Informatics 1, 2009

School of Informatics, University of Edinburgh

Data and Analysis

Part III

Corpora

Alex Simpson

Recommended reading

The recommended textbook for Part III is:

[CL] Corpus Linguistics

Tony McEnery & Andrew Wilson

Edinburgh University Press,

2nd Edition, 2001

Chapter 2: What is a Corpus and What is in It?

Part III — Corpora

III.1 Introduction to corpora

III.2 Building a corpus

III.3 Querying a corpus

Required reading: Chapter 2 of [CL], start of chapter to end of §2.2.1.

Natural language as data

Written or spoken natural language has plenty of *internal structure*: it consists of words, has phrase and sentence structure, etc.

Nevertheless, on a computer, it is represented as a *text file*: simply a sequence of characters.

This is an example of *unstructured data*: the data format itself has no structure imposed on it (other than the sequencing of characters).

Often, however, it is useful to annotate text by marking it up with additional information (e.g. linguistic information, semantic information).

Such marked-up text, is a widespread and very useful form of *semistructured data*.

What is a corpus?

The word *corpus* (plural *corpora*) is Latin for “body”.

It is used in (both computational and theoretical) linguistics as a word to describe *a body of text*, in particular a body of written or spoken text.

In practice, a *corpus* is a body of written or spoken text, from a particular language variety, that meets the following criteria.

1. sampling and representativeness;
2. finite size;
3. machine-readable form;
4. a standard reference.

Sampling and representativeness

In linguistics, corpora provide data for *empirical linguistics*

That is, corpora provide data that is used to investigate the nature of linguisitic practice (i.e., of real-world language usage), for the chosen language variety

For obvious practical reasons, a corpus can only contain a *sample* of instances of language usage (albeit a potentially large sample)

For such a sample to be useful for linguistic analysis, it must be chosen to be *representative* of the kind of language practice being analysed.

For example, the complete works of Shakespeare would not provide a representative sample for Elizabethan English.

Finiteness

Furthermore, corpora usually have a fixed *finite* size. It is decided at the outset how the language variety is to be sampled and how much data to include. An appropriate sample of data is then compiled, and the corpus content is fixed.

N.B. *Monitor corpora* (which are beyond the scope of this course) are an exception to the fixed size rule.

While the finite size rule for a corpus is obvious, it contrasts with theoretical lingustics, where languages are studied using *grammars* (e.g. context-free grammars) that potentially generate infinitely many sentences.

Machine readability

Historically, the word “corpus” was used to refer to a body of printed text.

Nowadays, corpora are almost universally machine (i.e. computer) readable. (Since this is an Informatics course, we are anyway only interested in such corpora.)

Machine-readable corpora have several obvious advantages over other forms:

- They can be huge in size (billions of words)
- They can be efficiently searched
- They can be easily (and sometimes automatically) annotated with additional useful information

Standard reference

A corpus is often a standard reference for the language variety it represents.

For this, the corpus has to be widely available to researchers.

Having a corpus as a standard reference allows competing theories about the language variety to be compared against each other on the same sample data

The usefulness of a corpus as a standard reference depends upon all the preceeding three features of corpora: representativeness, fixed finite size and machine readability.

Summarizing

In practice, a *corpus* is generally a widely available fixed-sized body of machine-readable text, sampled in order to be maximally representable of the language variety it represents.

Note, however, not every corpus will have all of these characteristics.

Some prominent English language corpora

- The *Brown Corpus* of American English was compiled at Brown University and published in 1967. It contains around 1,000,000 words.
- The *British National Corpus (BNC)*, published mid 1990's, is a 100,000,000-word text corpus intended to representative of written and spoken British English from the late 20th century.
- The *American National Corpus (ANC)* is an ongoing project to create an electronic text corpus of written and spoken American English since 1990. The aim is to create a 100,000,000-word corpus.

  The first release, made available (to subscribers only) in 2003, contains 11,000,000 words and was provided in XML format.
- The *Oxford English Corpus (OEC)* is an English corpus used by the makers of the Oxford English Dictionary. It is the largest text corpus of its kind, containing over 2,000,000,000 words. It is in XML format.

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 language processing system have to handle;
- algorithms exist to extract regularities from corpus data;
- text-based or speech-based computer applications can learn automatically from corpus data.

Two forms of corpus

There are two forms of corpus: *unannotated*, i.e. consisting of just the raw language data, and *annotated*.

Unannotated corpora are examples of *unstructured data*.

Annotated corpora are examples of *semistructured data*.

The four English language corpora on slide II: 11 are all annotated.

Annotations are extremely useful for many purposes. They will play an important role in future lectures.

