Machine Translation I

Wei Xu

(many slides from Greg Durrett)
Semester So-far

- Machine Learning Models
  - Linear models: Naive Bayes, Logistic Regression, SVM, Perceptron
  - Neural models: FeedForward Neural Networks, Back-prop, ...

- Sequence Models (NER, POS tagging, etc)
  - Hidden Markov Model, Viterbi Algorithm, Conditional Random Fields

- Word Embeddings

- Recurrent NN, Convolutional NN, Neural CRF
Rest of the Semester

- Applications in Natural Language Processing
  - Machine Translation (2 weeks)
  - Information Extraction
  - Reading Comprehension
  - Automatic Summarization (if time)
  - Dialog System
  - Contextual Word Embeddings
  - etc.
This Lecture

- MT and evaluation
- Word alignment
- Language models
- Phrase-based decoders
- Syntax-based decoders
MT Basics
Trump and the Pope family watch a hundred years a year in the White House balcony.
I have a friend => \( \exists x \) friend\((x, self) \) => J’ai un ami

May need information you didn’t think about in your representation

Hard for semantic representations to cover everything

Everyone has a friend => \( \exists x \forall y \) friend\((x, y) \) => Tous a un ami

Can often get away without doing all disambiguation — same ambiguities may exist in both languages
Today: mostly phrase-based, some syntax
History of MT

- **1950**: Rule Base Machine Translation
  - Direct Machine Translation
  - Transfer Based RBMT
  - Interlingua Machine Translation

- **1966**: 1966-ALPAC Report
- **1968**: 1968-SYSTRAN
- **1980**: Example Based Machine Translation

- **1990**: Statistical Machine Translation
  - Word Based
  - Syntax Based
  - Phrase Based Model

- **2007**: Google Translate

- **2015**: RNN LSTM

- **2016**: GNMT
<table>
<thead>
<tr>
<th>ID</th>
<th>Text</th>
<th>ID</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Weekend traffic bans and traffic jams are a curse to road transport</td>
<td>7</td>
<td>I think this is an excellent principle and I would like to see it applied in full, but not to traffic jams.</td>
</tr>
<tr>
<td>6</td>
<td>Some people also want to recoup the cost of traffic jams from those who get stuck in them, according to the 'polluter pays' principle.</td>
<td>8</td>
<td>Traffic jams are indicative of failed government policy on the infrastructure front, which is why the government itself, certainly in the Netherlands, must be regarded as the polluter.</td>
</tr>
<tr>
<td>7</td>
<td>I think this is an excellent principle and I would like to see it applied in full, but not to traffic jams.</td>
<td>9</td>
<td>This would increase traffic jams, weaken road safety and increase costs.</td>
</tr>
<tr>
<td>8</td>
<td>In the previous legislature, Parliament gave its opinion on the Commission’s proposals on the simplification of vertical directives on sugar, honey, fruit juices, milk and jams.</td>
<td>10</td>
<td>For jams, I personally reintroduced an amendment that was not accepted by the Committee on the Environment, Public Health and Consumer Policy, but which I hold to.</td>
</tr>
<tr>
<td>9</td>
<td>In the previous legislature, Parliament gave its opinion on the Commission’s proposals on the simplification of vertical directives on sugar, honey, fruit juices, milk and jams.</td>
<td>11</td>
<td>It concerns not accepting the general use of a chemical flavouring in jams and marmalades, that is vanillin.</td>
</tr>
<tr>
<td>10</td>
<td>For jams, I personally reintroduced an amendment that was not accepted by the Committee on the Environment, Public Health and Consumer Policy, but which I hold to.</td>
<td>12</td>
<td>This is highlighted particularly in towns where it is necessary to find ways of solving environmental problems and the difficulties caused by traffic jams.</td>
</tr>
<tr>
<td>11</td>
<td>It concerns not accepting the general use of a chemical flavouring in jams and marmalades, that is vanillin.</td>
<td>13</td>
<td>This is highlighted particularly in towns where it is necessary to find ways of solving environmental problems and the difficulties caused by traffic jams.</td>
</tr>
</tbody>
</table>
Key idea: translation works better the bigger chunks you use

Remember phrases from training data, translate piece-by-piece and stitch those pieces together to translate

How to identify phrases? Word alignment over source-target bitext

How to stitch together? Language model over target language

Decoder takes phrases and a language model and searches over possible translations

