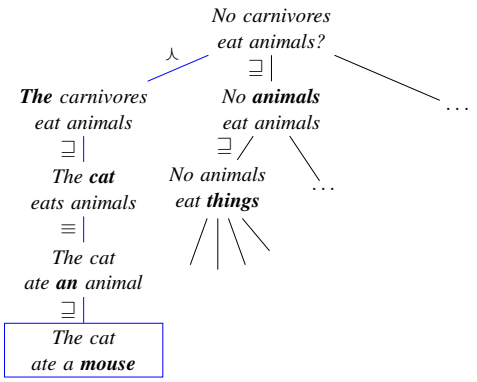


Figure 1: An illustration of NaturalLI searching for a candidate premise to support the hypothesis at the root of the tree. We are searching from a hypothesis *no carnivores eat animals*, and find a contradicting premise *the cat ate a mouse*. The edge labels denote Natural Logic inference steps.

ical mutations (e.g., *cat* → *carnivore*), with associated costs, until a premise is found.

An example search using NaturalLI is given in Figure 1. The relations along the edges denote relations between the associated sentences – i.e., the projected lexical relations from Section 2.2. Importantly, and in contrast with traditional entailment systems, NaturalLI searches over an arbitrarily large knowledge base of textual premises rather than a single premise/hypothesis pair.

3 Improving Inference in NaturalLI

We extend NaturalLI in a few theoretical ways to improve its coverage. We adapt the search algorithm to operate over dependency trees rather than the surface forms (Section 3.1). We enrich the class of inferences warranted by natural logic beyond hypernymy and operator rewording to also encompass meronymy and relational entailment (Section 3.2). Lastly, we handle token insertions during search more elegantly (Section 3.3).

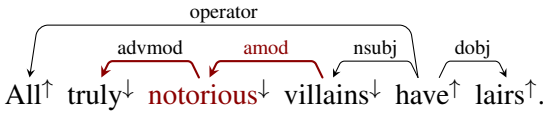
The general search algorithm in NaturalLI is parametrized as follows: First, an order is chosen to traverse the tokens in a sentence. For example, the original paper traverses tokens left-to-right. At each token, one of three operations can be performed: *deleting* a token (corresponding to inserting a word in the proof derivation), *mutating* a token, and *inserting* a token (corresponding to deleting a token in the proof derivation).

3.1 Natural logic over Dependency Trees

Operating over dependency trees rather than a token sequence requires reworking (1) the semantics of deleting a token during search, and (2) the order in which the sentence is traversed.

Recent work by Angeli et al. (2015) defined a mapping from Stanford Dependency relations to the associated lexical relation deleting the dependent subtree would induce. We adapt this mapping to yield the relation induced by *inserting* a given dependency edge, corresponding to our deletions in search; we also convert the mapping to use Universal Dependencies (de Marneffe et al., 2014). This now lends a natural deletion operation: at a given node, the subtree rooted at that node can be deleted to induce the associated natural logic relation.

For example, we can infer that *all truly notorious villains have lairs* from the premise *all villains have lairs* by observing that deleting an *amod* arc induces the relation ⊐, which in the downward polarity context of *villains*$^{\downarrow}$ projects to ⊑ (entailment):



An admittedly rare but interesting subtlety in the order we chose to traverse the tokens in the sentence is the effect mutating an operator has on the polarity of its arguments. For example, mutating *some* to *all* changes the polarity of its first argument. There are cases where we must mutate the argument to the operator before the operator itself, as well as cases where we must mutate the operator before its arguments. Consider, for instance:

P: *All felines have a tail*
H: *Some cats have a tail*

where we must first mutate *cat* to *feline*, versus:

P: *All cats have a tail*
H: *Some felines have a tail*

where we must first mutate *some* to *all*. Therefore, our traversal first visits each operator, then performs a breadth-first traversal of the tree, and then visits each operator a second time.

