NL → analysis → generation → R
Natural Language Generation

The classical view: $\mathcal{R}$ is a meaning representation language.

▶ Often very specific to the domain.
▶ For a breakdown of the problem space and a survey, see Reiter and Dale (1997)

In 2016: considerable emphasis on text-to-text generation, i.e., transformations:

▶ Translating a sentence in one language into another language
▶ Summarizing a long piece of text by a shorter one
▶ Paraphrase generation (Barzilay and Lee, 2003; Quirk et al., 2004)
Machine Translation
One naturally wonders if the problem of translation could be conceivably treated as a problem in cryptography. When I look at an article in Russian, I say: ‘This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.’
Noisy Channel Models

Review from January 4

A pattern for modeling a pair of random variables, $X$ and $Y$:

\[
\text{source} \longrightarrow Y \longrightarrow \text{channel} \longrightarrow X
\]
Noisy Channel Models

Review from January 4

A pattern for modeling a pair of random variables, $X$ and $Y$:

\[
\text{source} \rightarrow Y \rightarrow \text{channel} \rightarrow X
\]

- $Y$ is the plaintext, the true message, the missing information, the output
Noisy Channel Models

Review from January 4

A pattern for modeling a pair of random variables, $X$ and $Y$:

$$\text{source} \rightarrow Y \rightarrow \text{channel} \rightarrow X$$

- $Y$ is the plaintext, the true message, the missing information, the output
- $X$ is the ciphertext, the garbled message, the observable evidence, the input
Noisy Channel Models

Review from January 4

A pattern for modeling a pair of random variables, $X$ and $Y$:

$$\text{source} \rightarrow Y \rightarrow \text{channel} \rightarrow X$$

- $Y$ is the plaintext, the true message, the missing information, the output
- $X$ is the ciphertext, the garbled message, the observable evidence, the input
- Decoding: select $y$ given $X = x$.

$$y^* = \arg\max_y p(y \mid x)$$

$$= \arg\max_y \frac{p(x \mid y) \cdot p(y)}{p(x)}$$

$$= \arg\max_y \underbrace{p(x \mid y)}_{\text{channel model}} \cdot \underbrace{p(y)}_{\text{source model}}$$
Let $f$ and $e$ be two sequences in $V^\dagger$ (French) and $\bar{V}^\dagger$ (English), respectively.

We’re going to define $p(F \mid e)$, the probability over French translations of English sentence $e$.

In a noisy channel machine translation system, we could use this together with source/language model $p(e)$ to “decode” $f$ into an English translation.

Where does the data to estimate this come from?
IBM Model 2
(Brown et al., 1993)

Let $\ell$ and $m$ be the (known) lengths of $e$ and $f$.

Latent variable $a = \langle a_1, \ldots, a_m \rangle$, each $a_i$ ranging over $\{0, \ldots, \ell\}$ (positions in $e$).

- E.g., $a_4 = 3$ means that $f_4$ is “aligned” to $e_3$.

\[
p(f \mid e, m) = \sum_{a \in \{0, \ldots, n\}^m} p(f, a \mid e, m)
\]

\[
p(f, a \mid e, m) = \prod_{i=1}^{m} p(a_i \mid i, \ell, m) \cdot p(f_i \mid e_{a_i})
\]

\[
= \delta_{a_i \mid i, \ell, m} \cdot \theta_{f_i \mid e_{a_i}}
\]
IBM Model 2, Depicted

PLSA and LDA (topics)

hidden Markov model

IBM 2
Evaluation

Intuition: good translations are **fluent** in the target language and **faithful** to the original meaning.

**Bleu** score (Papineni et al., 2002):
- Compare to a human-generated reference translation
- Or, better: multiple references
- Weighted average of n-gram precision (across different n)

There are some alternatives; most papers that use them report Bleu, too.
Obtaining word alignments in a parallel corpus is a common first step in building a machine translation system.

1. Align the words.
2. Extract and score phrase pairs.
3. Estimate a global scoring function to optimize (a proxy for) translation quality.
4. Decode French sentences into English ones.

(Today we’ll discuss 2–4.)

The noisy channel pattern isn’t taken quite so seriously when we build real systems, but language models are really, really important nonetheless.
Phrases?

Phrase-based translation uses automatically-induced phrases . . . not the ones given by a phrase-structure parser.
# Examples of Phrases

Courtesy of Chris Dyer.

<table>
<thead>
<tr>
<th>German</th>
<th>English</th>
<th>$p(f \mid e)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>das Thema</td>
<td>the issue</td>
<td>0.41</td>
</tr>
<tr>
<td></td>
<td>the point</td>
<td>0.72</td>
</tr>
<tr>
<td></td>
<td>the subject</td>
<td>0.47</td>
</tr>
<tr>
<td></td>
<td>the thema</td>
<td>0.99</td>
</tr>
<tr>
<td>es gibt</td>
<td>there is</td>
<td>0.96</td>
</tr>
<tr>
<td></td>
<td>there are</td>
<td>0.72</td>
</tr>
<tr>
<td>morgen</td>
<td>tomorrow</td>
<td>0.90</td>
</tr>
<tr>
<td>fliege ich</td>
<td>will I fly</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>will fly</td>
<td>0.17</td>
</tr>
<tr>
<td></td>
<td>I will fly</td>
<td>0.13</td>
</tr>
</tbody>
</table>
Phrase-Based Translation Model
Originated by Koehn et al. (2003).

