Combining Natural Logic and Shallow Reasoning for Question Answering

Gabor Angeli, Neha Nayak, Christopher Manning (Stanford University)

Presenter - Saket Karve

Penn Engineering

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, August 7-12, 2016.

Motivation

- Broad domain question answering is often difficult in absence of structured KB.
- Shallow lexical methods offer broad coverage
- Logical reasoning offer high precision
- Incorporating both signals in a unified framework will help cover a broader domain with high precision

The authors propose a QA framework which utilizes Logic to find a premise which entails a candidate hypothesis (possible answer) from a large corpus of text along with a lexical overlap classifier to offer broad domain coverage.

Content

- What is textual entailment?
- How does textual entailment relate to QA?
- Background
  - Natural Logic : Icard and Moss semantics
  - NaturalLI framework
- Improvements to NaturalLI
- Experiment and Results
- Summary
- Shortcomings and Future Work

Textual Entailment

"A premise **P** is said to entail a hypothesis **H** if a human reading **P** would infer that **H** is *most likely* true"

- Wikipedia

Textual entailment has a slightly relaxed definition as compared to logical entailment

Textual Entailment

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

H: A flower produces the seeds.

The premise above entails the hypothesis, but requires a large amount of inference,

What is needed to make seeds? **Eggs** → What produces eggs? **Ovaries** → Where are ovaries present? **Flower**

In contrast, a simple lexical overlap classifier could correctly predict the entailment.

Textual Entailment

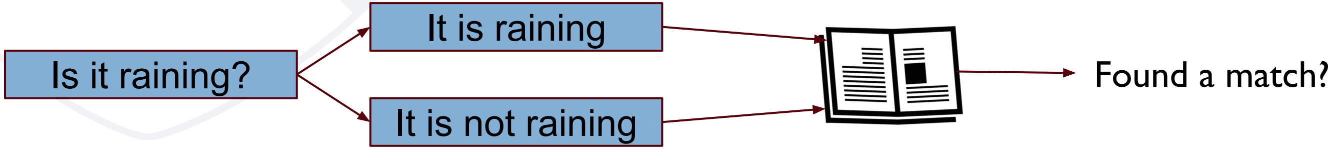
However, such bag-of-words lexical classifiers fail for trivial cases of non-entailment.

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

A technique which leverages benefits of both methods is needed.

Textual Entailment for QA

- QA is a useful application of textual entailment where we want the candidate answer (hypothesis) to entail from a supporting premise in the knowledge base.
- Standard textual entailment models work when a pair of premise and hypothesis is given.
- This means, a candidate hypothesis needs to be searched over all premises in the KB to find its truth value.



We need an elegant way to search over the space of premises given the hypothesis.

Background

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**."

- Uses logic introduced by the NatLog system - based on Monotonicity Calculus
- Adopts precise semantics of Icard and Moss

Natural Logic Proofs

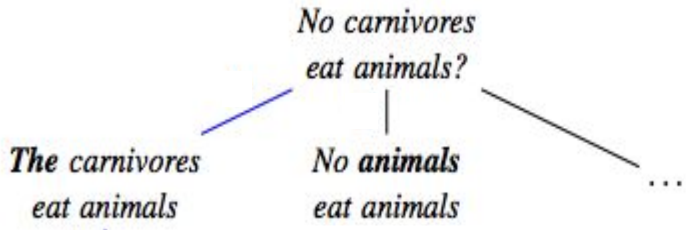
A natural logic proof typically operates as,

1. Mutate (change / insert / delete) a span of text
2. Define a *lexical relation* between the original and mutated span
3. Project the relation between words (or spans) to yield a relation between sentences
4. Repeat till mutation produces the premise (from hypothesis)
5. *Join all relations* to produce a relation between the premise and hypothesis

