Statistical Perspectives on Text-to-Text Generation

Noah Smith
Language Technologies Institute
Machine Learning Department
School of Computer Science
Carnegie Mellon University
nasmith@cs.cmu.edu
I’m A Learning Guy

• I use statistics for prediction
  – Computational social science research: discovery via prediction
  – Predicting the future from text

• Ideal: inputs and outputs
Prediction-Friendly Problems

Predicting the *whole* output from the *whole* input:

- **Linguistic Analysis** (morphology, syntax, semantics, discourse)
  - linguists can reliably annotate data (we think)
- **Machine Translation**
  - parallel data is abundant (in some cases)

- Generation?
But Generation is Unnatural!

• Relevant data do not occur in “nature.”
  – Consider the effort required to build datasets for paraphrase, textual entailment, factual question answering, summarization ...
  – Do people perform these tasks “naturally”?

• Datasets are small and highly task-specific.

• Do statistical techniques even make sense?
Three Kinds of Predictions

Assume a text-text relation of interest.

• Given a pair, does the relationship hold? (Yes or no.)

• Given an input, rank a set of candidates.

• Given an input, generate an output.
Three Kinds of Predictions

Assume a text-text relation of interest.

• Given a pair, does the relationship hold? (Yes or no.)

• Given an input, rank a set of candidates.

• Given an input, generate an output.
Outline

1. Quasi-synchronous grammars
2. Tree edit models
3. A foray into text-to-text generation
Synchronous Grammar

• Basic idea: one grammar, two languages.

\[
\begin{align*}
VP & \rightarrow \text{ne } V_1 \text{ pas } VP_2 / \text{not } V_1 \text{ VP}_2 \\
NP & \rightarrow N_1 A_2 / A_2 N_1
\end{align*}
\]

• Many variations:
  – formal richness (rational relations, context-free, ...)
  – rules from experts, treebanks, heuristic extraction, rich statistical models, ...
  – linguistic nonterminals or not
Quasi-Synchronous Grammar

• Compare:

\[ p(G = g, E = e) \]

\[ p(E = e \mid G = g) \]

• Developed by David Smith and Jason Eisner (SMT workshop 2006).
Quasi-Synchronous Grammar

• Basic idea: one grammar per source sentence.

(S₁ Je (VP₄ ne₅ (V₆ veux) pas₇ (VP₈ aller à l’ (NP₁₂ (N₁₃ usine) (A₁₄ rouge )) )) )

\[
\begin{align*}
\text{VP}_4 & \rightarrow \text{not}_{5,7} \text{ V}_6 \text{ VP}_8 \\
\text{NP}_{12} & \rightarrow \text{A}_{14} \text{ N}_{13}
\end{align*}
\]

• Doesn’t have to be CFG! We use dependency grammar.
Quasi-Synchronous Grammar

• The grammar is determined by the input sentence and only models output language.
  – Generalizes IBM models.
• Allows loose relationship between input and output.
  – “Divergences,” which we think of as non-standard configurations.
  – By disallowing some relationships, we can simulate stricter models; we explored this a good bit in MT ...
Aside: Machine Translation

- The QG formalism originated in translation research (D. Smith and Eisner, 2006).
- Gimpel and Smith (EMNLP 2009): QG as a framework for translation with a blend of dependency syntax features and phrase features. *Generation by lattice parsing.*
- Gimpel and Smith (EMNLP 2011): QG on phrases instead of words shown competitive for Chinese-English and Urdu-English.
Paraphrase (Basic Model)

Quasi-synchronous Grammar

\[ p(S_2 = s_2 \mid S_1 = s_1) \]

Note: Wu (2005) explored a *synchronous* grammar for this problem.
Alignment

Quasi-synchronous Grammar

s₁

s₂

deriveYon

fill

documentation event: “word aligned to fill is a synonym”

complete
Parent-Child Configuration

Quasi-synchronous Grammar

\[ s_1 \]

\[ s_2 \]

