Text-to-text generation

Micha Elsner

February 19, 2014
Natural language generation

Generation
High-level goal: output some text with specified meaning
  ▶ In a specified style

Why?
  ▶ Linguistics:
    ▶ How do we tell if language has some meaning?
    ▶ How do we identify style?
    ▶ Stress-test our syntax and semantics models
  ▶ Engineering:
    ▶ Machines we can talk to...
    ▶ Variety of “executive assistant” programs we could build
    ▶ ex tutorial dialogue systems for education
Two kinds of generation

**Concept-to-text**

“Specified meaning” in non-linguistic form:
- Databases (sports scores, weather reports)
- Logical/probabilistic format from prototype AI system
- Output of computer vision or other sensor network

Often very domain-specific

**Text-to-text**

“Specified meaning” given by other linguistic resource
- Single sentence
- Whole document

Often very shallow
- Major edits can mess up original...
This lecture

We’re not going to cover concept-to-text

- Ask Michael White
- Recent exciting work in CCG (categorial grammar) framework
Text to text

Given a text, try to preserve meaning (mostly) and:

▶ Make it short (summarization, compression)
▶ Make it simple (simplification)
▶ Make it different (paraphrase)
▶ Conceal the author’s identity (anonymization)

Many other possibilities exist but aren’t generally studied yet:
▶ Opportunity (or disaster) awaits
Compression

Probably about the simplest task out there: Make a sentence shorter

Example

Former Democratic National Committee finance director Richard Sullivan faced more pointed questioning from Republicans during his second day on the witness stand in the Senate’s fund-raising investigation.

- Richard Sullivan Republicans Senate.
- Richard Sullivan faced pointed questioning.
- Richard Sullivan faced pointed questioning from Republicans during day on stand in Senate fund-raising investigation.

Siddharthan, Nenkova and McKeown quoting Greffenstrete
Example 2

PAL, which has been unable to make payments on $2.1 billion in debt, was devastated by a pilots’ strike in June and by the region’s currency crisis, which reduced passenger numbers and inflated costs.

PAL was devastated by a pilots’ strike in June and by the region’s currency crisis.

Siddharthan, Nenkova and McKeown
Example 3

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

- Bigram model: Debugging, user-defined and variable-watching and message-watching, have been.
- Model 1: Many debugging features, including user-defined points and variable-watching and message-watching windows, have been added.
- Model 2: Many debugging features.
- Humans: Many debugging features have been added.

Knight and Marcu
Making a good compressor is partly a matter of prioritizing content
  ▶ TF-IDF and related
But mostly about getting the syntax right
  ▶ Can a sentence end with “have been”? Can that be the whole VP?
N-gram models alone don’t do very well...
  ▶ Can’t keep track even of simple things like number of verbs
So people mostly go for some kind of syntax
Cutting things out

From Berg-Kirkpatrick, Gillick and Klein “Jointly learning to extract and compress”

(a) $y_{s1} = 1$

- He stopped in France.
- $y_{s2} = 0$
- In France he remained.

$B(y) = \{(he, stopped), (stopped, in), (in france)\}$

(b) $y_{w2} = 1$

- $y_{w2} = 1$
- $y_{w3} = 1$

- He stopped in France.

- $y_{w4} = 1$
- $y_{w5} = 1$

- In France he remained.
- $y_{w6} = 0$
- $y_{w7} = 1$
- $y_{w8} = 1$

$B(y) = \{(he, stopped), (stopped, in), (in france), (he remained)\}$

$C(y) = \{(w_6, w_8)\}$
Features for cutting things

When can we get rid of a subtree?