NOT like standard discriminative models (take a bunch of translation pairs, learn a ton of parameters in an end-to-end way)
Phrase-Based MT

Unlabeled English data

Phrase table $P(f|e)$

Language model $P(e)$

Noisy channel model:
combine scores from
translation model +
language model to
translate foreign to
English

“Translate faithfully but make fluent English”
Evaluating MT

- Fluency: does it sound good in the target language?
- Fidelity/adequacy: does it capture the meaning of the original?
- BLEU score: geometric mean of 1-, 2-, 3-, and 4-gram precision vs. a reference, multiplied by brevity penalty

$$\text{BLEU} = BP \cdot \exp\left( \sum_{n=1}^{N} w_n \log p_n \right)$$

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>I am exhausted</th>
<th>1-gram</th>
<th>2-gram</th>
<th>3-gram</th>
</tr>
</thead>
<tbody>
<tr>
<td>hypothesis 1</td>
<td>___</td>
<td>3/3</td>
<td>1/2</td>
<td>0/1</td>
</tr>
<tr>
<td>hypothesis 2</td>
<td>Tired is I</td>
<td>1/3</td>
<td>0/2</td>
<td>0/1</td>
</tr>
<tr>
<td>hypothesis 3</td>
<td>III</td>
<td>1/3</td>
<td>0/2</td>
<td>0/1</td>
</tr>
<tr>
<td>reference 1</td>
<td>I am tired</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reference 2</td>
<td>I am ready to sleep now and so exhausted</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Evaluating MT

- Fluency: does it sound good in the target language?
- Fidelity/adequacy: does it capture the meaning of the original?
- BLEU score: geometric mean of 1-, 2-, 3-, and 4-gram precision vs. a reference, multiplied by brevity penalty

\[
\text{BLEU} = BP \cdot \exp \left( \sum_{n=1}^{N} w_n \log p_n \right)
\]

Typically \( n = 4, \ w_i = 1/4 \)

\[
BP = \begin{cases} 
1 & \text{if } c > r \\
\frac{1}{e^{(1-r/c)}} & \text{if } c \leq r 
\end{cases}
\]

\( r = \text{length of reference} \)
\( c = \text{length of system output} \)

- Does this capture fluency and adequacy?
Better methods with human-in-the-loop

HTER: human-assisted translation error rate

If you’re building real MT systems, you do user studies. In academia, you mostly use BLEU.
**Figure 3:** Screen shot of segment-rating portion of document-level direct assessment in the Appraise interface for an example English to German assessment from the human evaluation campaign. The annotator is presented with the machine translation output segment randomly selected from competing systems (anonymized) and is asked to rate the translation on a sliding scale.
Word Alignment
Word Alignment

- Input: a bitext, pairs of translated sentences
  - nous acceptons votre opinion . ||| we accept your view
  - nous allons changer d’avis ||| we are going to change our minds

- Output: alignments between words in each sentence
  - We will see how to turn these into phrases
  - “accept and acceptons are aligned”
1-to-Many Alignments

And₁ the₂ program₃ has₄ been₅ implemented₆

Le₁ programme₂ a₃ été₄ mis₅ en₆ application₇
Word Alignment

- Models $P(f|e)$: probability of “French” sentence being generated from “English” sentence according to a model.

- Latent variable model: $P(f|e) = \sum_a P(f, a|e) = \sum_a P(f|a, e)P(a)$

- Correct alignments should lead to higher-likelihood generations, so by optimizing this objective we will learn correct alignments.
IBM Model 1

- Each French word is aligned to at most one English word

\[
P(f, a | e) = \prod_{i=1}^{n} P(f_i | e_{a_i})P(a_i)
\]

e Thank you, I shall do so gladly.

f Gracias, lo hare de muy buen grado.

- Set P(a) uniformly (no prior over good alignments)

- \( P(f_i | e_{a_i}) \): word translation probability table

Brown et al. (1993)
HMM for Alignment

- Sequential dependence between a’s to capture monotonicity

\[ P(f, a|e) = \prod_{i=1}^{n} P(f_i|e_{a_i})P(a_i|a_{i-1}) \]

- Thank you, I shall do so gladly.

- Gracias, lo hare de muy buen grado.

- Alignment dist parameterized by jump size:

\[ P(a_j - a_{j-1}) \]

- \( P(f_i|e_{a_i}) \): same as before

Brown et al. (1993)
- Which direction is this?