Derivation event:
“complete and its dependent are in the parent-child configuration”

Questionnaire

Complete

Fill

Questionnaire
Child-Parent Configuration

Quasi-synchronous Grammar

s₁

s₂

dozens

wounded

of

injured

dozens
Grandparent-Child Configuration

Quasi-synchronous Grammar

$s_1$

$s_2$

chief

will

will

will

Clinton

Secretary
C-Command Configuration

Quasi-synchronous Grammar

\[ s_1 \]

\[ s_2 \]

necessary

collected

approaching twice the 897,158 needed
Same Node Configuration

Quasi-synchronous Grammar

$\text{s}_1$  $	ext{s}_2$

first-quarter

first

quarter
Sibling Configuration

Quasi-synchronous Grammar

U.S.\textsuperscript{1}s

U.S.

massive

treasury

treasury

refunding
Probabilistic QG

• Probabilistic grammars – well known from parsing.
• From “parallel data,” we can learn:
  – relative frequencies of different configurations for different words
  – includes basic syntax (POS, dependency labels)
• We can also incorporate:
  – lexical semantics features that notice synonyms, hypernyms, etc.
  – named entity chunking
Generative Story (Paraphrase)

Base grammar

\[ p(S_1 = s_1) \]

Paraphrase Quasi-synchronous Grammar

\[ p(\text{paraphrase}) \]

\[ p(S_2 = s_2 \mid S_1 = s_1, \text{paraphrase}) \]
Generative Story (Not Paraphrase)

- Base grammar
  - \( p(S_1 = s_1) \)
  - \( p(\text{not paraphrase}) \)
  - \( p(S_2 = s_2 \mid S_1 = s_1, \text{not paraphrase}) \)

- Quasi-synchronous Grammar
  - \( p(S_1 = s_1) \)
  - \( p(\text{not paraphrase}) \)
  - \( p(S_2 = s_2 \mid S_1 = s_1, \text{not paraphrase}) \)
“Not Paraphrase” Grammar?

• This is the result of opting for a fully generative story to explain an unnatural dataset.
  – See David Chen and Bill Dolan’s (ACL 2011) approach to building a better dataset!

• We must account, probabilistically, for the event that two sentences are generated that are not paraphrases.
  – (Because it happens in the data!)
  – Generating twice from the base grammar didn’t work; in the data, “non paraphrases” look much more alike than you would expect by chance.
“Not Paraphrase” Model

We Didn’t Use

Base grammar

\[ p(S_1 = s_1) \]

\[ p(S_2 = s_2) \]

\[ p(\text{not paraphrase}) \]
Notes on the Model

• Although it is generative, we train it discriminatively (like a CRF).

• The correspondences (alignment) between the two sentences is treated as a hidden variable.
  – We sum it out during inference; this means all possible alignments are considered at once.
  – This is the main difference with other work based on overlap features.
But Overlap Features are Good!

• Much is explained by simple overlap features that don’t easily fit the grammatical formalism (Finch et al., 2005; Wan et al., 2006; Corley and Mihalcea, 2005).

• Statistical modeling with a *product of experts* (i.e., two models that can veto each other) allowed us to incorporate shallow features, too.

• **We should not have to choose between two good, complementary representations!**
  – We just might have to *pay* for it.
Paraphrase Identification
Experiments

Test set: N = 1,725

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>p-Precision</th>
<th>p-Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>all paraphrase</td>
<td>66.49</td>
<td>66.49</td>
<td>100.00</td>
</tr>
<tr>
<td>Wan et al. SVM (reported)</td>
<td>75.63</td>
<td>77.00</td>
<td>90.00</td>
</tr>
<tr>
<td>Wan et al. SVM (replication on our test set)</td>
<td>75.42</td>
<td>76.88</td>
<td>90.14</td>
</tr>
<tr>
<td>Wan-like model</td>
<td>75.36</td>
<td>78.12</td>
<td>87.74</td>
</tr>
<tr>
<td>QG model</td>
<td>73.33</td>
<td>74.48</td>
<td>91.10</td>
</tr>
<tr>
<td>PoE (QG with Wan-like model)</td>
<td>76.06</td>
<td>79.57</td>
<td>86.05</td>
</tr>
<tr>
<td>Oracle PoE</td>
<td>83.19</td>
<td>100.00</td>
<td>95.29</td>
</tr>
</tbody>
</table>
Comments