Natural Logic Proofs

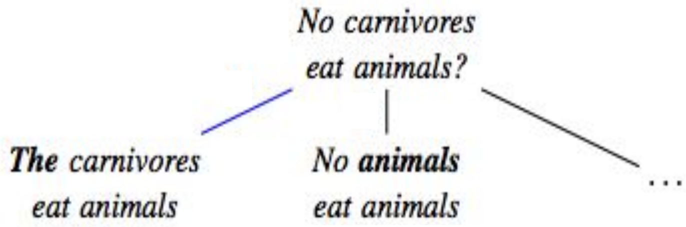
1. Mutate a span of text

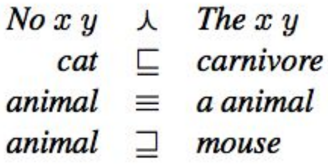
   - Starting from the hypothesis, we change *No* to *The*.



Natural Logic Proofs

1. Mutate a span of text
   - Starting from the hypothesis, we change *No* to *The.*
2. Identify the relation between mutated spans
   - Relations are defined by Icard and Moss semantics (next slide…)

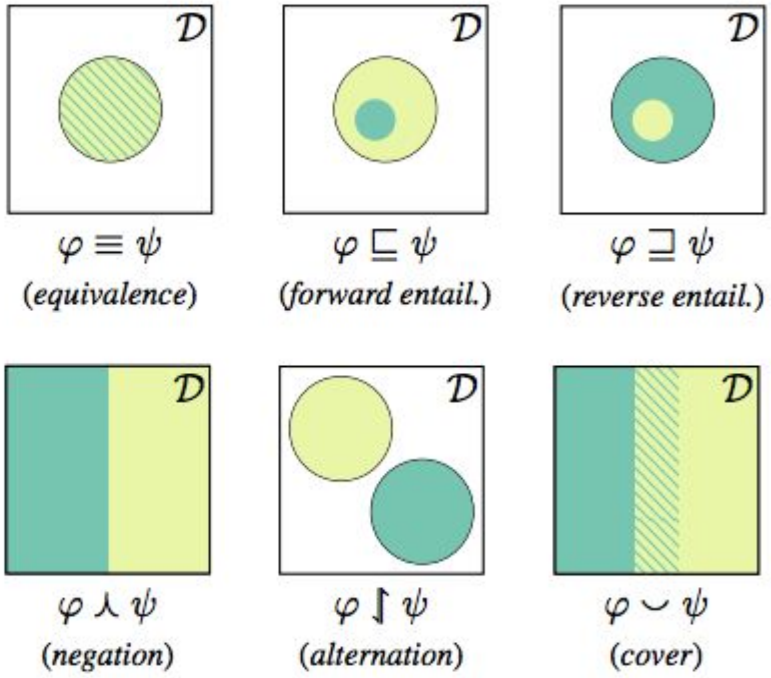




No x y	⅄	*The x y*
cat	⊑	*carnivore*
animal	≡	*a animal*
animal	⊒	*mouse*

Identifying relations

- McCartney and Manning define set-theoretic relations between denotations of any two objects.
- Denotations refers to the set of items in the universe which that lexical item refers. Example, denotation of *cat* denotes the set of all cats.
- The figure defines 6 relations and a 7th relation # corresponds to completely uninformative relation

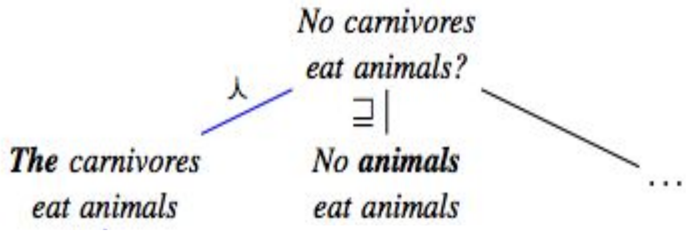


Source: NaturalLI: Natural Logic Inference for Common Sense Reasoning

Natural Logic Proofs

3. Project relation between spans to sentences

- Relations between tokens does not always correspond to the same relation between sentences.
- Example, *cat* ⊑ *animal*
  - *some cat meows* ⊑ *some animal meows.*
  - But, *no cat barks* ⋢ *no animal barks*
- We appeal to **monotonicity** and **polarity** to define these projections



No carnivores eat animals?