3.2 Meronymy and Relational Entailment

Although natural logic and the underlying monotonicity calculus has only been explored in the

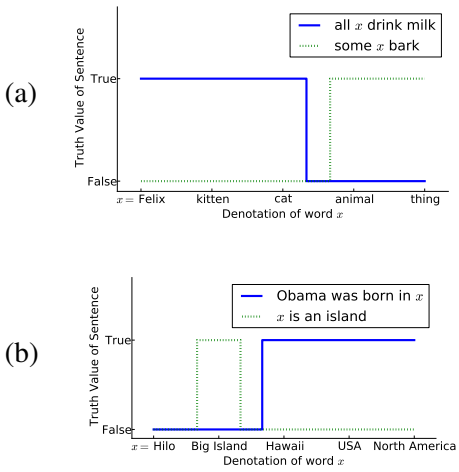


Figure 2: An illustration of monotonicity using different partial orders. (a) The monotonicity of *all* and *some* in their first arguments, over a domain of denotations. (b) An illustration of the *born in* monotone operator over the meronymy hierarchy, and the operator *is an island* as neither monotone or antitone.

context of hypernymy, the underlying framework can be applied to any partial order.

Natural language operators can be defined as a mapping from denotations of objects to truth values. The domain of word denotations is then ordered by the subset operator, corresponding to ordering by hypernymy over the words.[2] However, hypernymy is not the only useful partial ordering over denotations. We include two additional orderings as motivating examples: relational entailment and meronymy.

Relational Entailment For two verbs v_1 and v_2, we define $v_1 \leq v_2$ if the first verb entails the second. In many cases, a verb v_1 may entail a verb v_2 even if v_2 is not a hypernym of v_1. For example, to *sell* something (hopefully) entails *own*ing that thing. Apart from context-specific cases (e.g., *orbit* entails *launch* only for man-made objects), these hold largely independent of context. Note that the usual operators apply to relational entailments – if *all cactus owners live in Arizona* then *all cactus sellers live in Arizona*.

This information was incorporated using data from VERBOCEAN (**?**), adapting the confidence weights as transition costs. VERBOCEAN uses lexicosyntactic patterns to score pairs of verbs as candidate participants in a set of relations. We approximate the VERBOCEAN relations $\mathit{stronger\text{-}than}(v_1, v_2)$ (e.g., to kill is stronger than to wound) and $\mathit{happens\text{-}before}(v_2, v_1)$ (e.g., buying happens before owning) to indicate that v_1 entails v_2. These verb entailment transitions are incorporated using costs derived from the original weights from Chklovski and Pantel (2004).

Meronymy The most salient use-case for meronymy is with locations. For example, if Obama was born in Hawaii, then we know that Obama was born in America, because Hawaii is a meronym of (part of) America. Unlike relational entailment and hypernymy, meronymy is operated on by a distinct set of operators: if *Hawaii is an island*, we cannot necessarily entail that *America is an island*.

We semi-automatically collect a set of 81 operators (e.g., *born in*, *visited*) which then compose in the usual way with the conventional operators (e.g., *some*, *all*). These operators consist of dependency paths of length 2 that co-occurred in newswire text with a named entity of type *PERSON* and two different named entities of type *LOCATION*, such that one location was a meronym of the other. All other operators are considered non-monotone with respect to the meronym hierarchy.

Note that these are not the only two orders that can be incorporated into our framework; they just happen to be two which have lexical resources available and are likely to be useful in real-world entailment tasks.

3.3 Removing the Insertion Transition

Inserting words during search poses an inherent problem, as the space of possible words to insert at any position is on the order of the size of the vocabulary. In NaturalLI, this was solved by keeping a trie of possible insertions, and using that to prune this space. This is both computationally slow and adapts awkwardly to a search over dependency trees.

