Abstractive Summarization and Natural Language Generation

COMP-550

Nov 16, 2017

Outline

A3 reading discussion

Steps in NLG

Canned Text and Template Filling

Surface realization

- FUF/SURGE

Text-to-text generation

- Sentence compression
- Sentence fusion

A3 Reading Discussion

1. How does the reading relate to these concepts we have discussed in class?
   - Vector-based semantics
   - Principle of compositionality
2. What are some of the strengths of the proposed approach?
3. What are some of the limitations of the proposed approach?
4. What questions do you have about the paper?

Some of Your Questions and Remarks

Compositionality operator and sentence length:

- Does multiplication work for long sentences?

More recent methods for vector composition

- Any ideas or suggestions?
- What are the pros and cons of using such methods?

Multi-Document Summarization

Additional issues to consider:

- Conflicting or contradictory information
- Redundancy between documents
- Combining information from multiple documents

But the second point can actually work to our advantage

- If everybody is talking about the same thing, that thing is likely to be important information.

Conroy et al., 2006

This system combines the topic signature method, a sophisticated non-redundancy module, and the following eliminations:

- Gerund clauses

  Sally went to the store, skipping on one leg.

- Restricted relative-clause appositives

  Bob, who is the president of the club, disagreed.

- Intra-sentential attribution

  They would never do that, she said, without consulting us.

- Lead adverbs

  Hopefully, we will find a solution.

Performance

This simple method (with a few other details), achieves near-human performance on ROUGE-1:

Submission	Mean	95% CI Lower	95% CI Upper
F	0.36787	0.34442	0.39467
B	0.36126	0.33387	0.38754
O (ω)	0.35810	0.34263	0.37330
H	0.33871	0.31540	0.36423
A	0.33289	0.30591	0.35759
D	0.33212	0.30805	0.35628
E	0.33277	0.30959	0.35687
C	0.30237	0.27863	0.32496
G	0.30909	0.28847	0.32987
$\omega_{qs}^{(pr)}$	0.308	0.294	0.322
peer 65	0.308	0.293	0.323
SumBasic	0.302	0.285	0.319
peer 34	0.290	0.273	0.307
peer 124	0.286	0.268	0.303
peer 102	0.285	0.267	0.302

Table 4: Average ROUGE 1 Scores with stop words removed for DUC04, Task 2

Extraction vs. Abstraction

Reminder:

Extraction – take snippets from the source text and put them in the summary

Abstraction – compose novel text not found in the source

- Allows better aggregation of information
- Requires **natural language generation**

Natural Language Generation

Let's compare understanding and generation

Concerns of NLU:

- Ambiguity (e.g., get all possible parses)
- Disambiguation
- Underspecification

Concerns of NLG:

- Selecting appropriate content
- Selecting appropriate form to express content

Canned Text


MARIO
058700
×20
WORLD
1-4
TIME
200
THANK YOU MARIO!
BUT OUR PRINCESS IS IN
ANOTHER CASTLE!

Weather Tweets: Template Filling

Good for restricted domains.

Environment Canada's weather alert Twitter feeds:

https://twitter.com/ECAlertQC147

What is the generation template?

Steps in NLG

One potential architecture for an NLG system:

1. Content selection
2. Document structuring
3. Microplanning
4. Surface realization

Content Selection

Deciding what to say

Ingredients:

- Communicative goal
- Knowledge about the world

Application-specific

- How did we approach content selection last class in multi-document summarization?

Document Structuring

Deciding how to structure the contents of the output

What order should they be presented in? Some factors:

- Importance of the concepts
- Discourse relations
- Coherence

e.g., **Argumentation Theory** gives some guidelines on how to arrange information

- Present main claims first
- Arrange and discuss supporting evidence
- Present and debate opposing evidence

(Carenini and Moore, 2006)

Microplanning

Selecting lexical items

- (BLZRD, -5, -10, 30km/h, MONTREAL) -> *blizzard, low, high, wind speed, Montreal*

Deciding how they fit together into clauses and sentences (**sentence planning** or **aggregation**)

- First sentence: present location and time that weather forecast pertains to
- Second sentence: present details of forecast

Generating referring expressions

- *Justin Pierre James Trudeau PC MP; Justin Trudeau; the Prime Minister; Mr. Trudeau; that guy; he; him*

Surface Realization

Convert fully specified discourse plan to output form (individual sentences, or other kinds of output)