⅄

The carnivores eat animals

⊒

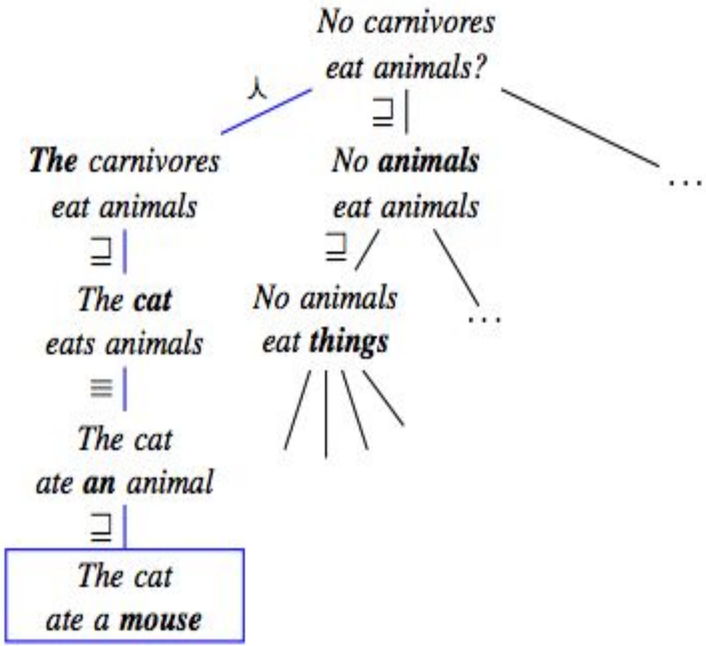
*No **animals** eat animals*

. . .

Natural Logic Proofs

4. Repeat till we get a matching premise
5. Join relations to produce a relation between the premise and the hypothesis

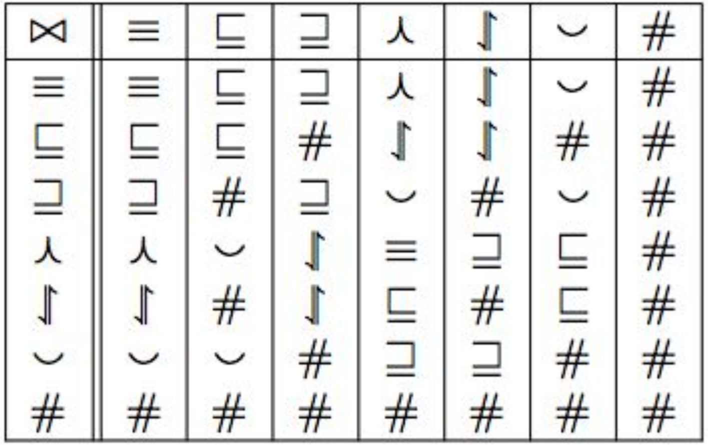
- Join relations starting from the premise, up the path till the root i.e. hypothesis



Joining relations

- Relations are joined top to bottom i.e. from premise to hypothesis in the search space
- The table defines how we join relations
- If we join the relations for the example in the previous slides

  ⊑ ⋈ ≡ ⋈ ⊑ ⋈ ⋏ ⇒ ⥮



⋈	≡	⊑	⊒	⋏	⥮	⌣	#
≡	≡	⊑	⊒	⋏	⥮	⌣	#
⊑	⊑	⊑	#	⥮	⥮	#	#
⊒	⊒	#	⊒	⌣	#	⌣	#
⋏	⋏	⌣	⥮	≡	⊒	⊑	#
⥮	⥮	#	⥮	⊑	#	⊑	#
⌣	⌣	⌣	#	⊒	⊒	#	#
#	#	#	#	#	#	#	#