- Based on nonterminal or CFG rule
- Or dependency label
- Based on words underneath
- Based on remaining string LM prob

Use features and weights to score, then make a decision

- Some older systems just use rules
- With only one feature, can count and divide
Estimating weights

Because we have lots of features

- And don’t want to assume they’re independent

We can’t just count and divide

- So use a gradient method
- Calculating the exact gradient would require the normalizing constant
- A sum over all possible compressions
- Which we could obtain using dynamic programming
- But it’s often simpler to just use a few sample compressions
- A “perceptron”-style algorithm like the one you used for max-ent
Estimation (perceptron-style)

Repeat till sufficiently bored (or scores/likelihoods stop changing):

- Pick an example, ask your system to compress it
- If this compression doesn’t match the human one
  - Or the best possible one your system can output
- Calculate the features that distinguish your wrong answer from the right answer
- Features that are ON for wrong and OFF for right have a negative gradient
  - (Should be punished by decreasing weight)
- And vv
Illustration from Elsner and Santhanam sentence fusion paper
Clarke and Lapata compressor (a lot like ours):

- $x_i$ variable: 1 iff $w_i$ in compressed sentence
- Language model incorporated via $z_{ij}$ variable: 1 if $w_i$ and $w_j$ both in compression
  - Just like transition vars in HMM

Then:

$$\max \sum_i s_i x_i + \sum_{ij} - \log P_{\text{bigram}}(w_j|w_i) z_{ij}$$

Subject to usual constraints to make $z$ do what we want
Ensuring grammaticality

Constraints:

- If non-clausal modifier included (adj or n) include head
- If determiner included, include head
- Require negation if head included
- Require possessive if head included
- If verb, take argument
- If clause, take complementizer
- If coordinates, take conjunction
Open issues

Playing around with your own system will probably show you these constraints aren’t sufficient to get grammatical output:

▶ For instance, what about verbs with required PP arguments?
  ▶ “He will beg for money”
▶ Contentless head plus adjective
  ▶ “I want the red one”
▶ Some relative clauses are more necessary than others
  ▶ “I like the dog [that you gave me]” vs “I like the fact [that it’s difficult]”

There’s been work on doing this better, but I think this is still open...
Paraphrasing

“Shorter” is not all that interesting...

Paraphrase

A vaguely-defined task...
- Retain semantics, change words

Example

DATE:NUM1 are killed and around NUM2 injured when suicide bomber blows up his explosive-packed belt at X1 in X2. Palestinian suicide bomber blew himself up at X1 in X2 DATE killing NUM1 and wounding NUM2. Police said Barzilay and Lee

Example 2

“Burst into tears” = “cried”; “comfort” = “console” Callison-Burch (various)
Overlap based

Barzilay and McKeown “Extracting paraphrases from a parallel corpus”
Barzilay and Lee “Learning to paraphrase: an unsupervised approach to multiple-sequence alignment”

Step 1: find a bunch of sentences that look similar

- Look in translations of the same document
  - “And finally, dazzlingly white, it shone above them in the empty [X]”
  - “It appeared white and dazzling in the empty [X]”
- Look in news items about similar events
A Palestinian suicide bomber blew himself up in a southern city Wednesday, killing two other people and wounding 27.

A suicide bomber blew himself up in the settlement of Efrat, on Sunday, killing himself and injuring seven people.

A suicide bomber blew himself up in the coastal resort of Netanya on Monday, killing three other people and wounding dozens more.

A Palestinian suicide bomber blew himself up in a garden cafe on Saturday, killing 10 people and wounding 54.

A suicide bomber blew himself up in the centre of Netanya on Sunday, killing three people as well as himself and injuring 40.
Which substrings can we take?

