Overview of Rest of Class

- **Today** - Phrase Based Machine Translation
- **Oct 24, 31** - Parsing (Guest Lecturers: Emily Pitler and Slav Petrov)
- **Nov 7** - Deep Learning for NLP
- **Nov 14** - Guest Lecture (Sam Bowman) - *no office hours*
- **Nov 21** - Adv Machine Translation (i.e. Neural MT)
- **Nov 28** - Compositional Semantics (Guest Lecturer: Tom Kwiatkowski)
- **Dec 5** - Summarization / Other Advanced Topics / Conclusion
Machine Translation

- Translate text from one language to another
- Recombines fragments of example translations
- Challenges:
  - What fragments? [learning to translate]
  - How to make efficient? [fast translation search]
  - Fluency vs fidelity
Corpus-Based MT

Modeling correspondences between languages

Sentence-aligned parallel corpus:

Yo lo haré mañana
I will do it tomorrow

Hasta pronto
See you soon

Hasta pronto
See you around

Machine translation system:

Yo lo haré pronto
I will do it soon

Novel Sentence
I will do it around

See you tomorrow
Levels of Transfer

interlingua

semantics

syntax

phrases

words

SOURCE

TARGET

P( | ) = 0.8

| English (E) | P(E | lo haré) |
|-------------|--------------|
| will do it  | 0.8          |
| will do so  | 0.2          |

| English (E) | P(E | mañana) |
|-------------|------------|
| tomorrow    | 0.7        |
| morning     | 0.3        |
MT System Components

Language Model

source
P(e)

best

e

decoder

Translation Model

channel
P(f|e)

observed

f

\[
e^* = \arg\max_e P(e|f) = \arg\max_e P(f|e)P(e)
\]
**Language Model**

- **Source**
  - $P(e)$

- **Decoder**
  - $e^* = \arg\max_e P(e|f) = \arg\max_e P(f|e)P(e)$

**Translation Model**

- **Channel**
  - $P(f|e)$

- **Observed**
  - $f$

- **Best**
  - $e$

---

**Equation**

$$e^* = \arg\max_e P(e|f) = \arg\max_e P(f|e)P(e)$$
MT: Evaluation

- Human evaluations: subject measures, fluency/adequacy

- Automatic measures: n-gram match to references
  - NIST measure: n-gram recall (worked poorly)
  - BLEU: n-gram precision (no one really likes it, but everyone uses it)

- BLEU:
  - P1 = unigram precision
  - P2, P3, P4 = bi-, tri-, 4-gram precision
  - Weighted geometric mean of P1-4
  - Brevity penalty (why?)
  - Somewhat hard to game…
Why Automatic Evaluation is Hard

*Exact reconstruction* (24 of 38)

Please give me your response as soon as possible.

⇒ Please give me your response as soon as possible.

*Reconstruction preserving meaning* (8 of 38)

Now let me mention some of the disadvantages.

⇒ Let me mention some of the disadvantages now.

*Garbage reconstruction* (6 of 38)

In our organization research has two missions.

⇒ In our missions research organization has two.
Candidate:  the the the the the the the the
Reference 1:  The cat is on the mat.
Reference 2:  There is a cat on the mat.

vanilla unigram precision = 1.0  (yikes!!)
BLEU [Papineni et al. 2002]

Use a modified unigram precision instead by clipping counts. Let $x_w$ = maximum times number of times a word appears in any given translation.

**Candidate:**

the the the the the the the the

**Reference 1:**

The cat is on the mat.

**Reference 2:**

There is a cat on the mat.

$x_{\text{the}} = 2$

modified unigram precision = $\frac{\min(7, 2)}{7} = \frac{2}{7}$
Can then combine $n$-grams and weighted precisions

$\exp\left(\sum_n w_n p_n\right)$

- weight
- (modified) precision for given $n$-gram size

What is the problem with this?
So far we only have a precision-based measure e.g. consider the problem case below:

**Candidate:**

**Reference 1:**  
It is a guide to action that ensures that the military will forever heed Party commands.

**Reference 2:**  
It is the guiding principle which guarantees the military forces always being under the command of the Party.

