

Creating a Bilingual Lexicon

Download data

Data normalization (depunct, lower case)

Build frequency tables

$\text{prob}(e)$

number of occurrences of e / all e

Find all pairs

For each sentence pair

for (f in french sentence)

for (e in english sentence)

output e, f

Cumulate the pairs

Find $p(e|f)$

Want to find the most probable e given f

$$p(e|f) = p(f|e) p(e)$$

$p(e)$ is calculated on the English words **not** pairs

$p(f|e)$ is calculated on the pairs

for all pairs with same e

$p(f|e)$ occurrence of f in all e pairs / number of e

$p(f|e) * p(e)$ gives score for each e, f

Sort them to find the best e for an f

Points to Note

- ▣ *Probabilities get *very* small*
 - ▣ *Use log probabilities or*
 - ▣ *Multiply every probability by 1000 or 1000000*
- ▣ *Very common words are at the top*
 - ▣ *From the resulting list remove common words*
 - ▣ *Maybe remove the top 20 frequent e.g.*
 - ▣ *#1 the chaise*
 - ▣ *#2 a chaise*
 - ▣ *#3 to chaise*
 - ▣ *#4 of chaise*
 - ▣ *#5 chair chaise*
 - ▣ *#6 in chaise*

Points to Note

- ▣ *Check you algorithm with dev-test*
 - ▣ *Use the dev-test (french english) lexicon*
 - ▣ *Extract the french words*
 - ▣ *Find your best english words for each of the french words*
 - ▣ *Calculate your F score with respect to the dev-test answers*
- ▣ *Submission*
 - ▣ *Use the french word list to generation your best English words.*

Takes up too much Space/Time

- *Think about efficient representations*
- *Use disk rather than memory*
- *Sort things to get faster access*