- Build a classifier
- Using POS tag sequences as features
- To score contexts as “can be a paraphrase”
- Or “cannot”

$NN_0 \text{ POS } NN_1 \rightarrow NN_1 \text{ IN DT } NN_0$

“King’s son” $\rightarrow$ “son of the king”
Or just match up the sequences

Figure 3: Lattice and slotted lattice for the five sentences from Figure 2. Punctuation and articles removed for clarity.
Bannard and Callison-Burch “Paraphrasing with bilingual parallel corpora”
Nowadays can use MT technology to translate...

what is more, the relevant cost dynamic is completely under control
im übrigen ist die diesbezügliche kostenentwicklung völlig unter kontrolle
we owe it to the taxpayers to keep the costs in check
You can do better with syntax

But not all paraphrases are constituents:

- “Create equal” vs “create genuinely fair”

Need a way to label items in the tree that aren’t constituents
Ganitkevich, van Durme and Burch, using Callison-Burch translation techniques

- 6.8M high-confidence paraphrases, 169M full set

Some particle verbs:
- speed up | accelerate
- blow up | explode
- throw up | puke
- set up | establish
- speed up | expedite
- give up | abandon

Some gendered titles:
- chairman | chairperson, chair
- policeman | cop
- craftsman | artisan
- hangman | executioner
- batman | the bat
Cohn and Lapata “Sentence compression beyond word deletion”

- Orig: The scheme was intended for people of poor or moderate means.
- Clarke-like: The scheme was intended for people of poor means.
- Paraphrasing: The scheme was planned for poor people.
- Humans: The scheme was intended for the poor.

Idea: use paraphrases to compress
Framework

Use *tree-to-tree* rewrite rules:

- \( NP \rightarrow (NP \ (CD) \ (ADJP) \ (NNS \ activists)) \)
- \( NNS \rightarrow activists \)
Inference

Search for the set of paraphrase rules that leads to highest-quality compression

- Like a (very complex) parsing problem
- Uses dynamic programming

Exciting because objective doesn’t have to be “shortest”

- Could specify something else...
- Designing other interesting objectives mostly still open
Document level

Summarization: like compression, but documents instead of sentences:

- Take a single document or a set of articles
- Output a short summary
- Summary should cover important information
- And not be redundant
  - Redundant information is boring and takes up space that could be better used
- And be coherent
  - Sentences should relate to one another
Bowing to a court ruling, Congress in 1988 laid out a statutory framework allowing American Indian tribes to offer high-stakes bingo games and casino-style gambling on their historic reservation lands. The lack of such approval violates a federal law, the Indian Gaming Regulatory Act of 1988, which requires an Indian tribe to strike an accord, or compact, with the state in which its reservation lies if it wants to open a casino. Tribal administrator Gary Goforth acknowledged few of the 675 jobs at the tribe’s two financially troubled casinos are filled by tribal members. Indians working on reservations avoid state income taxes but pay federal income and Social Security taxes. The country’s largest tribe, the Navajo, twice rejected gambling. Neah Bay, like many other Native American communities, is struggling with unemployment rates often exceeding 50 percent, deep poverty and the problems of crime, drug and alcohol abuse and domestic violence.
Extraction vs abstraction

Two main approaches:

**Extractive**
Take whole sentences from original document
- Grammaticality is good, but style is usually verbose
- And coherence suffers

**Abstractive**
Use compression and paraphrase to rewrite sentences
Or make up new sentences
- Grammaticality is poor and sometimes semantics are incorrect
- Sentences can be shorter and fit together better
Nenkova and Vanderwende “The impact of frequency on summarization”

▶ An older, simple system

Construct a unigram probability distribution estimated from the input documents:

\[ \hat{P}(w_i) = \frac{\#(w_i)}{n} \]

Let weight of each sentence be equal to average probability of its words:

\[ Q(S_j) = \sum_{w \in S_j} \frac{P(w)}{|S_j|} \]

We want to pick sentences whose words occur a lot in the documents
Selection and redundancy

Take next sentence with highest score:

\[ \text{summ} \leftarrow \max_S Q(S) \]

Now penalize all words we’ve taken (less useful to take them again):

\[ P_{\text{new}}(w) = P(w)^2 \]

This is still a reasonable summarizer

▶ Though one can do better nowadays
Conclusion

Text-to-text generation is an exciting area for computational syntax/semantics

- Simplest thing to do is just delete/select
- More complicated rewrites are possible
- Full spectrum of possible tasks still unexplored