Source: NaturalLI: Natural Logic Inference for Common Sense Reasoning

NaturalLI

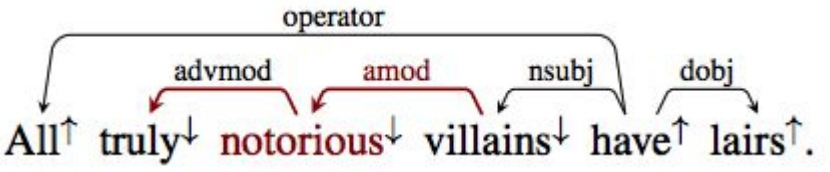
- The NaturalLI framework, thus, casts the inference problem as a search problem
  - Given a hypothesis and an arbitrarily large corpus of text, it searches through the space of lexical mutations, until a premise is found.

- Important to note is, this framework does not require a pair of premise and hypothesis to decide entailment. It rather searches through a large corpus to find a supporting premise.

Improvements to NaturalLI

Natural Logic over Dependency Trees

- In the extended NaturalLI framework, the authors adapt a search algorithm over dependency trees rather than lexical forms.
- Need to define mapping from dependency relations to the associated lexical relation
  - Adapted from Stanford Dependency relations (Angeli et. al. 2015)



  - Relation induced by deleting the *amod* dependency edge induces entailment

Support for more orderings

- Original framework only supports entailment corresponding to hypernymy over words.

- The extended framework adds support for two more ordering (inferences)

Support for more orderings

- **Relational Entailment:** For two verbs, *v1* and *v2*, we can say that *v1* may entail *v2* i.e. *v1* ≤ *v2* even if *v1* is not a hypernym of *v2*.

P: Dan *bought* a new house in Philadelphia

H: Dan *owns* a house in Philadelphia

The pairs of verbs which have the relations *stronger-than* and *happens-before* are approximated as entailment.

Support for more orderings

- **Meronymy:** Nouns having a *part-of* relation also corresponds to entailment.

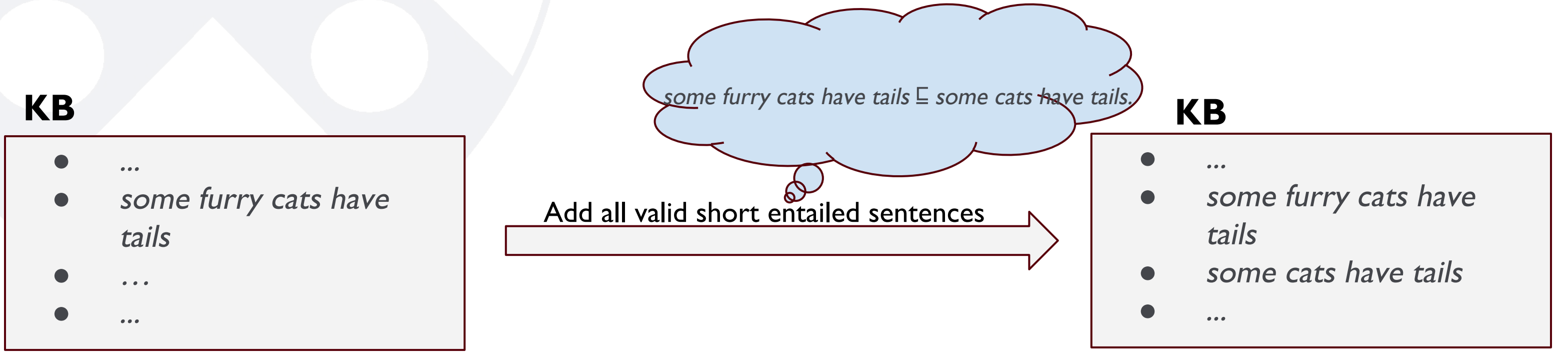
P: *Obama was born in Hawaii*
H: *Obama was born in America.*

Unlike the previous entailments, meronymy is operated on by a distinct set of operators. Because, *Hawaii is an island* $\not\sqsubseteq$ *America is an island.*

A set of 81 operators (e.g. *born-in, visited)* were chosen semi-automatically by the authors which are monotone with respect to meronymy.