**Reference 3:**  
It is the practical guide for the army always to heed the directions of the party.

Modified unigram/bigram precisions are both 1.0
Typically, recall is used to solve this problem, but recall doesn’t work well with multi-reference.

Candidate 1:  
I always do.

Candidate 2:  
I always invariably perpetually do.

Reference 1:  
I always do.

Reference 2:  
I invariably do.

Reference 3:  
I perpetually do.

Candidate 2 will get a high recall, even if it is incorrect.
BLEU [Papineni et al. 2002]

The proposed solution in BLEU is to use a brevity penalty (BP), encouraging the candidate translation to have length similar to one of the reference translations.

- \( c \) = candidate translation length
- \( r \) = effective reference translation length

\[
BP = \begin{cases} 
1 & c > r \\
\exp(1 - \frac{r}{c}) & c \leq r 
\end{cases}
\]

\[
BLEU = BP \times \exp(\sum_n w_n p_n)
\]
Automatic Metrics Work (?)
Translation with IBM Models

\[ e^* = \arg\max_e P(e|f) = \arg\max_e P(f|e)P(e) \]

Let us first pretend the translation model \( P(f|e) \) is IBM Model 2.

Two challenges:
- Computing the marginal probability \( P(f \mid e) \)
- Taking the argmax over all possible \( e \)
Translation with IBM Models

\[ e^* = \arg\max_e P(e|f) = \arg\max_e P(f|e)P(e) \]

Let us first pretend the translation model \( P(f|e) \) is IBM Model 2.

Two challenges:
- Computing the marginal probability \( P(f \mid e) \)
- Taking the argmax over all possible \( e \)
IBM Models: Computing $P(f|e)$

\[
P(f|e) = \sum_{a_1,\ldots,a_m} P(f_1, \ldots, f_m, a_1, \ldots, a_m|e_1, \ldots, e_l, m)
\]

\[
= \sum_{a_1,\ldots,a_m} \prod_{i=1}^{m} q(a_i|i, l, m) t(f_i|e_{a_i})
\]

*Intractable in general*
IBM Model 1: Computing $P(f|e)$

IBM Model 1 is special and it is possible to exactly compute the marginal probability exactly

$$
P(f|e) = \sum_{a_1,\ldots,a_m} \prod_{i=1}^{m} \frac{1}{\ell + 1} t(f_i|e_{a_i})
$$

$$
\prod_{i=1}^{m} \frac{1}{\ell + 1} \sum_{a_i} t(f_i|e_{a_i})
$$
IBM Model 2: Computing $P(f|e)$

One typical heuristic is to replace the sum over all possible alignments with just the top $Z$ most likely alignments.

$$P(f|e) = \sum_{a_1, \ldots, a_m} P(f_1, \ldots, f_m, a_1, \ldots, a_m|e_1, \ldots, e_l, m)$$

$$= \sum_{a_1, \ldots, a_m} \prod_{i=1}^{m} q(a_i|i, l, m)t(f_i|e_{a_i})$$

$$\approx \sum_{z=1}^{Z} \prod_{i=1}^{m} q(a_i^{\tilde{z}}|i, l, m)t(f_i|e_{a_i^{\tilde{z}}})$$

$(a_1^{\tilde{z}}, \ldots, a_m^{\tilde{z}})$ is the $z^{th}$ highest scoring alignment

In which situations would we expect this to work well?
Computing the most likely alignment is tractable for IBM Model 2

\[
\begin{align*}
a_1^*, \ldots, a_m^* &= \arg\max_{a_1, \ldots, a_m} P(f_1, \ldots, f_m, a_1, \ldots, a_m | e_1, \ldots, e_l, m) \\
&= \arg\max_{a_1, \ldots, a_m} \prod_{i=1}^{m} q(a_i | i, l, m) t(f_i | e_{a_i}) \\
\end{align*}
\]

\[
\begin{align*}
a_i^* &= \arg\max_{a_i} (q(a_i | i, l, m) t(f_i | e_{a_i}))
\end{align*}
\]
Decoding

\[ e^* = \arg\max_e P(e|f) = \arg\max_e P(f|e)P(e) \]

Let us first pretend the translation model \( P(f|e) \) is IBM Model 2.