• From a modeling point of view, this system is rather complicated.
  - Lots of components!
  - Training latent-variable CRFs is not for everyone.

• I’d like to see more elegant ways of putting together the building blocks (syntax, lexical semantics, hidden alignments, shallow overlap) within a single, discriminative model.
Jeopardy! Model
QG for QA

• Essentially the same model works quite well for an answer selection task.
  – (I have the same misgivings about the data.)

• Briefly: learn $p(\text{question} \mid \text{answer})$ as a QG from question-answer data.
  – Then rank candidates.

• Full details in Wang, Mitamura, and Smith (EMNLP 2007).
Question-Answer Data

• Setup from Shen and Klakow (2006):
  – Rank answer candidates

• TREC dataset of just a few hundred questions with about 20 answers each; we manually judged which answers were correct (around 3 per question).

• Very small dataset!
  – We explored adding in noisily annotated data, but got no benefit.
Answer Selection Experiments

• Test set: \( N = 100 \)

<table>
<thead>
<tr>
<th>Model</th>
<th>No Lexical Semantics</th>
<th>With WordNet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAP</td>
<td>MRR</td>
</tr>
<tr>
<td>TreeMatch</td>
<td>38.14</td>
<td>44.62</td>
</tr>
<tr>
<td>Cui et al. (2005)</td>
<td>43.50</td>
<td>55.69</td>
</tr>
<tr>
<td>QG model</td>
<td>48.28</td>
<td>55.71</td>
</tr>
</tbody>
</table>
QG: Summary

• QG is an elegant and attractive modeling component.
  – Really nice results on an answer selection task.
  – Okay results on a paraphrase identification task.

• Frustrations:
  – Integrating representations should be easier.
  – Is the model intuitive?
Outline

✓ Quasi-synchronous grammars
2. Tree edit models
3. A foray into text-to-text generation
A Different Approach: Tree Edits

I’m Mike Heilman, and I think those quasi-synchronous models are more complicated than they need to be.

• Full details in Heilman and Smith (NAACL 2010).
Tree Edit Models

• There are many algorithms for aligning trees or minimizing various tree edit distances (Klein, 1989; Zhang and Shasha, 1989).

• These allow deletion, insertion, and relabeling operations.
  – Simple, intuitive operations that transform the sentence incrementally.

• As noted in the QG work, movement is also desirable.
  – You can’t have that and stay efficient.
An Example from Entailment

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

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

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

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

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

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

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

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

built the home for his daughter off, as Pierce lives near Rossville Blvd.
An Example from Entailment

built the home for his daughter off, as Pierce lives near Rossville Blvd.
An Example from Entailment

built the home for his daughter off, as Pierce lives near Rossville Blvd.
An Example from Entailment

built the home for his daughter off,
as Pierce lives near Rossville Blvd.
An Example from Entailment

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

Pierce lives near Rossville Blvd.
Sketch of the Approach

1. Find a tree edit sequence for the sentences, allowing all the operations we want.
   – We use greedy heuristic search.
   – Don’t worry about whether it’s the “right” one.
2. Calculate features on the tree edit sequence.
3. Use a logistic regression model to classify the relationship.
Operations on Dependency Trees

- Insert child.
- Insert parent.
- Delete leaf.
- Delete and merge.
- Relabel node.
- Relabel edge.