Removing the insertion transition

- Inserting a token corresponds to a search over the vocabulary which is computationally slow and adapts awkwardly to dependency trees.
- The authors, thus eliminated the need of the insertion transition during search.



KB

- …
- *some furry cats have tails*
- …
- …

some furry cats have tails ⊑ some cats have tails.

Add all valid short entailed sentences

KB

- …
- *some furry cats have tails*
- *some cats have tails*
- …

- So, we would find a matching premise in the KB without needing to insert a token (*furry* in the example).

HASH every fact

- Adding more facts to the KB means more space required to store.
- The authors propose a **hash** of every fact to a 64 bit integer.
- The hash is constructed such that it operates over a bag of edges in the dependency tree and it allows us to run search directly over modifications to the hash.
- Any mutation, corresponds to an XOR of the hash saved in the parent state and the hash of the change.
- This makes the search very efficient and allows for large corpus of text.

Evaluation Classifier

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

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

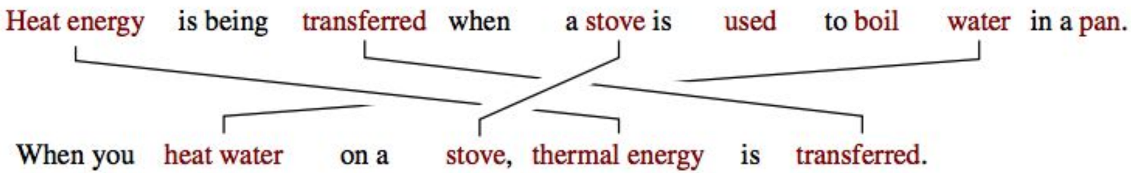
Above example requires reasoning with multiple implicit premises and a fairly non-trivial nonlocal reasoning.

However, there are lexical clues which a simple entailment classifier can get correct.

The classifier uses 5 core real valued features based on the alignment of *keyphrases* between the premise and the hypothesis.

Evaluation Classifier - Standalone

- The classifier uses 5 core real valued features based on the alignment of *keyphrases* between the premise and the hypothesis.
- First, keyphrases are identified and then are aligned between the premise and the hypothesis.



Heat energy is being transferred when a stove is used to boil water in a pan.

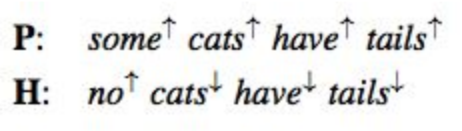
When you heat water on a stove, thermal energy is transferred.

- The 5 features are like number of completely aligned keyphrases, partially aligned keyphrases, etc.
- An optional 6th feature - the Solr score of the premise and hypothesis can also be included.

Evaluation Classifier in NaturalLI

The classifier used in NaturalLI aligns *keywords* instead of *keyphrases* because of the way mutations are defined in the system.

Some operators reverse the polarity of its arguments and can lead to misclassification.