- Alignments are generally monotonic (along diagonal)

- Some mistakes, especially when you have rare words (*garbage collection*)
Evaluating Word Alignment

- “Alignment error rate”: use labeled alignments on small corpus

<table>
<thead>
<tr>
<th>Model</th>
<th>AER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1 INT</td>
<td>19.5</td>
</tr>
<tr>
<td>HMM E→F</td>
<td>11.4</td>
</tr>
<tr>
<td>HMM F→E</td>
<td>10.8</td>
</tr>
<tr>
<td>HMM AND</td>
<td>7.1</td>
</tr>
<tr>
<td>HMM INT</td>
<td>4.7</td>
</tr>
<tr>
<td>GIZA M4 AND</td>
<td>6.9</td>
</tr>
</tbody>
</table>

- Run Model 1 in both directions and intersect “intelligently”
- Run HMM model in both directions and intersect “intelligently”
Phrase Extraction

- Find contiguous sets of aligned words in the two languages that don’t have alignments to other words
  
  d’assister à la reunion et ||| to attend the meeting and  
  
  assister à la reunion ||| attend the meeting  
  
  la reunion and ||| the meeting and  
  
  nous ||| we  
  
  ...  

- Lots of phrases possible, count across all sentences and score by frequency
Language Modeling
Phrase-Based MT

Phrase table $P(f|e)$

Unlabeled English data

Language model $P(e)$

Noisy channel model:
combine scores from
translation model +
language model to
translate foreign to
English

“Translate faithfully but make fluent English”
N-gram Language Models

I visited San _____ put a distribution over the next word

- Simple generative model: distribution of next word is a multinomial distribution conditioned on previous n-1 words

\[ P(x|\text{visited San}) = \frac{\text{count(visited San, } x\text{)}}{\text{count(visited San)}} \]

- Maximum likelihood estimate of this probability from a corpus

- Just relies on counts, even in 2008 could scale up to 1.3M word types, 4B n-grams (all 5-grams occurring >40 times on the Web)
Smoothing N-gram Language Models

- Smoothing is very important, particularly when using 4+ gram models.

\[ P(x|\text{visited San}) = (1 - \lambda) \frac{\text{count}(\text{visited San}, x)}{\text{count}(\text{visited San})} + \lambda \frac{\text{count}(\text{San}, x)}{\text{count}(\text{San})} \]

- One technique is “absolute discounting:” subtract off constant \( k \) from numerator, set lambda to make this normalize (\( k=1 \) is like leave-one-out).

\[ P(x|\text{visited San}) = \frac{\text{count}(\text{visited San}, x) - k}{\text{count}(\text{visited San})} + \lambda \frac{\text{count}(\text{San}, x)}{\text{count}(\text{San})} \]

- Kneser-Ney smoothing: this trick, plus low-order distributions modified to capture fertilities (how many distinct words appear in a context).
Engineering N-gram Models

- For 5+-gram models, need to store between 100M and 10B context-word-count triples.

- Make it fit in memory by *delta encoding* scheme: store deltas instead of values and use variable-length encoding.

Pauls and Klein (2011), Heafield (2011)
Neural Language Models

- Early work: feedforward neural networks looking at context

\[ P(w_i | w_{i-n}, \ldots, w_{i-1}) \]

- Variable length context with RNNs:
  - Works like a decoder with no encoder

- Slow to train over lots of data!

Mnih and Hinton (2003)
Evaluation

- (One sentence) negative log likelihood: \( \sum_{i=1}^{n} \log p(x_i|x_1, \ldots, x_{i-1}) \)

- Perplexity: \( 2^{-\frac{1}{n} \sum_{i=1}^{n} \log_2 p(x_i|x_1, \ldots, x_{i-1})} \)
  
  - NLL (base 2) averaged over the sentence, exponentiated
  
  - NLL = -2 -> on average, correct thing has prob 1/4 -> PPL = 4. PPL is sort of like branching factor
Results

- Evaluate on Penn Treebank: small dataset (1M words) compared to what’s used in MT, but common benchmark
- Kneser-Ney 5-gram model with cache: PPL = 125.7
- LSTM: PPL ~ 60-80 (depending on how much you optimize it)
- Melis et al.: many neural LM improvements from 2014-2017 are subsumed by just using the right regularization (right dropout settings). So LSTMs are pretty good