Therefore, this work instead opts to perform a bidirectional search: when constructing the knowledge base, we add not only the original sentence but also all entailments with subtrees deleted. For example, a premise of *some furry cats have tails* would yield two facts for the knowledge base: *some furry cats have tails* as well as *some cats have tails*. For this, we use the process de-

[2] Truth values are a trivial partial order corresponding to entailment: if $t_1 \leq t_2$ (i.e., $t_1 \sqsubseteq t_2$), and you know that t_1 is true, then t_2 must be true.

scribed in Angeli et al. (2015) to generate short entailed sentences from a long utterance using natural logic. This then leaves the reverse search to only deal with mutations and inference insertions, which are relatively easier.

The new challenge this introduces, of course, is the additional space required to store the new facts. To mitigate this, we hash every fact into a 64 bit integer, and store only the hashed value in the knowledge base. We construct this hash function such that it operates over a bag of edges in the dependency tree. This has two key properties: it allows us to be invariant to the word order of of the sentence, and more importantly it allows us to run our search directly over modifications to this hash function.

To elaborate, we notice that each of the two classes of operations our search is performing are done locally over a single dependency edge. When adding an edge, we can simply take the XOR of the hash saved in the parent state and the hash of the added edge. When mutating an edge, we XOR the hash of the parent state with the edge we are mutating, and again with the mutated edge. In this way, each search node need only carry an 8 byte hash, local information about the edge currently being considered (8 bytes), global information about the words deleted during search (5 bytes), a 3 byte backpointer to recover the inference path, and 8 bytes of operator metadata – 32 bytes in all, amounting to exactly half a cache line on our machines. This careful attention to data structures and memory layout turn out to have a large impact on runtime efficiency.

4 An Evaluation Function for NaturalLI

There are many cases – particularly as the length of the premise and the hypothesis grow – where despite our improvements NaturalLI will fail to find any supporting premises; for example:

P: *Food serves mainly for growth, energy and body repair, maintenance and protection.*

H: *Animals get energy for growth and repair from food.*

In addition to requiring reasoning with multiple implicit premises (a concomitant weak point of natural logic), a correct interpretation of the sentence requires fairly nontrivial nonlocal reasoning: *Food serves mainly for* $x \rightarrow$ *Animals get* x *from food*.

Nonetheless, there enough lexical clues in the sentence that even a simple entailment classifier would get the example correct. We build such a classifier and adapt it as an evaluation function inside NaturalLI in case no premises are found during search.

4.1 A Standalone Entailment Classifier

Our entailment classifier is designed to be as domain independent as possible; therefore we define only 5 unlexicalized real-valued features, with an optional sixth feature encoding the score output by the Solr information extraction system (in turn built upon Lucene). In fact, this classifier is a stronger baseline than it may seem: evaluating the system on RTE-3 (Giampiccolo et al., 2007) yielded **63.75**% accuracy – 2 points above the median submission.

All five of the core features are based on an alignment of keyphrases between the premise and the hypothesis. A keyphrase is defined as a span of text which is either (1) a possibly empty sequence of adjectives and adverbs followed by a sequence of nouns, and optionally followed by either *of* or the possessive marker (*'s*), and another noun (e.g., *sneaky kitten* or *pail of water*); (2) a possibly empty sequence of adverbs followed by a verb (e.g., *quietly pounce*); or (3) a gerund followed by a noun (e.g., *flowing water*). The verb *to be* is never a keyphrase. We make a distinction between a *keyphrase* and a *keyword* – the latter is a single noun, adjective, or verb.

We then align keyphrases in the premise and hypothesis by applying a series of sieves. First, all exact matches are aligned to each other. Then, prefix or suffix matches are aligned, then if either keyphrase contains the other they are aligned as well. Last, we align a keyphrase in the premise p_i to a keyphrase in the hypothesis h_k if there is an alignment between p_{i-1} and h_{k-1} and between p_{i+1} and h_{k+1}. This forces any keyphrase pair which is "sandwiched" between aligned pairs to be aligned as well. An example alignment is given in Figure 3.