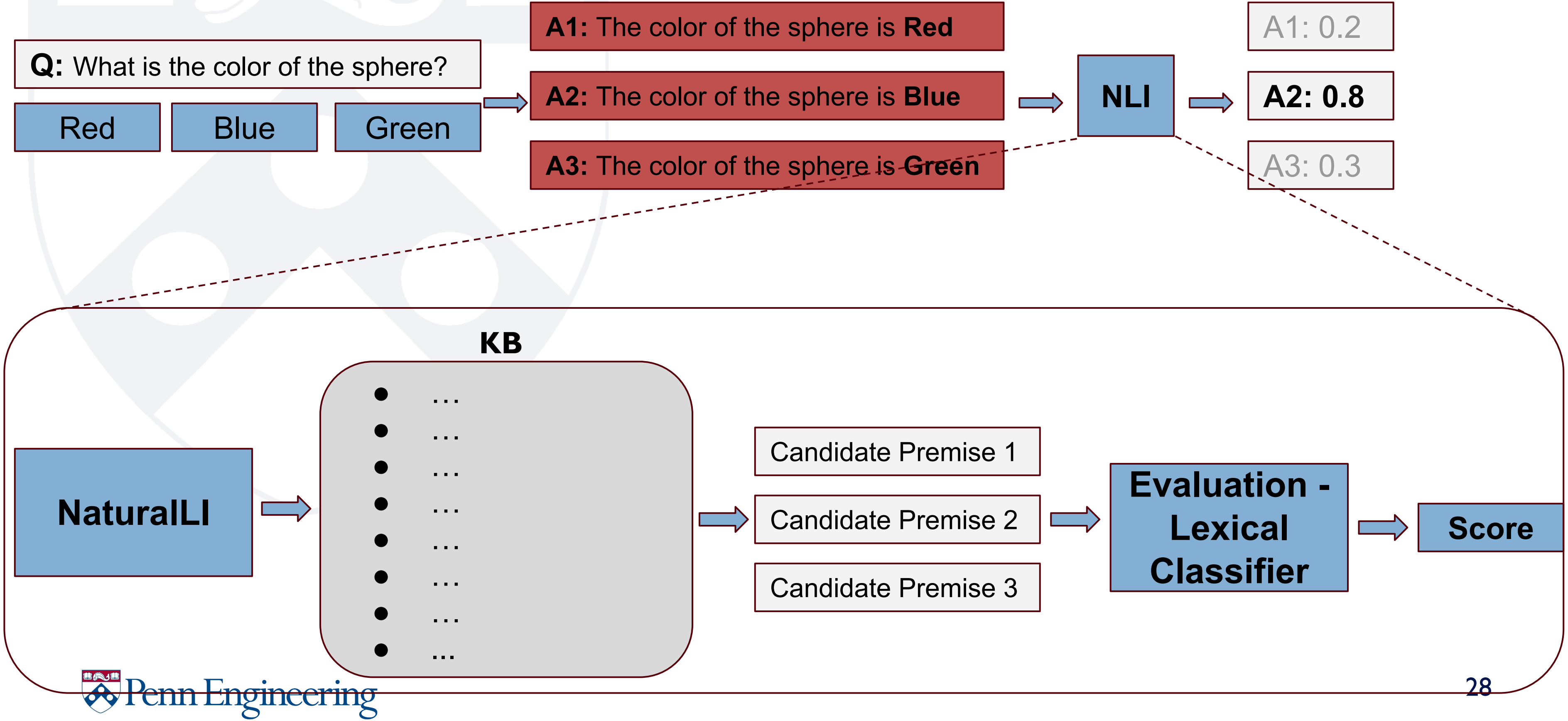


P: $\textit{some}^{\uparrow}$ $\textit{cats}^{\uparrow}$ $\textit{have}^{\uparrow}$ $\textit{tails}^{\uparrow}$

H: $\textit{no}^{\uparrow}$ $\textit{cats}^{\downarrow}$ $\textit{have}^{\downarrow}$ $\textit{tails}^{\downarrow}$

So, along with the *keywords* matching lexically, they should also have the same polarity to be considered ‘aligned’

Architecture Summary



Q: What is the color of the sphere?
Red
Blue
Green
A1: The color of the sphere is **Red**
A2: The color of the sphere is **Blue**
A3: The color of the sphere is **Green**
NLI
A1: 0.2
A2: 0.8
A3: 0.3
KB
● …
● …
● …
● …
● …
● …
● …
● …
NaturalLI
Candidate Premise 1
Candidate Premise 2
Candidate Premise 3
Evaluation - Lexical Classifier
Score

Experiments and Results

Data and Preprocessing

QA Dataset - Regents Science Exam from Aristo Dataset (Clark et. al. 2013)

Contains multiple-choice questions. Each choice is translated to a candidate hypothesis.