Merity et al. (2017), Melis et al. (2017)
Decoding
Phrase-Based Decoding

- Inputs:
  - Language model that scores $P(e_i|e_1, \ldots, e_{i-1}) \approx P(e_i|e_{i-n-1}, \ldots, e_{i-1})$
  - Phrase table: set of phrase pairs $(e, f)$ with probabilities $P(f|e)$

- What we want to find: $e$ produced by a series of phrase-by-phrase translations from an input $f$, possibly with reordering:

```
Morgen  fliege  ich  nach Kanada  zur Konferenz
```

```
Tomorrow  I  will fly  to the conference  in Canada
```
<table>
<thead>
<tr>
<th>the</th>
<th>7 people</th>
<th>including</th>
<th>by some</th>
<th>and</th>
<th>the russian</th>
<th>the</th>
<th>the astronauts</th>
</tr>
</thead>
<tbody>
<tr>
<td>it</td>
<td>7 people included</td>
<td>by france</td>
<td>and the</td>
<td>the russian</td>
<td>international astronomical</td>
<td>of rapporteur</td>
<td>.</td>
</tr>
<tr>
<td>this</td>
<td>7 out</td>
<td>including the</td>
<td>from</td>
<td>the french</td>
<td>and the russian</td>
<td>the fifth</td>
<td>.</td>
</tr>
<tr>
<td>these</td>
<td>7 among</td>
<td>including from</td>
<td>the french</td>
<td>and</td>
<td>of the russian</td>
<td>of space</td>
<td>members</td>
</tr>
<tr>
<td>that</td>
<td>7 persons</td>
<td>including from the</td>
<td>of france</td>
<td>and</td>
<td>to</td>
<td>russian</td>
<td>of the aerospace</td>
</tr>
<tr>
<td>7 include</td>
<td>from the</td>
<td>of france and</td>
<td>russian</td>
<td>astronauts</td>
<td></td>
<td>. the</td>
<td></td>
</tr>
<tr>
<td>7 numbers include</td>
<td>from france</td>
<td>and russian</td>
<td>of astronauts who</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 populations include</td>
<td>those from france</td>
<td>and russian</td>
<td>astronauts</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 deportees included</td>
<td>come from</td>
<td>france and russian</td>
<td>in</td>
<td>astronomical personnel</td>
<td>;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 philtrum</td>
<td>including those from</td>
<td>france and russian</td>
<td>a space</td>
<td>member</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>include</td>
<td>came from</td>
<td>france and russian</td>
<td>by cosmonauts</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>include</td>
<td>came from</td>
<td>french and russian</td>
<td>by cosmonauts</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>include</td>
<td>came from france</td>
<td>and russian</td>
<td>‘s cosmonauts</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>includes</td>
<td>coming from</td>
<td>french and</td>
<td>russian ‘s</td>
<td>cosmonaut</td>
<td>.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>french and russian</td>
<td>‘s</td>
<td>astronavigation</td>
<td>member</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>french</td>
<td>and russian</td>
<td>astronauts</td>
<td>special rapporteur</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and russian ‘s</td>
<td>and</td>
<td>russian</td>
<td>rapporteur</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>, and</td>
<td>russian</td>
<td>; rapporteur</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>, and</td>
<td>russian</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>, and</td>
<td>russian</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>or</td>
<td>russian ‘s</td>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Phrase-Based Decoding

- Input
  - lo haré rápidamente.

- Translations
  - I’ll do it quickly.
  - quickly I’ll do it.

- Decoding objective (for 3-gram LM)

\[
\arg \max_e [P(f|e) \cdot P(e)]
\]

\[
\arg \max_e \left[ \prod_{\langle e, f \rangle} P(f|e) \cdot \prod_{i=1}^{\left| e \right|} P(e_i|e_{i-1}, e_{i-2}) \right]
\]
If we translate with beam search, what state do we need to keep in the beam?

- What have we translated so far?
  \[ \arg \max_e \left[ \prod_{\langle \bar{e}, \bar{f} \rangle} P(f|\bar{e}) \cdot \prod_{i=1}^{\left| e \right|} P(e_i|e_{i-1}, e_{i-2}) \right] \]

- What words have we produced so far?

