Tree Edit Models for Recognizing Textual Entailments, Paraphrases, and Answers to Questions

Michael Heilman and Noah A. Smith
Summary

- Simple transformational approach for modeling sentence pair relations.
- Experiments for multiple problems:
  - Recognizing textual entailment
  - Paraphrase identification
  - Answer selection for question answering
- Competitive but not standout performance.
Intuition

Tree edits are syntactic transformations that can modify semantic properties in various ways.

We represent sentence pairs as sequences of edits that convert one tree into the other.
Outline

- Introduction
- **Connections to Prior Work**
- Finding & Classifying Edit Sequences
- Experiments
Prior Work on Sentence Pairs

- Numerous approaches for sentence pair relations, some task-specific.
- Considerable work involving tree and phrase alignments.

- Less work on transformational or tree edit approaches.

Das & Smith, 09; MacCartney et al., 08; Zanzotto, 09; Chang et al., NAACL-10; inter alia

Harmeling, 07; Bar Haim et al., 07
Prior Work on Tree Edit Distance

1. Local edits without reordering.
   • insert, relabel, delete

2. No learning of associations between labels and features of edit sequences.

Chawathe et al., 97; Punyakanok et al., 04; Wan et al., 06; Bernard et al., 08; inter alia
Our Method

1. Includes edits for reordering children and moving subtrees.
2. Learns associations between edit sequences and features of labeled data.
3. Does not require:
   - WordNet
   - Distributional Similarity
   - NER
   - Heavy task-specific tuning
   - Coreference resolution
   - Etc.

Possible future work
Outline

- Introduction
- Connections to Prior Work
- **Finding & Classifying Edit Sequences**
- Experiments
With a wry smile, Mr. Bush replied, “You're looking pretty young these days.”

Bush shot back: “You're looking pretty young these days.”

DELETE (a)
DELETE (wry)
DELETE (smile)
DELETE (with)
RELABEL (replied, shot)
DELETE (Mr.)
INSERT (back, shot)
RELABEL (comma, :)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td># edits</td>
<td>8</td>
</tr>
<tr>
<td># unedited nodes</td>
<td>11</td>
</tr>
<tr>
<td># DELETE</td>
<td>5</td>
</tr>
<tr>
<td># INSERT</td>
<td>1</td>
</tr>
<tr>
<td># delete subject</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Types of Tree Edits

- Inserting, Deleting, Relabeling Nodes
  - INSERT-CHILD
  - INSERT-PARENT
  - DELETE-LEAF
  - DELETE-AND-MERGE
  - RELABEL-NODE
  - RELABEL-EDGE

- Reordering Children
  - MOVE-SIBLING

- Moving Subtrees
  - MOVE-SUBTREE
  - NEW-ROOT
Complexity

- Tree edit distance with insert, relabel, delete edits:
  \[ O(n^3 \log n) \]

- With reordering and moving subtrees:
  Polynominal runtime algorithms not available

\textit{Klein, 98}
Greedy Best-First Search

- We choose the next tree according to the heuristic function only.
  - We ignore path cost.

Initial Tree (e.g., premise)

Target Tree (e.g., hypothesis)

Pearl, 84
Tree Kernel Search Heuristic

- Heuristic compares current tree to target tree 🌳.
- Tree kernel: similarity measure between trees based on similarities of all their subtrees.
  - Efficient dynamic programming solution.

D. Haussler, 99; Collins & Duffy, 01; Zanzotto & Moschitti, 06; Zelenko et al., 06
Tree Kernel Search Heuristic

- In general, larger trees will have larger kernel values.
- So we “normalize” to $[0, 1]$:

$$H(\text{tree}) = 1 - \frac{K(\text{tree}, \text{tree})}{\sqrt{K(\text{tree}, \text{tree}) \times K(\text{tree}, \text{tree})}}$$

heuristic function

tree kernel function
Finding Edit Sequences

- Operations are very expressive.
  - Search rarely fails (< 0.5%).

