Traitements de Corpus — 7

Background Reading:

Traitements probabilistes et corpus, Traitement automatique des langues Vol. 36, 1995, no. 1–2: E. Gaussier & J-M. Langé (Modèles statistiques pour l'extraction de lexiques bilingues), 133–155.

C. Manning & H. Schütze *Foundations of Statistical Natural Language Processing*, MIT Press, 1999. Chapter 7.

Learning lexical information (2)

- text alignment
- statistical machine translation
- collocations

Word sense disambiguation

Statistical generation

Learning Lexical Information (2)

Text alignment

Paragraph / sentence / word alignment, for:

- bilingual lexicography
- building terminology resources for translation
- support for foreign language teaching
- translation memory systems
- statistical machine translation

Calculate scores for possible alignments

- sentence length comparison: words / characters
- lexical overlap: cognates / translation equivalents

Church & Gale: sentences of same length more likely to be aligned

Kay & Röscheisen: a pair of sentences containing an aligned pair of words is itself aligned

Initial set of potentially alignable sentences

1. compute candidate set of pairs of aligned words (where most of the sentences in which one word occurs are alignable with sentences in which other word occurs)
2. find pairs of sentences containing many pair instances, commit to their alignment, and repeat from 1 until convergence

Example pairings: *article – Artikel*, *question – Frage*, *Board – Beschwerdekammer*, *Board – Grossen*

Potential problems

- phrasal translations
- structurally different languages (e.g. Japanese vs. Indo-European)
- alignments that are not 1:1, 2:1, 1:2, 2:2
- short texts (not enough repeated words)
- sensitive to sentence boundary accuracy

Statistical machine translation

$$\boxed{\text{language model } P(e)} \rightarrow \boxed{\text{translation model } P(f|e)} \rightarrow \boxed{\text{decoder } argmax_e P(e|f)}$$

Application of the *noisy channel model*

- for the language model, can use n-gram model of e (English)
- for the translation model, sum over all possible alignments of French (f) words to English words, and for each alignment multiply the individual words' translation probabilities, e.g.

  $P(Jean\ aime\ Marie|John\ loves\ Mary)$ for the alignment (*Jean*, *John*), (*aime*, *loves*), (*Marie*, *Mary*)
  $= P(Jean|John) \times P(aime|loves) \times P(Marie|Mary)$

- for the decoder, compute $argmax_e P(e|f) = argmax_e P(e)P(f|e)$ to score partial segments of (English) sentence built incrementally, at each stage extending current set of hypotheses with a small number of words and alignments and discarding least likely

Extensively experimented with at IBM in early 1990's. Around 50% accuracy. Errors were incorrect decodings, or ungrammatical decodings

Permettez que je donne un example à la chambre. → *Let me give an example in the House.*
Vous avez besoin de toute l'aide disponible. → *You need of the whole benefits available.*

More recent work aims at overcoming shortcomings due to: no notion of phrases, non-local dependencies, morphology, sparse data

Collocations

Non-compositionality, non-substitutability, non-modifiability — translation test. C.f. weaker notion of word association / co-occurrence

Particular important classes of collocation:

- light verbs
- phrasal verbs
- proper names
- terminological expressions

Raw frequency of word bigrams doesn't give decent collocation list, but in conjunction with part-of-speech sequence filter (e.g. AN, NN, AAN, ANN, NAN, NNN, NPN) does give reasonable results

Variance-based collocation discovery: fixed-width window to find variable phrase collocations

The plan has gathered strong business support.

mean and variance of distance between bigrams, e.g. *strong support*, *strong* at position -2 with respect to *support*

Hypothesis testing and collocations: do words occur together more than would be expected by chance?