- Move subtree.
- New root.
- Move sibling.
Heuristic Search

• Greedy best-first search (Pearl, 1984).
• Heuristic: 1 - Collins and Duffy’s (2001) tree kernel, normalized.
  – Completely different context!
  – We use it as a similarity function from the candidate transformed sentence to the true output.
  – Our kernel is based on Moschitti (2006) and Zelenko et al. (2003).
• Constraints (in brief): don’t insert elements not in the target; new edges take the most frequent label for the child POS.
• Maximum number of iterations (about 5 seconds per sentence pair). Fails less than 0.1% of the time.
33 Features of Edit Sequences

• Number of edits total, and by type
• Number of unedited nodes: total, verbs, nouns, numbers, proper nouns
• Relabel: same POS, same lemma, noun-to-pronoun, change of proper noun, numeric change greater than 5%
• Insert: noun-or-verb, proper noun
• Remove: noun-or-verb, proper noun, subject, object, verb complement, root edge
• Relabel (from or to): subject, object, verb complement, root edge
• Search failure
Experimental Notes

• Direction:
  – For entailment, from premise to hypothesis.
  – For paraphrase, both directions (double the features).
  – For answer selection, answer to question.
## RTE-3 Experiments

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harmeling (2007) - less general operations</td>
<td>59.5</td>
<td>66.49</td>
<td>100.00</td>
</tr>
<tr>
<td>de Marneffe et al. (2006) – align and classify</td>
<td>60.5</td>
<td>61.8</td>
<td>60.2</td>
</tr>
<tr>
<td>MacCartney and Manning (2008) – natural logic</td>
<td>59.4</td>
<td>70.1</td>
<td>36.1</td>
</tr>
<tr>
<td>MacCartney and Manning (2008) – hybrid</td>
<td>64.3</td>
<td>65.5</td>
<td>63.9</td>
</tr>
<tr>
<td>Tree edit model</td>
<td>62.8</td>
<td>61.9</td>
<td>71.2</td>
</tr>
</tbody>
</table>
## Paraphrase Identification Experiments

- **Test set:** $N = 1,725$

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>p-Precision</th>
<th>p-Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>all paraphrase</td>
<td>66.49</td>
<td>66.49</td>
<td>100.00</td>
</tr>
<tr>
<td>Wan et al. SVM (reported)</td>
<td>75.63</td>
<td>77.00</td>
<td>90.00</td>
</tr>
<tr>
<td>Wan et al. SVM (replication on our test set)</td>
<td>75.42</td>
<td>76.88</td>
<td>90.14</td>
</tr>
<tr>
<td>Wan-like model</td>
<td>75.36</td>
<td>78.12</td>
<td>87.74</td>
</tr>
<tr>
<td>QG model</td>
<td>73.33</td>
<td>74.48</td>
<td>91.10</td>
</tr>
<tr>
<td>PoE (QG with Wan-like model)</td>
<td>76.06</td>
<td>79.57</td>
<td>86.05</td>
</tr>
<tr>
<td>Tree edit model</td>
<td>73.3</td>
<td>76.2</td>
<td>87.0</td>
</tr>
</tbody>
</table>
Answer Selection Experiments

- Test set: $N = 100$

<table>
<thead>
<tr>
<th>Model</th>
<th>MAP</th>
<th>MRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>TreeMatch</td>
<td>38.14</td>
<td>44.62</td>
</tr>
<tr>
<td>with WordNet</td>
<td>41.89</td>
<td>49.39</td>
</tr>
<tr>
<td>Cui et al. (2005)</td>
<td>43.50</td>
<td>55.69</td>
</tr>
<tr>
<td>with WordNet</td>
<td>42.71</td>
<td>52.59</td>
</tr>
<tr>
<td>QG model with lex. sem. ablated</td>
<td>48.28</td>
<td>55.71</td>
</tr>
<tr>
<td>QG model, full</td>
<td>60.29</td>
<td>68.52</td>
</tr>
<tr>
<td>Tree edit model</td>
<td>60.91</td>
<td>69.17</td>
</tr>
</tbody>
</table>
Advantages of Tree Edit Model

• Very, very simple.
  – No lexical semantics, Bleu scores, hidden variable modeling, ...
  – ... but could be extended with these things.