- Resulting sequences:
  - Succinct and plausible upon inspection
  - Internally consistent representation
  - Lead to good performance
Example Edit Sequence

Premise

Pierce built the home for his daughter off Rossville Blvd, as he lives nearby.

Hypothesis

Pierce lives near Rossville Blvd.
Example Edit Sequence

Pierce built the home for his daughter off Rossville Blvd, as he lives nearby.

Pierce built the home for his daughter off Rossville Blvd, as he lives near.

Pierce built the home for his daughter off, as he lives near Rossville Blvd.

built the home for his daughter off, as Pierce he lives near Rossville Blvd.

Multiple RELABEL-EDGE, DELETE-LEAF, DELETE-AND-MERGE edits

Pierce lives near Rossville Blvd.
Classifying by Edit Sequences

- Logistic Regression with 33 features.
  - total number of edits
  - number of X edits
  - number of edits removing a subject
  - number of unedited nodes
  - etc.

- We learn separate parameters for each task from labeled sentence pairs.
Outline

- Introduction
- Connections to Prior Work
- Finding & Classifying Edit Sequences
- Experiments
Recognizing Textual Entailment

Challenge: Decide whether a hypothesis follows from a premise.

- **Testing:** RTE-3 test data.
- **Training:** RTE-3 dev. data and data from previous RTE tasks.

Giampiccolo et al., 07
RTE-3 Results

**de Marneffe et al. 06**
- Syntactic alignment + classification

**Tree edit model**

**MacCartney & Manning 08: Hybrid**
- de Marneffe et al. 06
- + Natural Logic technique
Paraphrase Identification

Challenge: Decide whether 2 sentences are paraphrases of each other.

- Paraphrase $\approx$ bidirectional entailment.
- Microsoft Research Paraphrase Corpus
  - Standard training and testing splits

*Dolan et al., 04*
Paraphrase Identification Results

**Tree edit model**

**Wan et al. 06**
SVM with syntactic dependency overlap, BLEU scores, tree edit distance, etc.

**Das & Smith 09**
Quasi-synchronous Grammar to model syntactic alignments + n-gram overlap
Answer Selection for QA

Challenge: rank sentences by correctness as answers to a given question.

- We find edit sequences from answers to questions.
- We rank by the estimated probabilities of correctness.
Answer Selection Data

- Q&A pairs from TREC-8 through TREC-13.
- Training, Dev., Testing data sets: about 100 questions and 500-1500 answers each
Punyakanok et al. 04
Tree edit distance

Wang et al. 07
Quasi-synchronous Grammar to model syntactic alignments

Wang et al. 07 + WN
plus lexical semantics from WordNet

Tree edit model

Answer Selection Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Ranking Quality (Mean Average Precision)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Punyakanok et al. 04</td>
<td>0.35</td>
</tr>
<tr>
<td>Wang et al. 07</td>
<td>0.45</td>
</tr>
<tr>
<td>Wang et al. 07 + WN</td>
<td>0.55</td>
</tr>
<tr>
<td>Tree edit model</td>
<td>0.75</td>
</tr>
</tbody>
</table>
Answer Selection Results

Punyakanok et al. 04
Tree edit distance

Wang et al. 07
Quasi-synchronous Grammar to model syntactic alignments

Wang et al. 07 + WN
plus lexical semantics from WordNet

Tree edit model
Answer Selection Results

Punyakanok et al. 04
Tree edit distance

Wang et al. 07
Quasi-synchronous Grammar to model syntactic alignments

Wang et al. 07 + WN
plus lexical semantics from WordNet

Tree edit model
Conclusions

Syntax-based tree edit algorithm for classifying sentence pairs according to semantic relationships.

- **Expressive**: includes tree edits for reordering and moving subtrees.
- **Data Driven**: learns parameters from labeled examples.
- **Useful for various tasks**