Two challenges:
- Computing the marginal probability \( P(f | e) \)
- Taking the \( \arg\max \) over all possible \( e \)

Number of possibilities is huge!!
- NP-hard, reduces to Traveling Salesman Problem (if \( P(e) \) is a bigram language model)
- Even if we knew the set of words in the optimal \( e \), still \( m! \) number of permutations.

We will discuss strategies for this problem later in the lecture.
No one actually uses IBM models as the translation model.

Instead we use the alignments extracted from the IBM models to discover phrases and do translation at the phrase level.
Phrase-Based Systems

Sentence-aligned corpus

Word alignments

Phrase table (translation model)

Morgen fliege ich nach Kanada zur Konferenz

Tomorrow I will fly to the conference in Canada

cat ||| chat ||| 0.9
the cat ||| le chat ||| 0.8
dog ||| chien ||| 0.8
house ||| maison ||| 0.6
my house ||| ma maison ||| 0.9
language ||| langue ||| 0.9
...

Phrase table
A phrase lexicon is a set of tuples. Phrases do not have to match in length.

\[(f, e, g)\]

- (au, to the, 0.5)
- (au banque, to the bank, 0.01)
- (allez au banque, go to the bank, -2.5)
Constructing a Phrase Lexicon

Assume an alignment matrix $A^{(k)}$ is given for each example $k$.

$$A^{(k)}_{i,j} = 1 \text{ if Foreign word } i \text{ is aligned to English word } j, \ 0 \text{ otherwise}$$

We can use IBM models (in combination with some heuristics) for example, to construct such a matrix.
**Bidirectional Alignment**

### English to Spanish

<table>
<thead>
<tr>
<th>english</th>
<th>Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>Maria no daba una</td>
</tr>
<tr>
<td>did</td>
<td>a la</td>
</tr>
<tr>
<td>not</td>
<td>bruja</td>
</tr>
<tr>
<td>slap</td>
<td>verde</td>
</tr>
<tr>
<td>the</td>
<td></td>
</tr>
<tr>
<td>green</td>
<td></td>
</tr>
<tr>
<td>witch</td>
<td></td>
</tr>
</tbody>
</table>

### Spanish to English

<table>
<thead>
<tr>
<th>spanish</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maria no daba una</td>
<td>Mary</td>
</tr>
<tr>
<td>a la</td>
<td>did</td>
</tr>
<tr>
<td>bruja</td>
<td>not</td>
</tr>
<tr>
<td>verde</td>
<td>slap</td>
</tr>
<tr>
<td></td>
<td>the</td>
</tr>
<tr>
<td></td>
<td>green</td>
</tr>
<tr>
<td></td>
<td>witch</td>
</tr>
</tbody>
</table>

### Intersection

<table>
<thead>
<tr>
<th>intersection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maria no daba una</td>
</tr>
<tr>
<td>a la</td>
</tr>
<tr>
<td>bruja</td>
</tr>
<tr>
<td>verde</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Constructing a Phrase Lexicon

- Once we have an alignment matrix, the next step is to use it to extract phrases / scores.

- We iterate over all training examples $k$
  - For each phrase pairs $(s, t)$ from the source and $(s', t')$ from the target add it to our phrase lexicon if it is consistent with the alignment matrix.
Constructing a Phrase Lexicon

$nous\ devons\ aussi\ prendre\ ces\ critiques\ au\ sérieux$\quad f$

$we\ must\ also\ take\ these\ criticisms\ seriously$\quad e$

$O(m^4)$ possible phrase pairs (but we will enumerate sparsely)

$(s,t) = (1, 2), (s', t') = (2, 5)$ corresponds to:

$(nous\ devons,\ must\ also\ take\ these)$

Add these phrases to the phrase lexicon if they are consistent with A
Consistent Phrases

- All words of the phrase have to align to each other (or to nothing)
Extracting Phrases

(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green), (Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the), (bruja verde, green witch),
(Maria no daba una bofetada, Mary did not slap),
(no daba una bofetada a la, did not slap the), (a la bruja verde, the green witch),
(Maria no daba una bofetada a la, Mary did not slap the),
(daba una bofetada a la bruja verde, slap the green witch),
(no daba una bofetada a la bruja verde, did not slap the green witch),
(Maria no daba una bofetada a la bruja verde, Mary did not slap the green witch)
Phrase Weights

How the MT community estimates $P(f|\bar{e})$

Parallel training sentences provide phrase pair counts.