Different possible levels of input specification:

- Highly detailed semantic structure, with all decisions made already (lexical items, tense, aspect and mood of verbs, referring expressions, etc.)
- Shallower kinds of semantics (e.g., similar to a dependency tree)

Reusable Components

There have been few standard tools or task definitions in NLG:

- Referring expression generation
- Surface realization

Let's look at a surface realization system: FUF/Surge

FUF/SURGE

A cascade of deterministic rules to convert a structured semantic representation to a string:

(Elhadad and Robin, 1996)

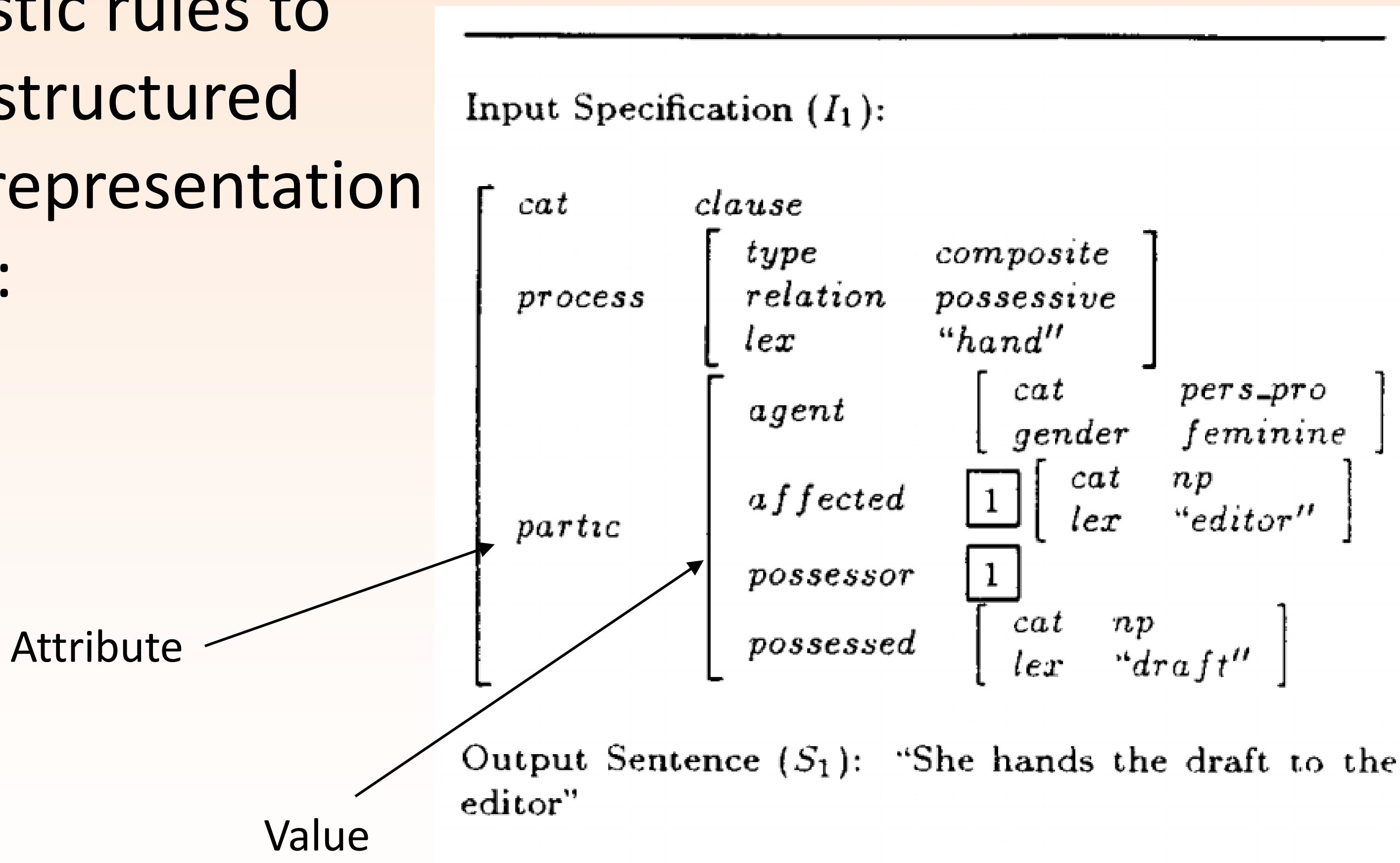


Figure 1: An example SURGE I/O

Components in FUF/SURGE

1. Map thematic structures (i.e., semantic roles) to syntactic roles

   e.g., agent -> subject

2. Handle syntactic alternations

   e.g., active-passive, dative alternation

3. Fill in default features, agreement features

   e.g., NPs are definite, if not otherwise specified

   subject and verb agree in number

4. Handle closed-class words

   e.g., [cat pers_pro, gender feminine] -> *she*

Components in FUF/SURGE

5. Order components with respect to each other

   e.g., subject > verb-group > indirect-object > direct object

6. Fill in inflections

   e.g., *to hand -> hands*

7. Linearize the tree into the final string, using precedence constraints