• Learned models for the three tasks were highly similar.

• Intuitive way of breaking down the problem
Toward Generation

• Both quasi-synchronous grammar and tree edit models suggest ways of going about generating output.
  – QG: take input, build grammar, “parse $\Sigma^*$.”
    This is sort of what we aim for in MT.
  – TE: search for high scoring transformations.
    Totally untested idea.
Outline

✓ Quasi-synchronous grammars
✓ Tree edit models
3. A foray into text-to-text generation
Finally, Generation

Two cases:

• Heilman and Smith (NAACL 2010) and Heilman (2011): factual question generation

• In brief: Martins and Smith (ILP Workshop 2009): sentence extraction and summarization
• in summarization paper, note the difficulty of not having a single dataset. scarce gains. could revisit this with better inference techniques (Lagrangian relaxation, etc.)
  – mention new Berkeley work that does this better?
• in question generation research: reliance on human judgments. still may be better than trying to build annotated data up front. play up that many errors resulted from parsing/analysis problems. could not get coref to help.
Question Generation

• Our formulation of the task:
• Given a document, generate questions that could be used to check comprehension.
  – Imagine a teacher who wants students to get reading practice on material of their choice, or current events. Can we help the teacher write a quiz?

  (Historical aside: this developed out of an undergrad course project on question answering!)
How It Works

1. Extract sentences.
Example

• Monrovia was named after James Monroe, who was president of the United States in 1922.
• Monrovia was named after James Monroe.
• Was Monrovia named after James Monroe.
• Was Monrovia named after who.
• Who was Monrovia named after?
1. Sentence Extraction

- Related to textual entailment, except we’re generating entailments.
- Preprocessing: parsing, supersense tagging, and coreference.
- Examples of operations:
  - removing discourse markers and adjuncts
  - splitting conjunctions
  - extract presupposed statements from certain well-catalogued constructions (Levinson, 1983)
  - pronouns replaced by more informative NPs (like Nenkova, 2008); using coreference
2. Question Formation

• Largely driven by syntax.
  – Robust, general rules written in a clean formalism, tregex.
  – Some semantic effects missed; overgenerates.

• Steps:
  1. Mark NPs, PPs, and subordinate clauses that can’t be answer phrases
  2. Pick an answer phrase, generate question phrase
  3. Verb decomposition, aux.-inversion
  4. Substitute question phrase for answer phrase
Linguistics Helps!

• If you took a GB-oriented syntax class, you could be forgiven for thinking linguistics was the study of questions.

*What does Chris like the woman who wears?*
*What does Dipanjan wonder where Mike went to buy?*
*Who do you believe that came to my talk?*

• Novel (to my knowledge): formulating these constraints in Tregex (Levy and Andrew, 2006).
3. Ranking

• Gathered human scores (1-5) of question quality.
  – In earlier work, we had them mark different kinds of errors; this was not helpful for the overall system and was more expensive.

• Learn a regression model from feature representation of question-answer to human acceptability scores.