Gracias, lo haré de muy buen grado. Thank you, I shall do so gladly.

lo haré $\iff$ I shall do so 44 times in the corpus

All phrase pairs are counted, and counts are normalized.

Gracias lo haré de muy buen grado.

Thank you I shall do so gladly.

$$P(f|\bar{e}) = \frac{\text{count}(f, \bar{e})}{\text{count}(\bar{e})}$$
Phrase Scoring

\[
\phi(\bar{e}_j | \bar{f}_i) = \frac{c(\bar{f}_i, \bar{e}_j)}{c(\bar{f}_i)}
\]

\[
g(\bar{e}_j, \bar{f}_i) = \log \phi(\bar{e}_j | \bar{f}_i)
\]

- Learning weights has been tried, several times:
  - [Marcu and Wong, 02]
  - [DeNero et al, 06]
  - … and others

- Seems not to work well, for a variety of partially understood reasons

- Main issue: big chunks get all the weight, obvious priors don’t help
  - Though, [DeNero et al 08]
Phrase Table Example

- Phrase translations for ‘der Vorschlag’:

| English               | $\phi(\bar{e}|f)$ | English               | $\phi(\bar{e}|f)$ |
|-----------------------|-------------------|-----------------------|-------------------|
| the proposal          | 0.6227            | the suggestions       | 0.0114            |
| ’s proposal           | 0.1068            | the proposed          | 0.0114            |
| a proposal            | 0.0341            | the motion            | 0.0091            |
| the idea              | 0.0250            | the idea of           | 0.0091            |
| this proposal         | 0.0227            | the proposal          | 0.0068            |
| proposal              | 0.0205            | its proposal          | 0.0068            |
| of the proposal       | 0.0159            | it                    | 0.0068            |
| the proposals         | 0.0159            | ...                   | ...               |

- Lexical variation, morphology, function words
Derivations

- A derivation $y$ is a finite sequence of phrases from the phrase table.

- Denote each phrase with the short-hand:

  $$ p = (s, t, e) $$

  - Start index of source phrase - denote as $s(p)$
  - End index of source phrase - denote as $t(p)$
  - Target phrase
Derivations

Example phrases in this notation:

- $p_1 = (1, 3, \text{we must also})$
- $p_2 = (7, 7, \text{take})$
- $p_3 = (4, 5, \text{this criticism})$
- $p_4 = (6, 6, \text{seriously})$

Derivation:

$y = p_1 \ p_2 \ p_3 \ p_4$

$y = (1, 3, \text{we must also}) \ (7, 7, \text{take}) \ (4, 5, \text{this criticism}) \ (6, 6, \text{seriously})$
Constraints on the Derivation

- Each phrase in the derivation must come from the phrase table.

- Each source word is translated exactly one time.

- A distortion limit that limits how far consecutive phrases can be found from each other.
Distortion Limit

- Distortion limit $d$, which limits how far consecutive phrases can be found from each other.

\[
|1 - s(p_1)| \leq d
\]

\[
|t(p_2) + 1 - s(p_3)| \leq d
\]

- In addition to improving performance, helps make decoding more efficient.
- Similar intuition to HMM alignment model
Scoring the Derivation

\[ y^* = \arg\max_y f(y) \]

\[ f(y) = h(e(y)) + \sum_{k=1}^{L} g(p_k) + \sum_{k=1}^{L-1} \eta \times |t(p_k) + 1 - s(p_{k+1})| \]

- target language model \( \log p(e) \)
- translation model \( \log p(f \mid e) \)
- extra distortion penalty (in addition to hard constraint)
Scoring the Derivation

Log form

\[ h(e(y)) + \sum_{k=1}^{L} g(p_k) + \sum_{k=1}^{L-1} \eta \times |t(p_k) + 1 - s(p_{k+1})| \]

