Informatics 1: Data & Analysis

Lecture 14: Example Corpora Applications

Ian Stark

School of Informatics
The University of Edinburgh

Tuesday 12 March 2013
Semester 2 Week 8

Lecture Plan

XML

We start with technologies for modelling and querying *semistructured data*.

- Semistructured Data: Trees and XML
- Schemas for structuring XML
- Navigating and querying XML with XPath

Corpora

One particular kind of semistructured data is large bodies of written or spoken text: each one a *corpus*, plural *corpora*.

- Corpora: What they are and how to build them
- Applications: corpus analysis and data extraction

Applications of Corpora

Answering empirical questions in linguistics and cognitive science:

- Corpora can be analyzed using statistical tools;
- Hypotheses about language processing and language acquisition can be tested;
- New facts about language structure can be discovered.

Engineering natural-language systems in AI and computer science:

- Corpora represent the data that these language processing systems have to handle;
- Algorithms can find and extract regularities from corpus data;
- Text-based or speech-based computer applications can learn automatically from corpus data.

Sample Linguistic Application: Collocations

A *collocation* is a sequence of words that occurs ‘atypically often’ in language usage. For example:

- To “run amok”: the verb “run” can occur on its own, but “amok” does not.
- To say “strong tea” is much more natural English than “powerful tea” although the literal meanings are much the same.
- Phrasal verbs such as “make up” or “make do”.
- “heartily sick”, “heated argument”, “commit a crime”,...

The *Macmillan Collocations Dictionary* provides extensive lists of collocations specifically for those learning English.

The inverted commas around ‘atypically often’ are because we need statistical ideas to make this precise.

Identifying Collocations

We would like to automatically identify collocations in a large corpus.

For example, collocations in the Dickens corpus involving the word "tea".

- The bigram "strong tea" occurs in the corpus. This is a collocation.
- The bigram "powerful tea", in fact, does not.
- However, "more tea" and "little tea" also occur in the corpus.

  These are not collocations. These word sequences do not occur with any frequency above what would be suggested by their component words.

The challenge is: how do we detect when a bigram (or n-gram) is a collocation?

Looking at the Data

Here are the most common bigrams from the Dickens corpus where the first word is “strong” or “powerful”.

strong	and	31
	enough	16
	in	15
	man	14
	emphasis	11
	desire	10
	upon	10
	interest	8
	a	8
	as	8
	inclination	7
	tide	7
	beer	7

powerful	effect	3
	sight	3
	enough	3
	mind	3
	for	3
	and	3
	with	3
	enchanter	2
	displeasure	2
	motives	2
	impulse	2
	struggle	2
	grasp	2

Filtering Collocations

We observe the following from the bigram tables.

- Neither "strong tea" nor "powerful tea" are frequent enough to make it into the top 13.
- Some potential collocations for "strong": like "strong desire", "strong inclination", and "strong beer".
- Some potential collocations for "powerful": like "powerful effect", "powerful motives", and "powerful struggle".
- A possible problem: bigrams like "strong and", "strong enough" and "powerful for", have high frequency. These do not seem like collocations.

To distinguish collocations from non-collocations, we need some way to filter out noise.

Statistics We Need

Problem: Words like "for" and "and" are very common anyway: they occur with "strong" by chance.

Solution: Use *statistical tests* to identify when the frequency of a bigram is atypically high given the frequencies of its constituent words.

	"beer"	¬"beer"	Total
"strong"	7	618	625
¬"strong"	127	2310422	2310549
Total	134	2311040	2311174

In general, statistical tools offer powerful methods for the analysis of all types of data. In particular, they provide the principal approach to the quantitative (and qualitative) analysis of unstructured data.

We shall return to the problem of finding collocations later in the course, when we have appropriate statistical tools at our disposal.

Engineering Natural-Language Systems

Two Informatics system-building examples which use corpora extensively:

- Natural Language Processing (NLP): Computer systems that understand or produce text. For example:
  - Summarization: Take a text, or multiple texts, and automatically produce an abstract or summary. See for example *Newsblaster*.
  - Machine Translation (MT): Take a text in a source language and turn it into a text in the target language. For example *Google Translate* or *Microsoft Translator*.
- Speech Processing: Systems that understand or produce spoken language.

Building these draws on probability theory, information theory and machine learning to extract and use the information in large text corpora.

Example: Machine Translation

The aim of *machine translation* is to automatically map sentences in one source language to corresponding sentences in a different target language, while preserving the meaning of the text.

Historically, there have been two major approaches:

- Rule-based Translation: Long history including *Systran* and *Babel Fish* (Alta Vista, then Yahoo, now disappeared).
- Statistical Translation: Much recent growth, leading to *Google Translate* and *Microsoft Translator*.