Features are extracted for the number of alignments, the numbers of alignments which do and do not match perfectly, and the number of keyphrases in the premise and hypothesis which were not aligned. A feature for the Solr score of the premise given the hypothesis is optionally included; we revisit this issue in the evaluation.

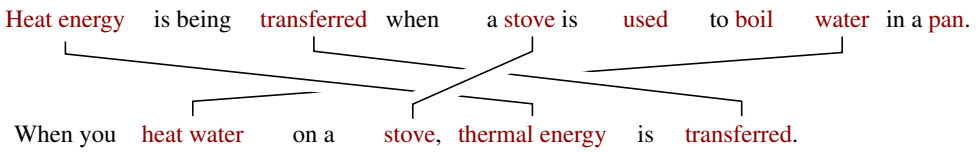


Figure 3: An illustration of an alignment between a premise and a hypothesis. Keyphrases can be multiple words (e.g., *heat energy*), and can be approximately matched (e.g., to *thermal energy*). In the premise, *used*, *boil* and *pan* are unaligned. Note that *heat water* is incorrectly tagged as a compound noun.

4.2 An Evaluation Function for Search

A version of the classifier constructed in Section 4.1, but over *keywords* rather than keyphrases can be incorporated directly into NaturalLI's search to give a score for each candidate premise visited. This can be thought of as analogous to the evaluation function in game-playing search – even though an agent cannot play a game of Chess to completion, at some depth it can apply an evaluation function to its leaf states.

Using keywords rather than keyphrases is in general a hindrance to the fuzzy alignments the system can produce. Importantly though, this allows the feature values to be computed incrementally as the search progresses, based on the score of the parent state and the mutation or deletion being performed. For instance, if we are deleting a word which was previously aligned perfectly to the premise, we would subtract the weight for a perfect and imperfect alignment, and add the weight for an unaligned premise keyphrase. This has the same effect as applying the trained classifier to the new state, and uses the same weights learned for this classifier, but requires substantially less computation.

In addition to finding entailments from candidate premises, our system also allows us to encode a notion of likely negation. We can consider the following two statements naïvely sharing every keyword. Each token marked with its polarity:

P: *some*$^{\uparrow}$ *cats*$^{\uparrow}$ *have*$^{\uparrow}$ *tails*$^{\uparrow}$
H: *no*$^{\uparrow}$ *cats*$^{\downarrow}$ *have*$^{\downarrow}$ *tails*$^{\downarrow}$

However, we note that all of the keyword pairs are in opposite polarity contexts. We can therefore define a pair of keywords as *matching* in NaturalLI if the following two conditions hold: (1) their lemmatized surface forms match exactly, and (2) they have the same polarity in the sentence. The second constraint encodes a good approximation for negation. To illustrate, consider the polarity signatures of common operators:

Operators	Subj. polarity	Obj. polarity
Some, few, etc.	$\uparrow$	$\uparrow$
All, every, etc.	$\downarrow$	$\uparrow$
Not all, etc.	$\uparrow$	$\downarrow$
No, not, etc.	$\downarrow$	$\downarrow$
Most, many, etc.	–	$\uparrow$

We note that most contradictory operators (e.g., *some*/*no*; *all*/*not all*) induce the exact opposite polarity on their arguments. Otherwise, pairs of operators which share half their signature are usually compatible with each other (e.g., *some* and *all*).

This suggests a criterion for likely negation: If the highest classifier score is produced by a contradictory candidate premise, we have reason to believe that we may have found a contradiction. To illustrate with our example, NaturalLI would mutate *no cats have tails* to *the cats have tails*, at which point it has found a contradictory candidate premise which has perfect overlap with the premise *some cats have tails*. Even had we not found the exact premise, this suggests that the hypothesis is likely false.