Exponential form

\[ \exp(h(e(y))) \prod_{k=1}^{L} \exp(g(p_k)) \prod_{k=1}^{L-1} (\exp(|t(p_k) + 1 - s(p_{k+1})|))^\eta \]

\[ P_{lm}(e(y)) \prod_{k=1}^{L} \phi(\bar{f}_k|\bar{e}_k) \prod_{k=1}^{L-1} (\exp(|t(p_k) + 1 - s(p_{k+1})|))^\eta \]

target language model \( p(e) \)

translation model \( p(fle) \)
Decoding

- Define a state:

\( (e_{inc}, b, r, \alpha) \)

- Example:

\[ y = (1, 3, \text{we must also}) (7, 7, \text{take}) (4, 5, \text{this criticism}) \]

(we must also take this criticism, 111101, 5, \( \alpha \))
Decoding

- Initialize the state to:

  $$(\cdot, 0000000, 0, 0)$$

- When considering the next step of the translation consider all possible phrases that could append to the above state.
  - Subject to constraints

- Keep the K-best incremental translations so far (beam search)
Decoding
Results

<table>
<thead>
<tr>
<th>Language Pair</th>
<th>Model4</th>
<th>Phrase</th>
<th>Lex</th>
</tr>
</thead>
<tbody>
<tr>
<td>English-German</td>
<td>0.2040</td>
<td>0.2361</td>
<td>0.2449</td>
</tr>
<tr>
<td>French-English</td>
<td>0.2787</td>
<td>0.3294</td>
<td>0.3389</td>
</tr>
<tr>
<td>English-French</td>
<td>0.2555</td>
<td>0.3145</td>
<td>0.3247</td>
</tr>
<tr>
<td>Finnish-English</td>
<td>0.2178</td>
<td>0.2742</td>
<td>0.2806</td>
</tr>
<tr>
<td>Swedish-English</td>
<td>0.3137</td>
<td>0.3459</td>
<td>0.3554</td>
</tr>
<tr>
<td>Chinese-English</td>
<td>0.1190</td>
<td>0.1395</td>
<td>0.1418</td>
</tr>
</tbody>
</table>

Koehn et al. 2003
## Phrase Translation Tables

<table>
<thead>
<tr>
<th>Max. Length</th>
<th>Training corpus size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10k</td>
</tr>
<tr>
<td>2</td>
<td>37k</td>
</tr>
<tr>
<td>3</td>
<td>63k</td>
</tr>
<tr>
<td>4</td>
<td>84k</td>
</tr>
<tr>
<td>5</td>
<td>101k</td>
</tr>
<tr>
<td>7</td>
<td>130k</td>
</tr>
</tbody>
</table>

Table 2: Size of the phrase translation table with varying maximum phrase length limits
Phrase Size

- Phrases do help
  - But they don’t need to be long
  - Why should this be?

Koehn et al. 2003
Sources of Alignments

<table>
<thead>
<tr>
<th>Method</th>
<th>Training corpus size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10k</td>
</tr>
<tr>
<td>AP</td>
<td>84k</td>
</tr>
<tr>
<td>Joint</td>
<td>125k</td>
</tr>
<tr>
<td>Syn</td>
<td>19k</td>
</tr>
</tbody>
</table>
Alignment Heuristics
Lexical Weighting

\[ \phi(f_i | e_i) = \frac{\text{count}(f_i, e_i)}{\text{count}(e_i)} p_w(f_i | e_i) \]

\[ p_w(f | e, a) = p_w(f_1 f_2 f_3 | e_1 e_2 e_3, a) = w(f_1 | e_1) \]
\[ \times \frac{1}{2} (w(f_2 | e_2) + w(f_2 | e_3)) \]
\[ \times w(f_3 | \text{NULL}) \]
Summary: Phrase-Based Systems

Sentence-aligned corpus

Word alignments

Phrase table (translation model)

- Morgen ||| fliege ||| ich ||| nach Kanada ||| zur Konferenz
  - Tomorrow ||| I will fly ||| to the conference ||| in Canada