- When using a 3-gram LM, only need to remember the last 2 words!
Monotonic Translation

\[
\text{score} = \log [P(Mary) P(\text{not} \mid Mary) P(Mary \mid Maria) P(\text{not} \mid \text{no})]
\]

In reality: score = \( \alpha \log P(LM) + \beta \log P(TM) \)

...and TM is broken down into several features
**Monotonic Translation**

- Several paths can get us to this state, max over them (like Viterbi)
- Variable-length translation pieces = semi-HMM
Non-Monotonic Translation

- Non-monotonic translation: can visit source sentence “out of order”
- State needs to describe which words have been translated and which haven’t
- Big enough phrases already capture lots of reorderings, so this isn’t as important as you think

<table>
<thead>
<tr>
<th>Maria</th>
<th>no</th>
<th>dio</th>
<th>una</th>
<th>bofetada</th>
<th>a</th>
<th>la</th>
<th>bruja</th>
<th>verde</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mary</td>
<td>not</td>
<td>give</td>
<td>a</td>
<td>slap</td>
<td>to</td>
<td>the</td>
<td>witch</td>
<td>green</td>
</tr>
<tr>
<td>did not</td>
<td></td>
<td>a slap</td>
<td>by</td>
<td></td>
<td>to the</td>
<td></td>
<td>green witch</td>
<td></td>
</tr>
<tr>
<td>no</td>
<td></td>
<td>slap</td>
<td></td>
<td></td>
<td>to</td>
<td>the</td>
<td></td>
<td></td>
</tr>
<tr>
<td>did not give</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>slap</td>
<td></td>
<td></td>
<td>the</td>
<td></td>
<td>the witch</td>
<td></td>
</tr>
</tbody>
</table>

- translated: Maria, dio, una, bofetada
Training Decoders

score = $\alpha \log P(LM) + \beta \log P(TM)$

...and TM is broken down into several feature

- Usually 5-20 feature weights to set, want to optimize for BLEU score which is not differentiable

- MERT (Och 2003): decode to get 1000-best translations for each sentence in a small training set (<1000 sentences), do line search on parameters to directly optimize for BLEU
Moses

- Toolkit for machine translation due to Philipp Koehn + Hieu Hoang
  - Pharaoh (Koehn, 2004) is the decoder from Koehn’s thesis

- Moses implements word alignment, language models, and this decoder, plus *a ton* more stuff
  - Highly optimized and heavily engineered, could more or less build SOTA translation systems with this from 2007-2013

- Next week: results on these and comparisons to neural methods

http://www.statmt.org/moses/
Syntax
Rather than use phrases, use a *synchronous context-free grammar*

\[
\begin{align*}
\text{NP} & \to [\text{DT}_1 \ \text{JJ}_2 \ \text{NN}_3; \ \text{DT}_1 \ \text{NN}_3 \ \text{JJ}_2] \\
\text{DT} & \to [\text{the}, \ \text{la}] \\
\text{NN} & \to [\text{car}, \ \text{voiture}] \\
\text{JJ} & \to [\text{yellow}, \ \text{jaune}] \\
\end{align*}
\]

Translation = parse the input with “half” of the grammar, read off the other half

Assumes parallel syntax up to reordering
Syntactic MT

- Rather than use phrases, use a *synchronous context-free grammar*

<table>
<thead>
<tr>
<th>Urdu</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \rightarrow NP_1\ VP_2$</td>
<td>$NP_1\ VP_2$</td>
</tr>
<tr>
<td>$VP \rightarrow PP_1\ VP_2$</td>
<td>$VP_2\ PP_1$</td>
</tr>
<tr>
<td>$VP \rightarrow V_1\ AUX_2$</td>
<td>$AUX_2\ V_1$</td>
</tr>
<tr>
<td>$PP \rightarrow NP_1\ P_2$</td>
<td>$P_2\ NP_1$</td>
</tr>
<tr>
<td>$NP \rightarrow hamd ansary$</td>
<td>$Hamid Ansari$</td>
</tr>
<tr>
<td>$NP \rightarrow nab sdr$</td>
<td>$Vice President$</td>
</tr>
<tr>
<td>$V \rightarrow namzd$</td>
<td>nominated</td>
</tr>
<tr>
<td>$P \rightarrow kylye$</td>
<td>for</td>
</tr>
<tr>
<td>$AUX \rightarrow taa$</td>
<td>was</td>
</tr>
</tbody>
</table>
Syntactic MT