5 Related Work

This work is similar in many ways to work on recognizing textual entailment – e.g., Schoenmackers et al. (2010), Berant et al. (2011), Lewis and Steedman (2013). In the RTE task, a single premise and a single hypothesis are given as input, and a system must return a judgment of either *entailment* or *nonentailment* (in later years, *nonentailment* is further split into contradiction and independence). These approaches often rely on alignment features, similar to ours, but do not generally scale to large premise sets (i.e., a comprehensive knowledge base). The discourse commitments in Hickl and Bensley (2007) can be

thought of as similar to the additional entailed facts we add to the knowledge base (Section 3.3). In another line of work, Tian et al. (2014) approach the RTE problem by parsing into Dependency Compositional Semantics (DCS) (Liang et al., 2011). This work particularly relevant in that it also incorporates an evaluation function (using distributional similarity) to augment their theorem prover – although in their case, this requires a translation back and forth between DCS and language. Beltagy et al. (2016) takes a similar approach, but encoding distributional information directly in entailment rules in a Markov Logic Network (Richardson and Domingos, 2006).

Many systems make use of structured knowledge bases for question answering. Semantic parsing methods (Zettlemoyer and Collins, 2005; Liang et al., 2011) use knowledge bases like Freebase to find support for a complex question. Knowledge base completion (e.g., Chen et al. (2013), Bordes et al. (2011), or Riedel et al. (2013)) can be thought of as entailment, predicting novel knowledge base entries from the original database. In contrast, this work runs inference over arbitrary text without needing a structured knowledge base. Open IE (Wu and Weld, 2010; Mausam et al., 2012) QA approaches – e.g., Fader et al. (2014) are closer to operating over plain text, but still requires structured extractions.

Of course, this work is not alone in attempting to incorporate strict logical reasoning into question answering systems. The COGEX system (Moldovan et al., 2003) incorporates a theorem prover into a QA system, boosting overall performance on the TREC QA task. Similarly, Watson (Ferrucci et al., 2010) incorporates logical reasoning components alongside shallower methods. This work follows a similar vein, but both the theorem prover and lexical classifier operate over text, without requiring either the premises or axioms to be in logical forms.

On the Aristo corpus we evaluate on, Hixon et al. (2015) proposes a dialog system to augment a knowledge graph used for answering the questions. This is in a sense an oracle measure, where a human is consulted while answering the question; although, they show that their additional extractions help answer questions other than the one the dialog was collected for.

6 Evaluation

We evaluate our entailment system on the Regents Science Exam portion of the Aristo dataset (Clark et al., 2013; Clark, 2015). The dataset consists of a collection of multiple-choice science questions from the New York Regents 4th Grade Science Exams (NYSED, 2014). Each multiple choice option is translated to a candidate hypotheses. A large corpus is given as a knowledge base; the task is to find support in this knowledge base for the hypothesis.

Our system is in many ways well-suited to the dataset. Although certainly many of the facts require complex reasoning (see Section 6.4), the majority can be answered from a single premise. Unlike FraCaS or the RTE challenges, however, the task does not have explicit premises to run inference from, but rather must infer the truth of the hypothesis from a large collection of supporting text.

6.1 Data Processing

We make use of two collections of unlabeled corpora for our experiments. The first of these is the Barron's study guide (BARRON'S), consisting of 1200 sentences. This is the corpus used by Hixon et al. (2015) for their conversational dialog engine Knowbot, and therefore constitutes a more fair comparison against their results. However, we also make use of the full SCITEXT corpus (Clark et al., 2014). This corpus consists of 1 316 278 supporting sentences, including the Barron's study guide alongside simple Wikipedia, dictionaries, and a science textbook.

Since we lose all document context when searching over the corpus with NaturalLI, we first pre-process the corpus to resolve high-precision cases of pronominal coreference, via a set of very simple high-precision sieves. This finds the most recent candidate antecedent (NP or named entity) which, in order of preference, matches either the pronoun's animacy, gender, and number. Filtering to remove duplicate sentences and sentences containing non-ASCII characters yields a total of 822 748 facts in the corpus.