A Matter of Inputs

Traditional NLG: data-to-text

What about starting from other text?

e.g., summarization can be seen as text-to-text generation

Advantages?

Disadvantages?

Goals of Text-to-Text Generation

Since we are already starting with some text, there must be something about the input that we are changing to produce the output:

- Length

  Informative summarization

- Complexity

  Text simplification

- Other factors?

Sentence Compression

(Knight and Marcu, 2000)

Assumptions:

- May drop some words in original sentence
- Remaining words stay in the same order

Example:

Orig: Beyond the basic level, the operations of the three products vary widely.

Noisy-C: The operations of the three products vary widely.

Human: The operations of the three products vary widely.

Noisy-Channel Model

View as a **noisy-channel model**

$$\text{Input, short string } s \xrightarrow{\text{Noisy channel}} \text{Output, long string } t$$

Source model $P(s)$

Channel model $P(t|s)$

Compression = finding $\text{argmax}_s\, P(s)P(t|s)$

Components of Model

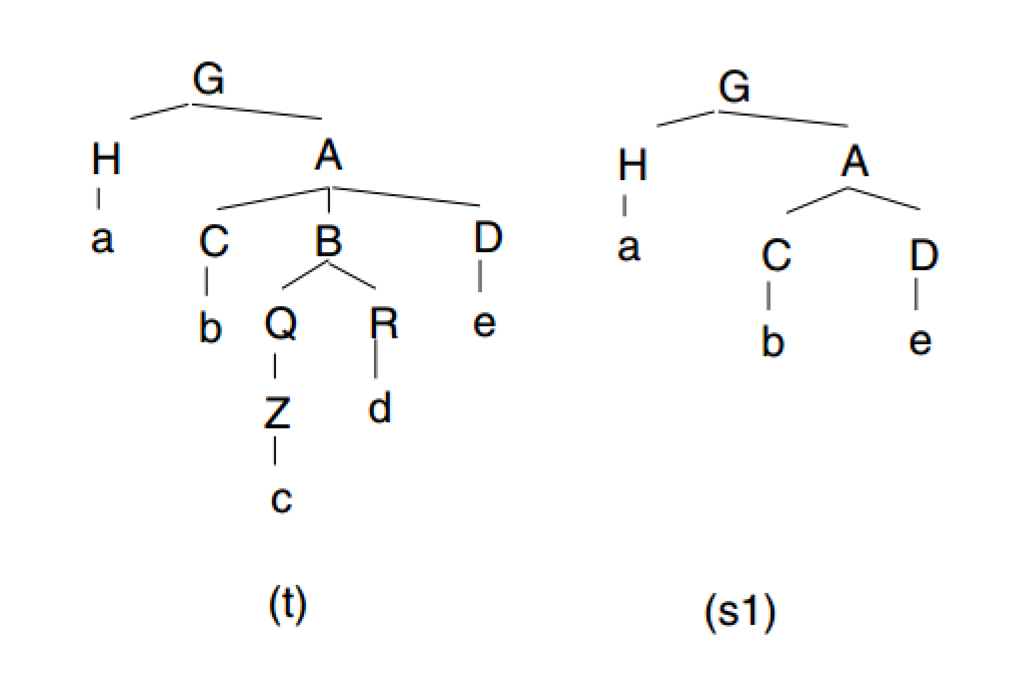
$P(s)$ – language model – combine a bigram language model with a PCFG language model

$P(t|s)$ – probably of long string given short string

View as a series of PCFG rule expansions:

Assign a probability to each operation that maps from a rule in s to a rule in t.