- χ^2 test compares observed frequencies with frequencies that would be expected if word occurrences were independent

	$w_1 = strong$	$w_1 \neq strong$	
$w_2 = support$	(4.09) **150**	(26594.9) **26449**	26599
$w_2 \neq support$	(14943.9) **14798**	(97242155) **97242301**	97257099
	14948	97268750	97283698

$$\frac{(150-4.09)^2}{4.09} + \frac{(26449-26594.9)^2}{26594.9} + \frac{(14798-14943.9)^2}{14943.9} + \frac{(97242301-97242155)^2}{97242155} = 5208$$

critical value for χ^2 at level 0.05 with 1 degree of freedom is 3.8

since $5208 > 3.8$, *strong support* occurs more than would be expected by chance

N.B. χ^2 test not reliable for small sample sizes (< 20 in total, or 20–40 and expected value in any cell ≤ 5)

Word Sense Disambiguation

Assigning senses to word usages. Done by looking at context. Some words have distinct senses (e.g. *bank*), others less distinct (e.g. *title*), and some where related senses can often not be distinguished in actual usage (e.g. *competition*):

Banks *close early today.* (= 'financial institution', not 'land bordering a river etc.')
This painting has no **title**. (= 'name', or 'label giving name')
For better or for worse, this would bring **competition** *into the industry.* (= 'act of competing', and 'competing companies')

Important for a number of language processing applications, such as automatic translation, information retrieval (more generally systems that depend on word meanings)

Related to part-of-speech tagging task (PoS tags vs. word sense tags / local structural cues vs. content words in broader context)

Supervised / unsupervised / dictionary-based disambiguation

Supervised — using Baysian classification (Gale, Church & Yarowsky, 1992)

for word w with senses s_k: $score(s_k) = \frac{C(s_k)}{C(w)} \Pi_{v \in context} \frac{C(v, s_k)}{C(s_k)}$

drug	Sense	Contextual clues
	medicine	*prescription, prices, patent, consumer*
	illegal substance	*abuse, alcohol, cocaine, traffickers*

In training data: $\frac{C(prescription,\text{medicine})}{C(\text{medicine})} > \frac{C(prescription,\text{illegal substance})}{C(\text{illegal substance})}$

Unsupervised — using context-group discrimination (Schütze, 1998)

maximise the probability of the corpus: $\Pi_{contexts} \Sigma_{1 \leq k \leq K} score(s_k)$

for given K, using the EM algorithm (iteratively re-estimating the probability of each sense of w and the probabilities of each context for a particular sense)

Dictionary-based — using dictionary sense definitions (Lesk, 1986)

a word's dictionary definitions are likely to be good contextual indicators for the senses they define. E.g. *cone*:

1. a mass of ovule-bearing or pollen-bearing scales or bracts in trees of the pine family or in cycads that are arranged ...
2. something that resembles a cone in shape: as in ... a crisp cone-shaped wafer for holding ice cream.

then ...*tree*... *cone* ... → sense 1; ...*ice*... *cone* ... → sense 2

or overlap between definitions of words in context and of word being disambiguated:

Mary often liked to collect fir cones in the autumn.

— using thesaurus categories (Walker, 1987)

$score(s_k) =$ number of words in the context that have the same subject code for one of their senses as s_k does

In LDOCE:

tree (1) PM (= 'plant names') a type of tall plant with a wooden trunk and branches, that lives for many years

cone (1) BO (= 'botany (not plant names)') ...

Topic-based techniques perhaps appropriate for general text, but problems with particular domain:

- context word *mouse* in computer manuals rarely evidence for sense 'mammal'
- proper names or new / technical vocab leads to problems in coverage (e.g. *Navratilova* very good indicator for 'sports')

Can overcome latter problem with categorisation of words based on categorisation of contexts they appear in. But still topic-based, and certain senses of some words spread over several topics (e.g. *interest* (stake) in a *farm* / *record label* / *company's US assets*)

In general, WSD much easier for words with uneven distribution of senses (high *baseline*)

Statistical Generation

Still only relatively small amount of work compared with statistical analysis. Motivations for statistical approach to tactical generation:

- use broad-coverage grammars, but they tend to overgenerate
- underspecify input LF and leave it to system to find best realisation

Basic approach: get standard realiser to produce several alternatives, and train an n-gram language model to choose between them

turning —need —you
—after —...

→

—the —road junction —you —need —is —after
—turning —require —later than

—your —required —turning
—needed

Some phenomena may be better treated separately rather than with the generic language model (e.g. article choice, adjective ordering)