These sentences were then indexed using Solr. The set of promising premises for the soft alignment in Section 4, as well as the Solr score feature in the lexical classifier (Section 4.1), were obtained by querying Solr using the default similarity metric and scoring function. On the query

side, questions were converted to answers using the same methodology as Hixon et al. (2015). In cases where the question contained multiple sentences, only the last sentence was considered. As discussed in Section 6.4, we do not attempt reasoning over multiple sentences, and the last sentence is likely the most informative sentence in a longer passage.

6.2 Training an Entailment Classifier

To train a soft entailment classifier, we needed a set of positive and negative entailment instances. These were collected on Mechanical Turk. In particular, for each true hypothesis in the training set and for each sentence in the Barron's study guide, we found the top 8 results from Solr and considered these to be candidate entailments. These were then shown to Turkers, who decided whether the premise entailed the hypothesis, the hypothesis entailed the premise, both, or neither. Note that each pair was shown to only one Turker, lowering the cost of data collection, but consequently resulting in a somewhat noisy dataset. The data was augmented with additional negatives, collected by taking the top 10 Solr results for each false hypothesis in the training set. This yielded a total of 21 306 examples.

The scores returned from NaturalLI incorporate negation in two ways: if NaturalLI finds a contradictory premise, the score is set to zero. If NaturalLI finds a soft negation (see Section 4.2), and did not find an explicit supporting premise, the score is discounted by 0.75 – a value tuned on the training set. For all systems, any premise which did not contain the candidate answer to the multiple choice query was discounted by a value tuned on the training set.

6.3 Experimental Results

We present results on the Aristo dataset in Table 1, alongside prior work and strong baselines. In all cases, NaturalLI is run with the evaluation function enabled; the limited size of the text corpus and the complexity of the questions would cause the basic NaturalLI system to perform poorly. The test set for this corpus consists of only 68 examples, and therefore both perceived large differences in model scores and the apparent best system should be interpreted cautiously. NaturalLI consistently achieves the best training accuracy, and is more stable between configurations on the test set. For instance, it may be consistently discarding lexically similar but actually contradictory premises that often confuse some subset of the baselines.

System	Barron's		SCITEXT	
	Train	Test	Train	Test
KNOWBOT (held-out)	45	–	–	–
KNOWBOT (oracle)	57	–	–	–
Solr Only	49	42	62	58
Classifier	53	52	68	60
+ Solr	53	48	66	64
Evaluation Function	52	**54**	61	63
+ Solr	50	45	62	58
NaturalLI	52	51	65	61
+ Solr	**55**	49	73	61
+ Solr + Classifier	**55**	49	**74**	**67**

Table 1: Accuracy on the Aristo science questions dataset. All NaturalLI runs include the evaluation function. Results are reported using only the Barron's study guide or SCITEXT as the supporting KNOWBOT is the dialog system presented in Hixon et. al (2015). The held-out version uses additional facts from other question's dialogs; the oracle version made use of human input on the question it was answering. The test set did not exist at the time KNOWBOT was published.

KNOWBOT is the dialog system presented in Hixon et al. (2015). We report numbers for two variants of the system: *held-out* is the system's performance when it is not allowed to use the dialog collected from humans for the example it is answering; *oracle* is the full system. Note that the *oracle* variant is a human-in-the-loop system.

We additionally present three baselines. The first simply uses Solr's IR confidence to rank entailment (*Solr Only* in Table 1). The max IR score of any premise given a hypothesis is taken as the score for that hypothesis. Furthermore, we report results for the entailment classifier defined in Section 4.1 (*Classifier*), optionally including the Solr score as a feature. We also report performance of the evaluation function in NaturalLI applied directly to the premise and hypothesis, without any inference (*Evaluation Function*).