R.v. $A$ captures segmentation of sentences into phrases, alignment between them, and reordering.

$$p(f, a \mid e) = p(a \mid e) \cdot \prod_{i=1}^{\mid a \mid} p(f_i \mid e_i)$$
Extracting Phrases

After inferring word alignments, apply heuristics.
Extracting Phrases

After inferring word alignments, apply heuristics.
Extracting Phrases

After inferring word alignments, apply heuristics.
Extracting Phrases

After inferring word alignments, apply heuristics.
Extracting Phrases

After inferring word alignments, apply heuristics.
Extracting Phrases

After inferring word alignments, apply heuristics.
Extracting Phrases

After inferring word alignments, apply heuristics.
Extracting Phrases

After inferring word alignments, apply heuristics.
Scoring Whole Translations

\[ s(e, a; f) = \log p(e) + \log p(f, a | e) \]

Remarks:

- Segmentation, alignment, reordering are all predicted as well (not marginalized).
- This does not factor nicely.
Scoring Whole Translations

\[ s(e, a; f) = \underbrace{\log p(e)}_{\text{language model}} + \underbrace{\log p(f, a \mid e)}_{\text{translation model}} + \underbrace{\log p(e, a \mid f)}_{\text{reverse t.m.}} \]

Remarks:

- Segmentation, alignment, reordering are all predicted as well (not marginalized).
- This does not factor nicely.
- I am simplifying!
  - Reverse translation model typically included.
Scoring Whole Translations

\[ s(e, a; f) = \beta_{l.m.} \log p(e) + \beta_{t.m.} \log p(f, a | e) + \beta_{r.t.m.} \log p(e, a | f) \]

language model                                    translation model

+ reverse t.m.

Remarks:

- Segmentation, alignment, reordering are all predicted as well (not marginalized).
- This does not factor nicely.
- I am simplifying!
  - Reverse translation model typically included.
  - Each log-probability is treated as a “feature” and weights are optimized for Bleu performance.
Maria no dio una bofetada a la bruja verde

Mary not give a slap to the witch green

did not slap by hag bawdy

no slap to the green witch

did not give the the witch
Decoding: Example

Maria no dio una bofetada a la bruja verde

Mary not give a slap to the witch green

did not slap by hag bawdy
do not slap to the green witch

did not give the the witch
Decoding: Example

Maria no dio una bofetada a la bruja verda

Mary did not give a slap to the witch green

did not slap

no slap
did not give

the green witch
Decoding
Adapted from Koehn et al. (2006).

Typically accomplished with beam search.

Initial state: \( \langle \circ \circ \ldots \circ, "" \rangle \) with score 0
\[ |f| \]

Goal state: \( \langle \bullet \bullet \ldots \bullet, e^* \rangle \) with (approximately) the highest score
\[ |f| \]

Reaching a new state:
- Find an uncovered span of \( f \) for which a phrasal translation exists in the input \((\bar{f}, \bar{e})\)
- New state appends \( \bar{e} \) to the output and “covers” \( \bar{f} \).
Decoding Example

Maria no dio una bofetada a la bruja verda

Mary did not give a slap to the witch green

did not slap by hag bawdy

no slap to the green witch

did not give the

the witch

⟨◦◦◦◦◦◦◦◦◦, “”⟩, 0
Decoding Example

\langle \bullet \circ \circ \circ \circ \circ \circ \circ \circ \circ, \text{“Mary”} \rangle, \quad \log p_{l.m.}(\text{Mary}) + \log p_{t.m.}(\text{Maria} \mid \text{Mary})
Decoding Example

Maria no dio una bofetada a la bruja verda

Mary
did not

\[
\langle \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet, \text{"Mary did not"}\rangle,
\]

\[
\log p_{l.m.}(\text{Mary did not}) + \log p_{t.m.}(\text{Maria | Mary}) + \log p_{t.m.}(\text{no | did not})
\]
Decoding Example

\[ \langle \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots , \text{“Mary did not slap”} \rangle, \]
\[ \log p_{t.m.}(\text{Mary did not slap}) + \log p_{t.m.}(\text{Maria} \mid \text{Mary}) \]
\[ + \log p_{t.m.}(\text{no} \mid \text{did not}) + \log p_{t.m.}(\text{dio una bofetada} \mid \text{slap}) \]
Sometimes phrases are organized hierarchically (Chiang, 2007).

Extensive research on syntax-based machine translation (Galley et al., 2004), but requires considerable engineering to match phrase-based systems.

Recent work on semantics-based machine translation (Jones et al., 2012); remains to be seen!

Neural models have become popular and are competitive (e.g., Devlin et al., 2014); impact remains to be seen!

Some good pre-neural overviews: Lopez (2008); Koehn (2009)
Summarization
Automatic Text Summarization

Survey from before statistical methods came to dominate: Mani, 2001

Parallel history to machine translation:
- Noisy channel view (Knight and Marcu, 2002)
- Automatic evaluation (Lin, 2004)

Differences:
- Natural data sources are less obvious
- Human information needs are less obvious

We’ll briefly consider two subtasks: compression and selection
Sentence Compression as Structured Prediction
(McDonald, 2006)

Input: a sentence

Output: the same sentence, with some words deleted

McDonald’s approach:
- Define a scoring function for compressed sentences that factors locally in the output.
  - He factored into bigrams but considered input parse tree features.
- Decoding is dynamic programming (not unlike Viterbi).
- Learn feature weights from a corpus of compressed sentences, using structured perceptron or similar.
Sentence Selection

Input: one or more documents and a “budget”

Output: a within-budget subset of sentences (or passages) from the input

Challenge: diminishing returns as more sentences are added to the summary.

Classical greedy method: “maximum marginal relevance” (Carbonell and Goldstein, 1998)

Casting the problem as submodular optimization: Lin and Bilmes (2009)

Joint selection and compression: Martins and Smith (2009)
Readings and Reminders

- Collins (2013)
- Submit a suggestion for an exam question by Friday at 5pm.
- Your project is due Wednesday.


