Bill MacCartney

# The Stanford RTE System

- Introduction and Background Information
- The Stanford RTE System

# Introduction and Background Information

- Introduction to NLI and RTE
- Alignment for NLI

# NLI

Natural Language Inference (NLI) is the problem of determining whether a natural language **hypothesis *h*** can reasonably be inferred (entailed) from a natural language **premise *p***.

The emphasis of NLI challenges is on: *in*formal reasoning, lexical semantic knowledge, and variability of linguistic expression.

# Inference and Reasoning

*p*: Several airlines polled saw costs grow more than expected, even after adjusting for inflation.

*h*: Some of the companies in the poll reported cost increases.

Though *h* is not a strict logical consequence of *p*, this pair is still a valid inference.

# RTE

RTE: Recognising Textual Entailment.

RTE Challenge is a recent and well-known formulation of the NLI task.

# Why Is NLI Needed?

A fundamental phenomenon of natural language is the variability of semantic expression. Many NLP applications need a model for this variability phenomenon in order to recognise that a particular target meaning can be inferred from different text variants.

The RTE Challenge is an attempt to promote an abstract generic task that captures major semantic inference needs.

# Application

- Question answering
- Semantic search
- Automatic summarisation
- Evaluation of machine translation systems

# Previous Approaches to NLI Task

- Shallow approaches

  Bag-of-words model: matching each word in *h* to the word in *p* with which it is most similar. If most words in *h* can be matched quite well to a word in *p*, this model would likely predict the inference is valid.

- Deep approaches

  Relying on full semantic interpretation: translate *p* and *h* into some formal meaning representation (like FOL), then apply automated reasoning tools to determine inferential validity.

# Drawbacks of these Approaches

Bag-of-words model: could be fooled easily:

- *p:* The main race track in Qatar is located in Shahaniya, on the Dukhan Road.
- *h:* Qatar is located in Shahaniya.

Deep approach: fails badly on open-domain NLI evaluations such as Recognising Textual Entailment.

# Solutions?

- The natural logic approach
  *"natural logic": a logic whose vehicle of inference is natural language*
- Using a multi-stage architecture

# Alignment: Goal

Establishing links between corresponding entities and predicates in the premise $p$ and the hypothesis $h$ is needed.

# A Phrase-based Alignment Representation

We represent an alignment between p and h as a set of phrase edits. Here are the types from an aligner that may apply:

- an EQ edit connects a phrase in *p* with an equal (by word lemmas) phrase in *h*
- a SUB edit connects a phrase in *p* with an unequal phrase in *h*
- a DEL edit covers an unaligned phrase in *p*
- an INS edit covers an unaligned phrase in *h*

# A Phrase-based Alignment Representation

After alignment, we could get

$$\begin{aligned}
&\text{DEL}(In_1)\\
&\text{DEL}(most_2)\\
&\text{DEL}(Pacific_3)\\
&\text{DEL}(countries_4)\\
&\text{DEL}(there_5)\\
&\text{EQ}(are_6,\ are_2)\\
&\text{SUB}(very_7\ few_8,\ poorly_3\ represented_4)\\
&\text{EQ}(women_9,\ Women_1)\\
&\text{EQ}(in_{10},\ in_5)\\
&\text{EQ}(parliament_{11},\ parliament_6)\\
&\text{EQ}(.12,\ .7)
\end{aligned}$$

from the pair

*p*: In most Pacific countries there are very few women in parliament.

*h*: Women are poorly represented in parliament.

# A Feature-based Scoring Function

To score alignments, we use a simple feature-based linear scoring function.

If $E$ is a set of edits constituting an alignment, $\mathbf{\Phi}$ is a vector of feature functions, and $\mathbf{w}$ is the feature weights, the score $s$ is given by:

$$s(E) = \sum_{e \in E} s(e) = \sum_{e \in E} \mathbf{w} \cdot \mathbf{\Phi}(e)$$

# Costs and Predicting Entailment

The score is transformed to costs, which has a value of zero when aligning equal words, and a very high value for very different words.