Two collections of unlabeled corpora used,

- Barron's study guide - 1200 sentences
- SCITEXT corpus - 1,316,278 sentences

Preprocessing and filtering these corpora yields a total of **822, 748** facts in the corpus.

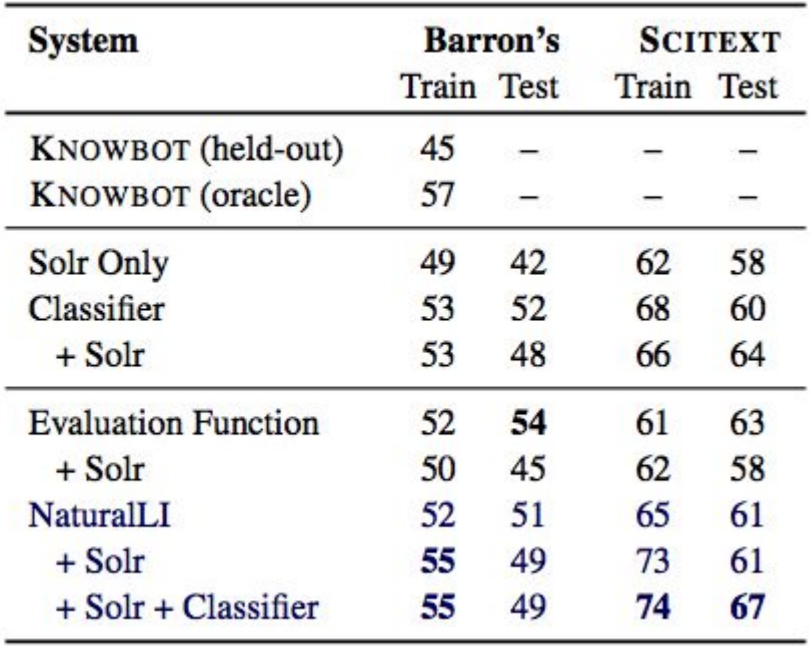
Training the evaluation classifier

- Positive and negative instances collected on Mechanical Turk

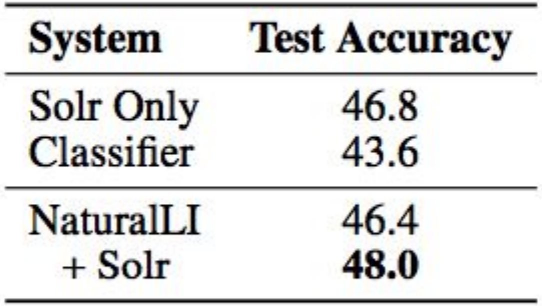
- **Positive instances -** For each true hypothesis, top 8 results from Solr considered candidate entailments and shown to turkers.
- **Negative instances -** For each false hypothesis, top 10 results from Solr taken.

- Total **21,306** instances used for training the soft entailment classifier.

Results



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**



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

Where does NaturalLI fail?

- Questions requiring complex reasoning about **multiple premises**. (26% of examples in test set)
- Cases where the system produces the **same score for multiple answers**. (7% examples)
- Questions having **no support** in the corpus.

Summary

- NaturalLI incorporates **logic** (Natural Logic) and **shallow lexical methods** (evaluation classifier) to answer broad-domain questions with high precision.

- Extension to the original framework - running inference over **dependency trees**, **pre-computing deletions** and **incorporating soft evaluation function** makes the system more robust for QA.

- New inferences like **meronymy** and **relational entailment** can be easily added allowing large scale broad domain QA.

Comments and Future Work

- Works only with questions requiring **single-hop reasoning**.
- Fails to answer questions where **multiple premises** are required.
- Only works when the question has a **limited number of known candidate answers**.
- The authors do not test the framework against an open domain dataset contradicting their motivation.

THANK YOU

Penn Engineering