Topics: One simple approach to the problem of learning a translation dictionary; this approach illustrates a general method for learning when hidden structure is involved, and echoes an approach we’ve considered in a different context earlier in the course.

I. Example translation pair

Un program a été mis en application
vs.
And a program has been implemented

II. Example alignments

Here is a graphical depiction of two out of the 120 possible alignments\(^1\) for the sentence pair “la maison bleue est bleue vs. the blue house is blue”, where the French sentence is the source.

Formally, we would denote \([A1]\) by \((1 \leftrightarrow 1, 2 \leftrightarrow 3, 3 \leftrightarrow 2, 4 \leftrightarrow 4, 5 \leftrightarrow 5)\).

III. Notation

- For a sentence pair \(p\), let \(\text{Aligns}(p)\) be the set of all possible alignments of the two sentences in \(p\), and let \(\text{NumAligns}(p)\) be the size of this set.

- Let \(\text{freq}(s \leftrightarrow t, A)\) be the number of times we have the source word \(s\) “matched” to the target word \(t\) in alignment \(A\). In our example above, we have \(\text{freq}(\text{bleue} \leftrightarrow \text{blue}, [A1]) = 2\).

\(^1\)There are only 120 because we only consider “one-to-one and onto” alignments.
IV. An iterative learning algorithm for MT  
Inspired by IBM’s Candide system from the 80s and 90s.

1. Initialization: For every sentence pair \( p \), for every alignment \( A \) of \( p \), set \( \text{Awt}^{(0)}(A) = 1/(\text{NumAligns}(p)) \).

   Let \( i \) be increasing from 1 on, until the translation weights “converge”:

2. Compute temporary translation weights:
   For every source/target word pair \( (s, t) \), set \( \text{TempTr}(s \rightarrow t) = \sum_{A} \text{freq}(s \leftrightarrow t, A) \text{Awt}^{(i-1)}(A) \).

3. Get the translation weights by sum-normalizing the temporary ones:
   For each source word \( s \), compute \( \text{norm}_s = \sum_{t'} \text{TempTr}(s \rightarrow t') \);
   then, set each \( \text{Tr}^{(i)}(s \rightarrow t) = \text{TempTr}(s \rightarrow t)/\text{norm}_s \).

4. Compute temporary alignment weights: For every alignment \( A = (1 \leftrightarrow a(1); 2 \leftrightarrow a(2); \cdots; \ell \leftrightarrow a(\ell)) \),
   set \( \text{TempAwt}(A) = \text{Tr}^{(i)}(s_1 \rightarrow t_{a(1)}) \times \text{Tr}^{(i)}(s_2 \rightarrow t_{a(2)}) \times \cdots \times \text{Tr}^{(i)}(s_\ell \rightarrow t_{a(\ell)}) \).

5. Get the alignment weights by sum-normalizing the temporary ones: For each pair \( p \),
   compute \( \text{norm}_p = \sum_{A' \in \text{Aligns}(p)} \text{TempAwt}(A') \);
   then, for every \( A \) in \( \text{Aligns}(p) \), set \( \text{Awt}^{(i)}(A) = \text{TempAwt}(A)/\text{norm}_p \).

V. Example\(^2\) partial execution

Suppose we have two sentence pairs, \( p_1 = \text{“chat bleu vs. blue cat”} \) and \( p_2 = \text{“chat vs. cat”} \). This yields three alignments:

\[
A_1 = (1 \leftrightarrow 1; 2 \leftrightarrow 2) \quad \text{(so chat aligned to blue in \( p_1 \))} \\
A_1' = (1 \leftrightarrow 2; 2 \leftrightarrow 1) \quad \text{(so chat aligned to cat in \( p_1 \))} \\
A_2 = (1 \leftrightarrow 1) \quad \text{(only one possible choice)}
\]

<table>
<thead>
<tr>
<th>Alignment weights</th>
<th>( A_1 )</th>
<th>( A_1' )</th>
<th>( A_2 )</th>
<th>chat → blue</th>
<th>chat → cat</th>
<th>bleu → blue</th>
<th>bleu → cat</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Init</td>
<td>1/2</td>
<td>1/2</td>
<td>1</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>b. TempTr’s</td>
<td>“”</td>
<td>“”</td>
<td>“”</td>
<td>1/2</td>
<td>3/2</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(insert normalization computations here)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c. Tr’s</td>
<td>“”</td>
<td>“”</td>
<td>“”</td>
<td>1/4</td>
<td>3/4</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(insert normalization computations here)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d. TempAwt’s</td>
<td>1/8</td>
<td>3/8</td>
<td>3/4</td>
<td>“”</td>
<td>“”</td>
<td>“”</td>
<td>“”</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(insert normalization computations here)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e. Awt’s</td>
<td>1/4</td>
<td>3/4</td>
<td>1</td>
<td>“”</td>
<td>“”</td>
<td>“”</td>
<td>“”</td>
</tr>
<tr>
<td>f. TempTr’s</td>
<td>“”</td>
<td>“”</td>
<td>“”</td>
<td>1/4</td>
<td>7/4</td>
<td>3/4</td>
<td>1/4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(insert normalization computations here)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g. Tr’s</td>
<td>“”</td>
<td>“”</td>
<td>“”</td>
<td>1/8</td>
<td>7/8</td>
<td>3/4</td>
<td>1/4</td>
</tr>
</tbody>
</table>

\(^2\)Adapted from Sections 26 (“Chicken and egg”) and 27 (“Now for the Magic”) of Kevin Knight’s (1999) *A Statistical MT Tutorial Workbook* (http://www.isi.edu/natural-language/mt/wkbk.rtf). The tutorial also discusses more advanced models, and is often fairly amusing to boot.