Last, we evaluate NaturalLI with the improvements presented in this paper (*NaturalLI* in Table 1). We additionally tune weights on our training set for a simple model combination with (1) Solr (with weight 6:1 for NaturalLI) and (2) the standalone classifier (with weight 24:1 for NaturalLI). Empirically, both parameters were ob-

System	Test Accuracy
Solr Only	46.8
Classifier	43.6
NaturalLI	46.4
+ Solr	**48.0**

Table 2: Results of our baselines and NaturalLI on a larger dataset of 250 examples. All NaturalLI runs include the evaluation function.

served to be fairly robust.

To demonstrate the system's robustness on a larger dataset, we additionally evaluate on a test set of 250 additional science exam questions, with an associated 500 example training set (and 249 example development set). These are substantially more difficult as they contain a far larger number of questions that require an understanding of a more complex process. Nonetheless, the trend illustrated in Table 1 holds for this larger set, given in Table 2. Note that with a web-scale corpus, accuracy of an IR-based system can be pushed up to 51.4%; a PMI-based solver, in turn, achieves an accuracy of 54.8% – admittedly higher than our best system (Clark et al., 2016).[3] An interesting avenue of future work would be to run NaturalLI over such a large web-scale corpus, and to incorporate PMI-based statistics into the evaluation function.

6.4 Discussion

We analyze some common types of errors made by the system on the training set. The most common error can be attributed to the question requiring complex reasoning about multiple premises. 29 of 108 questions in the training set (26%) contain multiple premises. Some of these cases can be recovered from (e.g., *This happens because the smooth road has less friction.*), while others are trivially out of scope for our method (e.g., *The volume of water most likely decreased.*). Although there is usually still some signal for which answer is most likely to be correct, these questions are fundamentally out-of-scope for the approach.

Another class of errors which deserves mention are cases where a system produces the same score for multiple answers. This occurs fairly frequently in the standalone classifier (7% of examples in training; 4% loss from random guesses), and especially often in NaturalLI (11%; 6% loss from random guesses). This offers some insight into why incorporating other models – even with low weight – can offer significant boosts in the performance of NaturalLI. Both this and the previous class could be further mitigated by having a notion of a *process*; e.g., as in Berant et al. (2014).

Other questions are simply not supported by any single sentence in the corpus. For example, *A human offspring can inherit blue eyes* has no support in the corpus that does not require significant multi-step inferences.

A remaining chunk of errors are simply classification errors. For example, *Water freezing is an example of a gas changing to a solid* is marked as the best hypothesis, supported incorrectly by *An ice cube is an example of matter that changes from a solid to a liquid to a gas*, which after mutating *water* to *ice cube* matches every keyword in the hypothesis.

7 Conclusion

We have presented two theoretical improvements to natural logic inference to make the formalism more robust for question answering. We augment the formalism with a theory for handling relational entailment and meronymy, and we incorporate a soft evaluation function for predicting likely entailments when formal support could not be found. These features allow us to perform large-scale broad domain question answering, achieving the best published numbers on the Aristo science exams corpus.

Acknowledgments

We thank the anonymous reviewers for their thoughtful comments. We gratefully acknowledge the support of the Allen Institute for Artificial Intelligence, and in particular Peter Clark for valuable discussions, as well as for access to the Aristo corpora and associated preprocessing. We would also like to acknowledge the support of the Defense Advanced Research Projects Agency (DARPA) Deep Exploration and Filtering of Text (DEFT) Program under Air Force Research Laboratory (AFRL) contract no. FA8750-13-2-0040. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of AI2, DARPA, AFRL, or the US government.

[3] Results from personal correspondence with the authors.

References

Gabor Angeli and Christopher D. Manning. 2014. NaturalLI: Natural logic inference for common sense reasoning. In *EMNLP*.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *ACL*.

Islam Beltagy, Stephen Roller, Pengxiang Cheng, Katrin Erk, and Raymond J. Mooney. 2016. Representing meaning with a combination of logical and distributional models. *The special issue of Computational Linguistics on Formal Distributional Semantics*.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *Proceedings of ACL*, Portland, OR.

Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Brad Huang, Christopher D Manning, Abby Vander Linden, Brittany Harding, and Peter Clark. 2014. Modeling biological processes for reading comprehension. In *Proc. EMNLP*.