- Rather than use phrases, use a synchronous context-free grammar

<table>
<thead>
<tr>
<th>Urdu</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>S $\rightarrow$ NP① VP②</td>
<td>NP① VP②</td>
</tr>
<tr>
<td>VP $\rightarrow$ PP① VP②</td>
<td>VP① PP①</td>
</tr>
<tr>
<td>VP $\rightarrow$ V① AUX②</td>
<td>AUX② V①</td>
</tr>
<tr>
<td>PP $\rightarrow$ NP① P②</td>
<td>P① NP①</td>
</tr>
<tr>
<td>NP $\rightarrow$ hamd ansary</td>
<td>Hamid Ansari</td>
</tr>
<tr>
<td>NP $\rightarrow$ na}b sdr</td>
<td>Vice President</td>
</tr>
<tr>
<td>V $\rightarrow$ namzd</td>
<td>nominated</td>
</tr>
<tr>
<td>P $\rightarrow$ kylye</td>
<td>for</td>
</tr>
<tr>
<td>AUX $\rightarrow$ taa</td>
<td>was</td>
</tr>
</tbody>
</table>
Hamid Ansari was nominated for Vice President.
Hamid Ansari was nominated for Vice President.
Hamid Ansari was nominated for Vice President.
Hamid Ansari

Vice President

for

VP nominated was

was nominated

Hamid Ansari

for Vice President

VP
Hamid Ansari was nominated for Vice President.
Use lexicalized rules, look like “syntactic phrases”

Leads to HUGE grammars, parsing is slow

Grammar

\[ S \rightarrow \langle \text{VP} . ; \text{I VP} . \rangle \quad \text{OR} \quad S \rightarrow \langle \text{VP} . ; \text{you VP} . \rangle \]

\[ \text{VP} \rightarrow \langle \text{lo haré ADV} ; \text{will do it ADV} \rangle \]

\[ S \rightarrow \langle \text{lo haré ADV} . ; \text{I will do it ADV} . \rangle \]

\[ \text{ADV} \rightarrow \langle \text{de muy buen grado} ; \text{gladly} \rangle \]
Joshua

- Toolkit for syntactic machine translation due to many researchers at JHU (Weese, Ganitkevitch, Callison-Burch, Post, Lopez, ...)

- Joshua implements synchronized grammar extraction (Thrax!), parsing, language modeling, pruning, plus *a ton* more stuff

- Joshua uses two types of SCFG: Hiero grammar (Chiang, 2007), SAMT grammar (Zollmann & Venugopal, 2007)

https://cwiki.apache.org/confluence/display/JOSHUA/
Case Studies: Monolingual MT
Style Transfer

If you will not be turned, you will be destroyed!

If you will not be turn’d, you will be undone!

- Applied phrase-based MT (Moses Toolkit) to Shakespearean bitext

Slightly more fourth-graders nationwide are reading proficiently compared with a decade ago, but only a third of them are now reading well, according to a new report.

Most fourth-graders are better readers than they were 10 years ago. But few of them can actually read well.
Slightly more fourth-graders nationwide are reading proficiently compared with a decade ago, but only a third of them are now reading well, according to a new report.

Most fourth-graders are better readers than they were 10 years ago. But few of them can actually read well.
**Text Simplification**

**Feature Functions**
(Readability, language modeling, etc.)

**Objective Function**
(Xu et al., 2016)

\[
SARI = d_1 F_{add} + d_2 F_{keep} + d_3 P_{del}
\]

\[
p_{add}(n) = \frac{\sum_{g \in O} \min\left(\#_g(O \cap \overline{I}), \#_g(R)\right)}{\sum_{g \in O} \#_g(O \cap \overline{I})}
\]

\[
r_{add}(n) = \frac{\sum_{g \in O} \min\left(\#_g(O \cap \overline{I}), \#_g(R)\right)}{\sum_{g \in O} \#_g(R \cap \overline{I})}
\]

**Large-scale Paraphrases**
(lexical, phrasal, syntactic)

**Tuning Data**
(crowdsourced multi-references)

**Pairwise Ranking Optimization**

\( g(i, j) > g(i, j') \iff h_w(i, j) > h_w(i, j') \)

\( \iff h_w(i, j) - h_w(i, j') > 0 \)

\( \iff w \cdot x(i, j) - w \cdot x(i, j') > 0 \)

\( \iff w \cdot (x(i, j) - x(i, j')) > 0 \)

Implemented by modifying 4 major components of syntax-based MT (Joshua Toolkit); SARI is now part of tensor2tensor library.

Phrase-based systems consist of 3 pieces: aligner, language model, decoder

- HMMs work well for alignment
- N-gram language models are scalable and historically worked well
- Decoder requires searching through a complex state space

- Lots of system variants incorporating syntax
- Next week: neural MT