Example: P(s1)



$P(s1)$ =

$P(TOP \rightarrow G)$

$P(G \rightarrow H\ A)$

$P(H \rightarrow a)$

$P(A \rightarrow C\ D)$

$P(C \rightarrow b)$

$P(D \rightarrow e)$

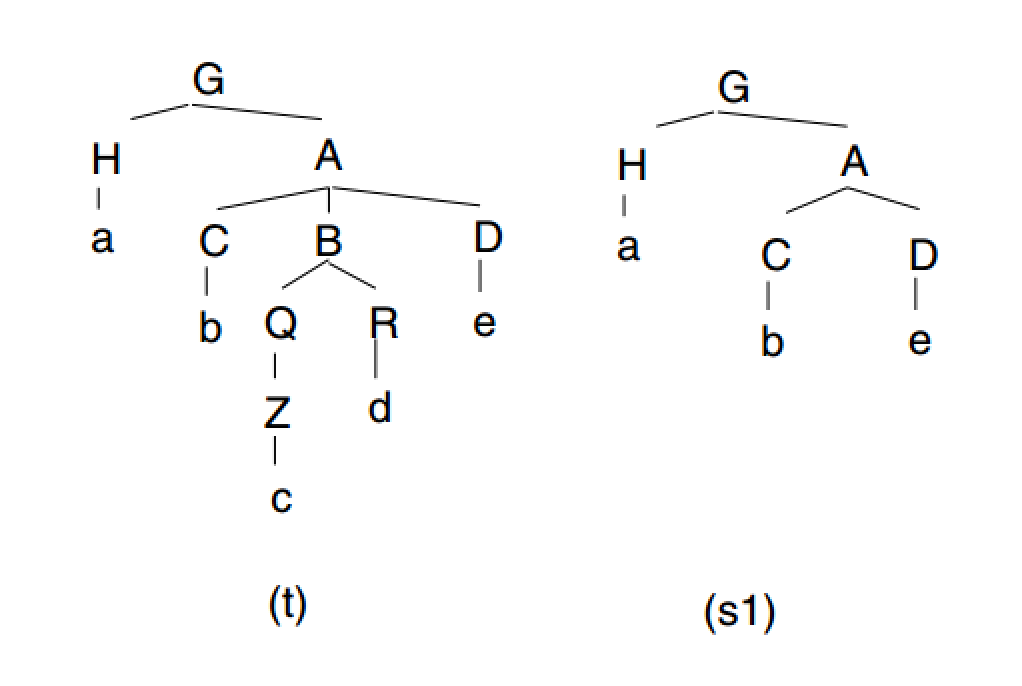
$P(a|START)$

$P(b|a)$

$P(e|b)$

$P(END|e)$

Example P(t|s1)



$$P(t|s1) =$$
$$P(G \rightarrow HA|G \rightarrow HA)$$
$$P(A \rightarrow CBD|A \rightarrow CD)$$
$$P(B \rightarrow QR)$$
$$P(Q \rightarrow Z)$$
$$P(Z \rightarrow c)$$
$$P(R \rightarrow d)$$

More Details

To learn the model probabilities, need a corpus of sentences with simplifications.

Need a little more work to:

- Align PCFG productions between s and t
- Efficiently search for the best possible s given a trained model
- See paper for details

Sample Output

Orig: Arborscan is reliable and worked accurately in testing, but it produces very large dxf files.

Noisy-C: Arborscan is reliable and worked accurately in testing, but it produces very large dxf files.

Human: Arborscan produces very large dxf files.

Orig: Many debugging features, including user-defined break points and variable-watching and message-watching windows, have been added.

Noisy-C: Many debugging features, including user-defined points and variable-watching and message-watching windows, have been added.

Human: Many debugging features have been added.

Original: Beyond the basic level, the operations of the three products vary widely.

NC/Human: The operations of the three products vary widely.

Sentence Fusion

(Barzilay and McKeown, 2005; Filippova and Strube, 2008; Thadani and McKeown, 2013; Cheung and Penn, 2014)

Combine information from multiple sentences. Take a *union* of information.