Antoine Bordes, Jason Weston, Ronan Collobert, Yoshua Bengio, et al. 2011. Learning structured embeddings of knowledge bases. In *AAAI*.

Danqi Chen, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2013. Learning new facts from knowledge bases with neural tensor networks and semantic word vectors. *arXiv preprint arXiv:1301.3618*.

Timothy Chklovski and Patrick Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In *EMNLP*.

Peter Clark, Philip Harrison, and Niranjan Balasubramanian. 2013. A study of the knowledge base requirements for passing an elementary science test. In *AKBC*.

Peter Clark, Niranjan Balasubramanian, Sumithra Bhakthavatsalam, Kevin Humphreys, Jesse Kinkead, Ashish Sabharwal, and Oyvind Tafjord. 2014. Automatic construction of inference-supporting knowledge bases.

Peter Clark, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Turney, and Daniel Khashabi. 2016. Combining retrieval, statistics, and inference to answer elementary science questions.

Peter Clark. 2015. Elementary school science and math tests as a driver for ai: Take the aristo challenge! *AAAI*.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D Manning. 2014. Universal stanford dependencies: A cross-linguistic typology. In *Proceedings of LREC*.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *KDD*.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, Jmes Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. 2010. The AI behind Watson. *The AI Magazine*.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proc. of the ACL-PASCAL workshop on textual entailment and paraphrasing*. Association for Computational Linguistics.

Andrew Hickl and Jeremy Bensley. 2007. A discourse commitment-based framework for recognizing textual entailment. In *ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

Ben Hixon, Peter Clark, and Hannaneh Hajishirzi. 2015. Learning knowledge graphs for question answering through conversational dialog. *NAACL*.

Thomas Icard, III and Lawrence Moss. 2014. Recent progress on monotonicity. *Linguistic Issues in Language Technology*.

Mike Lewis and Mark Steedman. 2013. Combined distributional and logical semantics. *TACL*, 1:179–192.

Percy Liang and Christopher Potts. 2015. Corpus-based semantics and pragmatics. *Annual Review of Linguistics*, 1(1).

Percy Liang, Michael Jordan, and Dan Klein. 2011. Learning dependency-based compositional semantics. In *ACL*.

Bill MacCartney and Christopher D Manning. 2007. Natural logic for textual inference. In *ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

Bill MacCartney and Christopher D Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Coling*.

Bill MacCartney and Christopher D Manning. 2009. An extended model of natural logic. In *Proceedings of the eighth international conference on computational semantics*.

Bill MacCartney. 2009. *Natural Language Inference*. Ph.D. thesis, Stanford.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open language learning for information extraction. In *EMNLP*.

David A McAllester and Robert Givan. 1992. Natural language syntax and first-order inference. *Artificial Intelligence*, 56(1):1–20.

Dan Moldovan, Christine Clark, Sanda Harabagiu, and Steve Maiorano. 2003. COGEX: A logic prover for question answering. In *NAACL*.

NYSED. 2014. The grade 4 elementary-level science test. `http://www.nysedregents.org/Grade4/Science/home.html`.

Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine learning*, 62(1-2):107–136.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *NAACL-HLT*.

Víctor Manuel Sánchez Valencia. 1991. *Studies on natural logic and categorial grammar*. Ph.D. thesis, University of Amsterdam.

Stefan Schoenmackers, Oren Etzioni, Daniel S Weld, and Jesse Davis. 2010. Learning first-order horn clauses from web text. In *EMNLP*.

Ran Tian, Yusuke Miyao, and Takuya Matsuzaki. 2014. Logical inference on dependency-based compositional semantics. In *ACL*.

Johan van Benthem. 1986. *Essays in logical semantics*. Springer.

Fei Wu and Daniel S Weld. 2010. Open information extraction using wikipedia. In *ACL*. Association for Computational Linguistics.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI*. AUAI Press.