Simple questions corpora can answer

Assume a corpus that consists of the Arthur Conan Doyle story *A Case of Identity*.

Question 1. Find all lines containing the word “Holmes”.

- My dear fellow.” said Sherlock Holmes as we sat on either
- a realistic efect,” remarked Holmes. “This is wanting in the
- said Holmes, taking the paper and glancing his eye down
- “I have seen those symptoms before,” said Holmes, throwing
- merchant-man behind a tiny pilot boat. Sherlock Holmes welcomed
- You’ve heard about me, Mr. Holmes,” she cried, “else how

. . .

Question 2. Find all lines beginning with the word “Holmes”.

- Holmes, when she married again so soon after father’s death,
- Holmes alone, however, half asleep, with his long, thin form
- Holmes. “He has written to me to say that he would be here at
- Holmes had been talking, and he rose from his chair now with a

. . .

Question 3. Find all lines starting with an upper case letter.

- A Case of Identity
- The husband was a teetotaler,
- There was no other woman
- Take a pinch of snuff, Doctor, and acknowledge that I
- The larger crimes are apt to be the simpler, for the
- And yet even here we may discriminate.
- When a woman has a secret
- Etherege, whose husband you found so easy when the

But is the kind of information provided by these three questions really useful?

Frequencies

Frequency information obtained from corpora is often useful for answering scientific or engineering questions.

Token count N: number of tokens (words, punctuation marks, etc.) in a corpus (i.e., size of the corpus).

Type count: number of *different* tokens in a corpus.

Absolute frequency $f(t)$ *of a type* t: number of tokens of type t in a corpus.

Relative frequency of a type t: absolute frequency of t normalized by the token count, i.e., $f(t)/N$.

Frequencies (example)

The British National Corpus (BNC) is an important reference.

Let's compare some counts from the BNC with counts from our sample corpus *A Case of Identity*

	BNC	A Case of Identity
Token count N	100,000,000	7,006
Type count	636,397	1,621
f(Holmes)	890	46
f(Sherlock)	209	7
f(Holmes)$/N$	.0000089	.0066
f(Sherlock)$/N$	.00000209	.000999

Unigrams

We can now ask questions such as: what are the most frequent words in a corpus?

- Count absolute frequencies of all word types in the corpus;
- tabulate them in an ordered list;
- results: list of *unigram* frequencies (frequencies of individual words).

The next slide compares unigram frequencies for BNC and *A Case of Identity*.

Unigrams (example)

BNC		A Case of Identity	
6,184,914	the	350	the
3,997,762	be	212	and
2,941,372	of	189	to
2,125,397	a	167	of
1,812,161	in	163	a
1,372,253	have	158	I
1,088,577	it	132	that
917,292	to	117	it

N.B. The article “the” is the most frequent word in both corpora; prepositions like “of” and “to” appear in both lists; etc.

n-grams

The notion of unigram can be generalized:

- *bigrams* — pairs of adjacent words
- *trigrams* — triples of adjacent words
- n*-grams* — n-tuples of adjacent words.

As the value of n increases, the units become more linguistically meaningful.

n-grams (example)

Compute the most frequent n-grams in *A Case of Identity*, for $n = \mathbf{2, 3, 4}$.

bigrams		trigrams		4-grams	
40	of the	5	there was no	2	very morning of the
23	in the	5	Mr. Hosmer Angel	2	use of the money
21	to the	4	to say that	2	the very morning of
21	that I	4	that it was	2	the use of the
20	at the	4	that it is	2	the King of Bohemia

N.B. n-gram frequencies get smaller with increasing n. As more word combinations become possible, there is increased *data sparseness*.

Corpora in Informatics

Corpora are used extensively in two areas of informatics:

- *Natural Language Processing (NLP)* builds computer systems that understand or produce text. Example applications that rely on corpus data include:
    - *Summarization:* take a text and compress it, i.e., produce an abstract or summary. Example: Newsblaster.
    - *Machine Translation (MT):* take a text in a source language and turn it into a text in the target language. Example: Babel Fish.
- *speech processing* develops systems that understand or produce spoken language.

The techniques applied rely on probability theory, information theory and machine learning to extract statistical regularities from corpora.

Example translation by AltaVista Babel Fish.

O, my love is like a red, red rose,
That is newly sprung in June.

Robert Burns (1759–1796)

English ⟶ Italian:

La O, il mio amore è come un rosso, colore rosso è aumentato,
che recentemente è balzato in giugno.

Italian ⟶ English:

Or, my love is like a red one, red color is increased,
than recently it is jumped in June.