• Rank on acceptability prediction.
<table>
<thead>
<tr>
<th>Source</th>
<th>Question</th>
<th>Mean Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>In 1924 the site was chosen to serve as the capital of the new Tajik Autonomous S. S. R. ..., and rapid industrial and population growth followed.</td>
<td>What followed?</td>
<td>1.00</td>
</tr>
<tr>
<td>Parliament offered the Crown not to James’s eldest son ... but to William and Mary as joint Sovereigns.</td>
<td>Who did Parliament offer the Crown not to?</td>
<td>2.00</td>
</tr>
<tr>
<td>The National Archives has exhibits of historical documents.</td>
<td>What does the National Archives have exhibits of?</td>
<td>3.00</td>
</tr>
<tr>
<td>After the People’s Republic of China took control of Tibet in 1950 and suppressed a Tibetan uprising in 1959, the passes into Sikkim became a conduit for refugees from Tibet.</td>
<td>What did the People’s Republic of China take control of in 1950?</td>
<td>3.67</td>
</tr>
<tr>
<td>Asmara was a village of the Tigre people until the late 19th century.</td>
<td>What was Asmara until the late 19th century?</td>
<td>4.00</td>
</tr>
<tr>
<td>Each year the banquet for the winners of the Nobel Prizes is held in City Hall.</td>
<td>Where is the banquet for the winners of the Nobel Prizes held?</td>
<td>4.67</td>
</tr>
</tbody>
</table>
Agreement

• Fleiss $\kappa$ in the 0.5-0.6 range, for various ways of making the comparison.

• Moderately difficult task.

• We were more careful in choosing annotators for the test set.
179 Question Quality Features

- Length of question, answer phrase, source
- Which WH word?
- Negation?
- Language model probability
- Grammatical phrase types in the answer
- Tense of the main verb
- Main verb is a form of *be*?
- Which sentence transformations were applied in stage 1?
- WH is subject?
- “Vagueness” features (e.g., pronouns, common nouns without modification)
Intrinsic Evaluation

• We considered sentence-level and document level tasks, and a few different datasets (encyclopedic text, elementary version, and Wikipedia).
• Precision at 5 is about 49% on a document-level evaluation.
• Michael’s thesis quantifies the benefits of different components.
  – Ranking is very important.
• See thesis for lots of error analysis.
  – Punchline: keep working on core NLP.
User Study

• 17 teachers were given articles and asked to write quizzes (three questions).

• In one condition they got to use a tool that suggested questions. Control: no suggestions.

• Measured time, self-reported mental effort, question acceptability.
1. Mouse over a sentence in the article to see suggested questions for that sentence (click to “lock” the sentence).

2. View ranked lists of suggested questions here.

3. Click on questions to select them (i.e., to add them to your quiz).

4. View your list of selected questions at the bottom of the screen.

5. Add your own questions by clicking on “add your own” and typing them in.

6. Revise questions by clicking on the question or answer textboxes. Your revisions are saved automatically.

7. Search for questions containing certain keywords.

8. Click on shortcuts to see questions for the whole article (not just for 1 sentence).
Results

• Time: 5.0 minutes reduced to 3.8.
• Small, significant reduction in self-reported mental effort.
• Small, insignificant drop in acceptability rate; major change in distribution of questions.
  – Suggested questions led teachers to make easier quizzes.
• 10/17 would use the tool; 16/17 found it easy to use.
A Second Foray

• (Going beyond single sentences.)
• Martins and Smith (ILP workshop 2009): a joint model of sentence selection and sentence compression for extractive summarization.
  – This is hard. We used integer linear programming to solve the problem jointly, but learned two separate models on two separate datasets.
  – See more recent work by Berg-Kirkpatrick (ACL 2011) that overcomes the data problem.
Conclusions (1)

• Statistical models are building blocks, not black boxes.
  – We can put them together in naïve ways or sophisticated ways.
  – They don’t require us to forgo good linguistic representations.
  – Talk with the parsing and machine learning people!
Conclusions (2)

• Small, noisy, imperfect data scenarios: need more knowledge in the model.
  – I talked about formalisms, features, informed overgeneration, ...
  – We should also think about: priors, exploiting raw data, ...

• But building new datasets is honest work.
Conclusions (3)

• Predictive tasks as a useful abstraction that help us design better models that work for a range of problems.
• But let’s not get stuck on the same tasks!
Acknowledgments

Student collaborators:
- Dipanjan Das
- Kevin Gimpel
- Michael Heilman
- André Martins
- Mengqiu Wang (Stanford)

Sponsors: ICTI, NSF, Google, DARPA