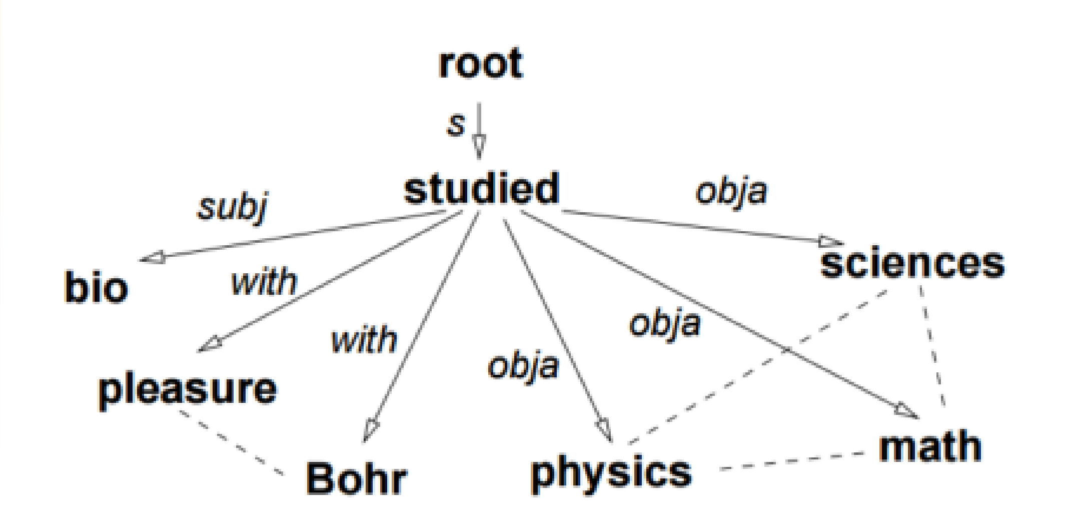
> *Bohr studied at the University of Copenhagen and got his PhD there.*
>
> *After graduating, he studied physics and mathematics at the University of Copenhagen.*
>
> *After graduating, Bohr studied physics and mathematics at the University of Copenhagen and got his PhD there.*

Step 1: Sentence Graph

Create a **sentence graph** by merging the input sentences' dependency trees at the nodes with the same words.

e.g.: *He studied sciences with pleasure.*

+ He studied math and physics with Bohr.



(Filippova and Strube, 2008)

Step 2: Extract a New Sentence

Select a subset of nodes in sentence graph that will form a new dependency tree, from which a new sentence can be generated.

Problem: many desiderata and constraints

- Nodes must form a tree
- Selected nodes must contain the important words
- Selected nodes should make sense in relation to each other
- Desired output length

Would like a method that allows us to write down all of these hard and soft constraints

Solution: Integer Linear Programming

For each edge in the sentence graph from word *h* to word *w* with label *l*, create a variable x_{hw}^{l}.

$$x_{hw}^{l} = \begin{cases} 1 & \text{select this edge} \\ 0 & \text{don't select this edge} \end{cases}$$

Optimize the following objective:

$$f(X) = \sum_{x} x_{hw}^{l} \times P(l|h) \times I(w)$$

"Grammaticality" – how often this head word generates a dependent with this label

Importance of the dependent

Constraints in ILP

maximize $f(X) = \sum_x x_{hw}^l \times P(l|h) \times I(w)$

subject to

$$\forall w \in W, \sum_{h,l} x_{hw}^l \leq 1$$

$$\forall w \in W, \sum_{h,l} x_{hw}^l - \frac{1}{|W|} \sum_{u,l} x_{wu}^l \geq 0$$

First constraint ensures each word has at most one head

Second ensures that selected nodes form a connected tree

How would we constrain the number of words in the output?

ILP for NLG

Various other syntactic and semantic constraints

e.g., ensure that conjoints are similar to each other (*math and physics* is likely, *math and Bohr* is unlikely)

In general, ILP is popular for NLG:

- Allows *declarative* specification of diverse objectives and constraints
- Can be solved fairly efficiently using off-the-shelf solvers

  http://lpsolve.sourceforge.net/5.5/

  http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

Brainstorm

How can you formulate multi-document extractive summarization as an ILP? What would be the objective and what would be some constraints?

How can you formulate sentence compression as an ILP? What would be the objective and what would be some constraints?

References

Carenini and Moore. 2006. Generating and evaluating evaluative arguments. *Artificial Intelligence.*

Elhadad and Robin. 1996. An Overview of SURGE: A Reusable Comprehensive Syntactic Realization Component. *INLG*.

Filippova and Strube. 2008. Sentence Fusion via Dependency Graph Compression. *EMNLP.*

Knight and Marcu. 2000. Statistics-based Summarization – Step One: Sentence Compression. *AAAI*.