Costs could be used for predicting entailment, directly applied to a formula or as the input to a machine learning classifier.

# Using Alignment To Predict RTE Answers

For a given RTE problem, we predict YES (valid) if its alignment score exceeds a threshold $\tau$, and NO otherwise.

# The Stanford RTE System

- General Ideas
- System
- Feature representation
- Evaluation and Conclusion

# How the System Works

1. finding the best alignment between premise and hypothesis
2. extracting high-level semantic features of the entailment problem
3. inputting these features to a statistical classifier to make an entailment decision

So, this is a multi-stage architecture.

# A Middle Ground Between Two Approaches

Approaches to NLI based on lexical similarity and those based on full semantic interpretation have their own strengths and drawbacks:

- based on lexical similarity: robust but imprecise
- based on semantic interpretation: precise but brittle

The Stanford RTE system is an attempt to find a middle ground between them.

# Another Reason for Being Multi-stage

There are three fundamental semantic limitations:

- assumption of monotonicity,
  *if a good match is found with a part of the premise, other material in the premise could/couldn't affect the validity*
- assumption of locality,
  *locality is needed to allow practical search, but many entailment decisions rely on global features of the alignment*
- confounding of alignment and evaluation of entailment.

All three problems can be resolved in a multi-stage architecture, where the alignment phase is followed by a separate phase of entailment determination.

# How the Mechanism Works

1. The approach to alignment emphasizes structural correspondence, and downplays issues like polarity and quantity *(which can be left to a subsequent entailment decision)*.
2. Given a good alignment, the determination of entailment reduces to a simple classification decision.
3. Because we already have a complete alignment, the classifier's decision can be conditioned on arbitrary global features of the aligned graphs, and it can detect inversions of monotonicity.

# Stages of the System

- linguistic analysis
- alignment
- entailment determination

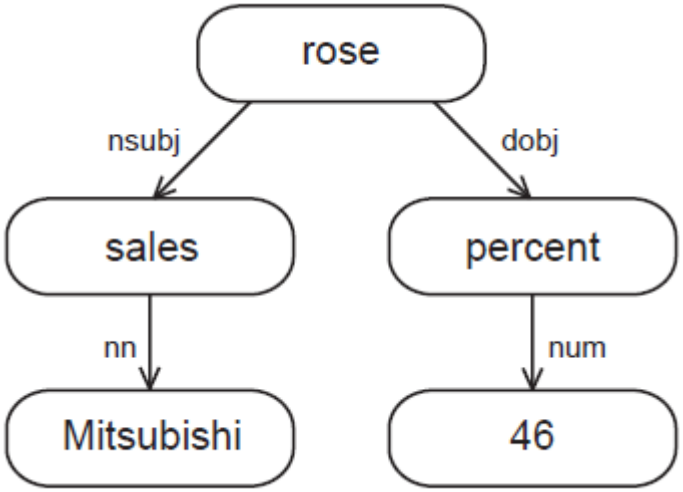
# Linguistic Analysis: Goal

To compute linguistic representation of p and h that contain as much information as possible about their semantic content.

# Linguistic Analysis: Tool

Typed dependency graphs: node for each word, labelled edges representing grammatical relations.

*Mitsubishi sales rose 46 percent.*



# Linguistic Analysis: General Approach

1. to parse the input sentences,
2. to convert each of the resulting phrase structure trees to a typed dependency graph.

# Alignment: Goal and Content

Goal: to find a good partial alignment between the typed dependency graphs representing *p* and *h*.

An alignment consists of a mapping from each node (word) in the *h* graph to a single node in the *p* graph, or to null.

# Alignment: An Example

*p*: Mitsubishi Motors Corp.'s new vehicle sales in the US fell 46 percent in June.

*h*: Mitsubishi sales rose 46 percent.

| | | |
|---|---|---|
| *rose* | → | *fell* |
| *sales* | → | *sales* |
| *Mitsubishi* | → | *Mitsubishi_Motors_Corp.* |
| *percent* | → | *percent* |
| *46* | → | *46* |

SCORE: −0.8962

# Alignment: Complexity and Scoring

There are $O((m+1)^n)$ possible alignments for a $p$ graph with m nodes and an $h$ graph with n nodes.

