Computation, Information, and Intelligence (COMS/ENGRI/INFO/COGST 172), Fall 2005
11/2/05: **Lecture 29 aid — Statistical machine translation**

Agenda: "Blank-slate" learning of correspondence tables (word-for-word dictionaries). Note: while the notation below may seem a bit complicated, the underlying ideas are intuitive.

Announcements: We will be using the same prelim seating arrangement as before. That is, to make it easier for the course staff to answer individual questions at a minimum of disturbance to other students, we ask that you sit as much as possible in *alternate* rows, starting with the row closest to the front. We would like to have a maximum of two people in the "no-mans-land" rows in between, sitting as much in the middle of the row as possible.

I. Data This consists of a set of mutual-translation sentence pairs. A realistic example would be

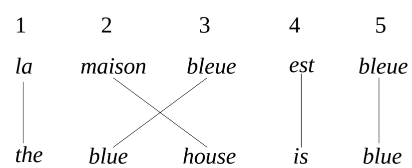
Un program a été mis en application
vs.
And a program has been implemented

However, for simplicity, we will assume that each of the two sentences within a given pair have the same number of words in them, although the sentences within a given pair can have a different length than those in another pair.

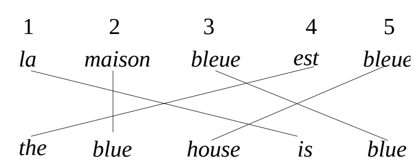
II. Alignments The idea is to treat the target sentence as basically just a reordering of the source sentence.

Let "s vs. t" be a sentence pair, and let $n \geq 1$ be the length of sentence s. Let $s = s_1 s_2 \cdots s_n$, where each s_i is a word (repeats allowed), and similarly let $t = t_1 t_2 \cdots t_n$. An alignment specifies for each position i in the source sentence a position $a(i)$ in the target sentence, where the sentence-pair-specific function a is one-to-one and onto (thus, each source position gets matched to exactly one target position and vice versa). We write this formally as $(1 \leftrightarrow a(1), 2 \leftrightarrow a(2), \ldots, n \leftrightarrow a(n))$. Under our (restricted) definition of alignment, there are $n! = n \times (n-1) \times (n-2) \times \cdots 2 \times 1$ alignments for a sentence pair in which each of the two component sentences has length n.

III. Example alignments Here is a graphical depiction of two out of the 120 possible alignments for the sentence pair "*la maison bleue est bleue* vs. *the blue house is blue*".



[A1]



[A2]

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

IV. 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 $\text{Contains}(s \leftrightarrow t)$ be the set of all alignments A (across all sentence pairs) that contain a position match $i \leftrightarrow j$ where the i^{th} source word was s and the j^{th} target word was t. In the example above, alignment [A1] is in $\text{Contains}(\textit{maison} \leftrightarrow \textit{house})$ but [A2] isn't.
- 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 $\text{freq}(\textit{bleue} \leftrightarrow \textit{blue}, [A1]) = 2$.

V. 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 $\text{awt}(A) = 1/(\text{NumAligns}(p))$.

2. Repeat the following steps in order until no "significant" change:

3. Update translation weights: For every source/target word pair (s,t), change $\text{tr}(s \rightarrow t)$ to $\sum_{A \text{ in Contains}(s\leftrightarrow t)} \text{freq}(s \leftrightarrow t, A)\text{awt}(A)$.

4. Sum-normalize translation weights: for each source word s, compute $\text{norm}_s = \sum_{t'} \text{tr}(s \rightarrow t')$; then, change each $\text{tr}(s \rightarrow t)$ to $\text{tr}(s \rightarrow t)/\text{norm}_s$.

5. Update alignment weights: For every alignment $A = (1 \leftrightarrow a(1); 2 \leftrightarrow a(2); \cdots; \ell \leftrightarrow a(\ell))$, change $\text{awt}(A)$ to $\text{tr}(s_1 \rightarrow t_{a(1)}) \times \text{tr}(s_2 \rightarrow t_{a(2)}) \cdots \times \text{tr}(s_\ell \rightarrow t_{a(\ell)})$ (note that ℓ can be different for different A).

6. Sum-normalize alignment weights: For each pair p, compute $\text{norm}_p = \sum_{A' \in \text{Aligns}(p)} \text{awt}(A')$; then, for every A in $\text{Aligns}(p)$, change $\text{awt}(A)$ to $\text{awt}(A)/\text{norm}_p$.

Note that translation weights are normalized across all the data, whereas alignment weights are normalized with respect to a given sentence pair.

VI. Example partial execution Suppose we have two sentence pairs, p_1 = "*chat bleu* vs. *blue cat*" and p_2 = "*chat* vs. *cat*". This yields three alignments:

$$\begin{aligned}
A_1 &= (1 \leftrightarrow 1; 2 \leftrightarrow 2) \quad (\text{so } \textit{chat} \text{ aligned to } \textit{blue} \text{ in } p_1) \\
A_1' &= (1 \leftrightarrow 2; 2 \leftrightarrow 1) \quad (\text{so } \textit{chat} \text{ aligned to } \textit{cat} \text{ in } p_1) \\
A_2 &= (1 \leftrightarrow 1) \quad (\text{only one possible choice})
\end{aligned}$$

		$\mathrm{awt}(A_1)$	$\mathrm{awt}(A_1')$	$\mathrm{awt}(A_2)$	$\mathrm{tr}(\mathit{chat} \rightarrow \mathit{blue})$	$\mathrm{tr}(\mathit{chat} \rightarrow \mathit{cat})$	$\mathrm{tr}(\mathit{bleu} \rightarrow \mathit{blue})$	$\mathrm{tr}(\mathit{bleu} \rightarrow \mathit{cat})$
a.	Init	1/2	1/2	1	–	–	–	–
b.	Up-tr	“	“	“	1/2	3/2	1/2	1/2
c.	SNorm-tr	“	“	“	1/4	3/4	1/2	1/2
d.	Update-a	1/8	3/8	3/4	“	“	“	“
e.	SNorm-a	1/4	3/4	1	“	“	“	“
f.	Update-tr	“	“	“	1/4	7/4	3/4	1/4
g.	Snorm-tr	“	“	“	1/8	7/8	3/4	1/4