- cat ||| chat ||| 0.9
- the cat ||| le chat ||| 0.8
- dog ||| chien ||| 0.8
- house ||| maison ||| 0.6
- my house ||| ma maison ||| 0.9
- language ||| langue ||| 0.9
- ...
“Tuning”

More generally, define feature functions $h$:

$$p(e|f) \propto \exp \sum_{k=1}^{m} \lambda_k \cdot h_k(e, f)$$

- $P(e)$ and $P(f|e)$ are just two of many possible feature functions:
  - $P(e|f)$, word counts, phrase counts, etc.
- Need to learn how to set weights $\lambda$
  - What function to maximize?
  - How to maximize? Is it differentiable?
Minimum Error Rate Training [Och ’03]

- Non-convex, non-differentiable objective:
  - Generate n-best list
  - Line search 1 dir. at a time
  - Use random restarts

- Each hypothesis in n-best list contributes a line:

![Diagram of training process involving model, generate n-best list, score translations, find feature weights, and change feature weights.]

[Och '03]
Machine Translation Preordering

- English has Subject-Verb-Object word order, while Japanese has Subject-Object-Verb order
- Use hand-written or automatically learned rules to change word order prior to translation \[\text{[Collins et al. '05]}\]
- Dependency-based reordering for English-Japanese \[\text{[Xu et al. '09]}\]
Machine Translation Reordering

- Source-side reordering for machine translation
  Use source-side syntax to guide reordering decisions
- Dependency-based reordering for English-Japanese

[Collins et al. ’05]
[Collins et al. ’05]

Sie kann Pasta jeden Tag essen.

[Collins et al. ’05]
[Collins et al. ’05]

Sie glaubt dass sie Pasta jeden Tag essen kann.

[Collins et al. ’05]
[Collins et al. ’05]
Classifier Reordering Results

[Lerner & Petrov ’13]

**BLEU Improvement**

<table>
<thead>
<tr>
<th>to:</th>
<th>Arabic</th>
<th>Hebrew</th>
<th>Indonesian</th>
<th>Irish</th>
<th>Malay</th>
<th>Welsh</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU</td>
<td>0.9</td>
<td>0.9</td>
<td>2.4</td>
<td>1.5</td>
<td>2.1</td>
<td>1.8</td>
</tr>
<tr>
<td>Improvement</td>
<td>1.1</td>
<td>1.4</td>
<td>3.0</td>
<td>2.5</td>
<td>2.5</td>
<td>3.1</td>
</tr>
</tbody>
</table>

**Japanese**

<table>
<thead>
<tr>
<th></th>
<th>rule-based reordering</th>
<th>1-step classifier reordering</th>
<th>2-step classifier reordering</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU</td>
<td>6.0</td>
<td>7.1</td>
<td>7.6</td>
</tr>
<tr>
<td>Improvement</td>
<td>3.1</td>
<td>3.3</td>
<td>3.7</td>
</tr>
</tbody>
</table>

**Japanese**

<table>
<thead>
<tr>
<th></th>
<th>rule-based reordering</th>
<th>1-step classifier reordering</th>
<th>2-step classifier reordering</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU</td>
<td>7.1</td>
<td>7.6</td>
<td>8.6</td>
</tr>
<tr>
<td>Improvement</td>
<td>6.0</td>
<td>6.5</td>
<td>7.7</td>
</tr>
</tbody>
</table>

**Korean**

<table>
<thead>
<tr>
<th></th>
<th>rule-based reordering</th>
<th>1-step classifier reordering</th>
<th>2-step classifier reordering</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU</td>
<td>7.7</td>
<td>7.7</td>
<td>8.6</td>
</tr>
<tr>
<td>Improvement</td>
<td>3.1</td>
<td>3.1</td>
<td>4.6</td>
</tr>
</tbody>
</table>
Dutch Parser in Machine Translation

- No preordering (Lerner and Petrov, 2013)
- Arc-eager preordering (L&P, 2013)
- Arc-eager preordering (updated baseline)
- Two-Registers preordering (updated baseline)

Dutch-English BLEU scores:

- No preordering: 34.9
- Arc-eager preordering: 35.1
- Arc-eager preordering (updated baseline): 35.4
- Two-Registers preordering (updated baseline): 35.9