Scored by a locally decomposable scoring function: the score of an alignment is the sum of the local node and edge alignment scores.
*Scoring measure used here favours alignments which align semantically similar sub-graphs, irrespective of polarity.*

# Entailment Determination

Conditioned on the typed dependency graphs of *p* and *h*, as well as the best alignment between them, a decision about whether or not *h* is entailed by *p* is made.

# Entailment Determination: Method

Because we have a data set of examples that are labelled for entailment, we can use techniques from supervised machine learning to learn a classifier. One can apply any statistical learning algorithm *(such as support vector machines, logistic regression, or naive Bayes)* to this classification task.

# Advantage of Statistical Classifiers

They can be configured to output a probability over possible answers rather than just the most likely answer, which makes computing a confidence weighted score easier.

# Feature Representation

In the entailment determination phase, the entailment problem is reduced to a representation as a vector of features, over which the statistical classifier described above operates. The features must capture factors distinguish good/bad entailments from good/bad alignments.

# Important Features

- Polarity features: capture the presence of linguistic markers of negative polarity contexts *(such as not, no, few, without, etc.)* and superlatives
- Adjunct features: Indicating the dropping/adding of syntactic adjuncts when moving from *p* to *h*.
  Generally, dropping an adjunct preserves truth *(Dogs barked loudly ⊨ Dogs barked)*, while adding an adjunct does not *(Dogs barked ⊭ Dogs barked today)*. But in negative-polarity contexts, adding but not dropping adjuncts is safe.

# Important Features

- Antonymy features: check whether any aligned pairs of $\langle p, h \rangle$ words appear to be antonymous by consulting a pre-computed list.
- Modality features: capture simple patterns of modal reasoning (like must or maybe), which illustrates that possibility does not entail actuality.
  (NOT POSSIBLE $\vDash$ NOT ACTUAL)? $\Rightarrow$ YES
  (POSSIBLE $\vDash$ NECESSARY)? $\Rightarrow$ WEAK NO

# Important Features

- Factivity features. The context in which a verb phrase is embedded may carry semantic presuppositions affecting entailments.
  *The gangster tried to escape ⊭ The gangster escaped.*
  *The gangster managed to escape ⊨ The gangster escaped.*
- Quantifier features: to capture entailment relations among simple sentences involving quantification.
  *Every company must report ⊨ A company/The company/IBM must report.*

# Important Features

- Number, date, and time features
- Alignment features: three real-valued features intended to represent the quality of the alignment: SCORE is the raw score returned from the alignment phase, while GOOD SCORE and BAD SCORE try to capture whether the alignment score is “good” or “bad”.

# Evaluation

Results presented based on the PASCAL RTE1 Challenge. The Challenge recommended two evaluation metrics: raw accuracy and confidence weighted score (CWS).

| Algorithm | RTE1 Dev Set | | RTE1 Test Set | |
|---|---|---|---|---|
| | Acc % | CWS % | Acc % | CWS % |
| Random | 50.0 | 50.0 | 50.0 | 50.0 |
| Jijkoun et al. 05 | 61.0 | 64.9 | 55.3 | 55.9 |
| Raina et al. 05 | 57.8 | 66.1 | 55.5 | 63.8 |
| Haghighi et al. 05 | — | — | 56.8 | 61.4 |
| Bos & Markert 05 | — | — | 57.7 | 63.2 |
| Stanford RTE, alignment only | 58.7 | 59.1 | 54.5 | 59.7 |
| Stanford RTE, hand-tuned | 60.3 | 65.3 | **59.1** | **65.0** |
| Stanford RTE, learning | 61.2 | 74.4 | **59.1** | **63.9** |

# Evaluation

The real-valued alignment features GOOD SCORE and BAD SCORE proved to be highly informative. However, it is suggested that the SCORE feature carries little additional information for weighting.

# Thank You