Both approaches make use of multilingual corpora.

"The Babel fish," said The Hitchhiker's Guide to the Galaxy quietly,
" is small, yellow and leech-like, and probably the oddest thing in the Universe"

Rule-Based Machine Translation

A typical rule-based machine translation (RBMT) scheme might include:

1. Automatically assign part-of-speech information to source sentence.
2. Build up syntax tree of source sentence using grammatical rules.
3. Map parse tree in source language to translated sentence, using a dictionary to perform translation at the word level, and a collection of rules to infer correct inflections and word ordering for translated sentence.

Some systems use an *interlingua* between the source and target language.

In any real implementations each of these steps will be much refined; nonetheless, the central point is to have the system translate sentence by identifying its structure and, to some extent, its meaning.

RBMT systems use corpora for machine learning of part-of-speech information and grammatical structures.

Examples of Rule-Based Translation

From http://www.systranet.com/translate

The capital city of Scotland is Edinburgh

English $\longrightarrow$ German

Die Hauptstadt von Schottland ist Edinburgh

German $\longrightarrow$ English

The capital of Scotland is Edinburgh

Examples of Rule-Based Translation

From http://www.systranet.com/translate

Sales of processed food collapsed across Europe after the news broke.

English $\longrightarrow$ French

Les ventes de la nourriture traitée se sont effondrées à travers l'Europe après que les actualités se soient cassées.

French $\longrightarrow$ English

The sales of treated food crumbled through Europe after the news broke.

Examples of Rule-Based Translation

From http://www.systranet.com/translate and Robert Burns.

My love is like a red, red rose
That's newly sprung in June

English $\longrightarrow$ Italian

Il mio amore è come un rosso, rosa rossa
Quello recentemente è balzato a giugno

Italian $\longrightarrow$ English

My love is like red, pink a red one
That recently is jumped to june

Issues with Rule-Based Translation

A major difficulty with rule-based translation is to include a large enough collection of rules to sufficiently cover the very many special cases and nuances in natural language usage.

As a result of this, rule-based translations often have a very unnatural feel.

This issue is a major one, and rule-based translation systems have not yet overcome this problem.

However, even the translations seem a little rough to read, they may still be enough to successfully communicate meaning.

(The problem with the example translation on the last slide is of a different nature. The source text is poetry, where huge liberties are taken with grammar and use of vocabulary. This puts it far outside the scope of rule-based translation.)

Statistical Machine Translation

This uses a corpus of *parallel texts*, where the same text is rendered in both source and target language. Translation might go like this:

1. Match words and phrases from the source sentence with occurrences of that word or phrase in the corpus.
2. Look at the matched words and phrases, as they occur in the parallel texts, and use statistical methods to select preferred translations.
3. Do some *smoothing* to find appropriate unit sizes for phrases and to glue translated phrases together to produce the translated sentence.

Again, real implementations will refine these stages; for example, identifying the exact matching parts of parallel texts can be challenging.

To be effective, statistical translation requires a large and representative corpus of parallel texts. This corpus does not need to be heavily annotated.

Examples of Statistical Machine Translation

From http://translate.google.com

The capital city of Scotland is Edinburgh

English $\longrightarrow$ German

Die Hauptstadt von Schottland ist Edinburgh

German $\longrightarrow$ English

The capital of Scotland is Edinburgh

Examples of Statistical Machine Translation

From http://translate.google.com

Sales of processed food collapsed across Europe after the news broke.

English $\longrightarrow$ French

Les ventes de produits alimentaires transformés s'est effondré à travers l'Europe après les nouvelles brisé.

French $\longrightarrow$ English

Sales of processed food products collapsed across Europe after the news broke.

Examples of Statistical Machine Translation

From http://translate.google.com and Robert Burns.

My love is like a red, red rose
That's newly sprung in June

English $\longrightarrow$ Italian

Il mio amore è come un rosso, rosa rossa
Questo è appena nata nel mese di giugno

Italian $\longrightarrow$ English

My love is like a red, red rose
This is just born in June

Features of Statistical Machine Translation

Statistical machine translation has challenges: it requires a very large corpus of parallel texts, and is computationally expensive to carry out.

In recent years, these problems have diminished: large corpora have become available, and there have been improvements to algorithms and hardware.

Given a large enough corpus, statistical translations can produce more natural translations than rule-based translations.

Because it is not tied to grammar, statistical translation may work better with less rigid uses of language, such as poetry.

However, if statistical translation is applied to a sentence that uses uncommon phrases, not in the corpus, then it can result in nonsense, while rule-based translation may survive.

At the moment, statistical translation is dominant.