There is still room for research!

Part III — Corpora

III.1 Introduction to corpora

III.2 Building a corpus

III.3 Querying a corpus

Required reading: Chapter 2 of [CL], §2.2.2.

Building a corpus

in the last lecture we defined a corpus as a collection of textual or spoken data satisfying:

- sampled in a certain way;
- finite in size;
- available in machine-readable form;
- often serving as a standard reference.

To build a corpus we need to perform two tasks:

- Collect corpus data — this involves *balancing* and *sampling*
- In the case of an annotated corpus, add meta-information —- this is called *annotation*

Balancing and sampling

Balancing ensures that the linguistic content of a corpus represents the full variety of the language sources that the corpus is intended to provide a reference for.

Example A balanced text corpus includes texts from many diffeerent types of source (depending on the language variety); e.g., books, newspapers, magazines, letters, etc.

Sampling ensures that the material is representative of the types of source.

Example Sampling from newspaper text: select texts randomly from different newspapers, different issues, different sections of each newspaper.

Balancing

Things to take into account when balancing:

- *language type*: may wish to include samples from some or all of:
  - edited text (e.g., articles, books, newswire);
  - spontaneous text (e.g., email, Usenet news, letters);
  - spontaneous speech (e.g., conversations, dialogs);
  - scripted speech (e.g., formal speeches).
- *genre:* fine-grained type of material (e.g., 18th century novels, scientific articles, movie reviews, parliamentary debates)
- *domain*: what the material is about (e.g., crime, travel, biology, law);

Examples of balanced corpora

Brown Corpus: a balanced corpus of written American English:

- one of the earliest machine-readable corpora;
- developed by Francis and Kucera at Brown in early 1960's;
- 1M words of American English texts printed in 1961;
- sampled from 15 different genres.

British National Corpus: large, balanced corpus of British English.

- one of the main reference corpora for English today;
- 90M words text; 10M words speech;
- text part sampled from newspapers, magazines, books, letters, school and university essays;
- speech recorded from volunteers balanced by age, region, and social class; also meetings, radio shows, phone-ins, etc.

Genres and domains in the Brown Corpus

The 15 genres are labelled A to R (letters I, O and Q are omitted); e.g.:

Genre A: PRESS (Reportage) — 44 texts
Domains: Political; Sports; Society; Spot News; Financial; Cultural

Genre B: PRESS (Editorial) — 27 texts
Domains: Institutional Daily; Personal; Letters to the Editor

Genre C: PRESS (Reviews) — 17 texts
Domains: theatre; books; music; dance

Genre J: LEARNED — 80 texts
Domains: Natural Sciences; Medicine; Mathematics; Social and Behavioral Sciences; Political Science, Law, Education; Humanities; Technology and Engineering

Comparison of some standard corpora

Corpus	Size	Genre	Modality	Language
Brown Corpus	1M	balanced	text	American English
British National Corpus	100M	balanced	text/speech	British English
Penn Treebank	1M	news	text	American English
Broadcast News Corpus	300k	news	speech	7 languages
MapTask Corpus	147k	dialogue	speech	British English
CallHome Corpus	50k	dialogue	speech	6 languages

Pre-processing and annotation

Raw data from a linguistic source can't be exploited directly. We first have to perform:

- *pre-processing:* identify the basic units in the corpus:
  - tokenization;
  - sentence boundary detection;
- *annotation:* add task-specific information:
  - parts of speech;
  - syntactic structure;
  - dialogue structure, prosody, etc.

Tokenization

Tokenization: divide the raw textual data into tokens (words, numbers, punctuation marks).

Word: a continuous string of alphanumeric characters delineated by whitespace (space, tab, newline).

Example: potentially difficult cases:

- amazon.com, Micro$oft
- John’s, isn’t, rock’n’roll
- child-as-required-yuppie-possession

  (As in: “The idea of a child-as-required-yuppie-possession must be motivating them.”)
- cul de sac

Sentence Boundary Detection

Sentence boundary detection: identify the start and end of sentences.

Sentence: string of words ending in a full stop, question mark or exclamation mark.

This is correct 90% of the time.

Example: potentially difficult cases:

- Dr. Foster went to Gloucester.
- He said “rubbish!”.
- He lost cash on lastminute.com.

The detection of word and sentence boundaries is particularly difficult for *spoken data*.

Corpus Annotation

Annotation: adds information that is not explicit in the data itself, increases its usefulness (often application-specific).

Annotation scheme: basis for annotation, consists of a tag set and annotation guidelines.

Tag set: is an inventory of labels for markup.

Annotation guidelines: tell annotators (domain experts) how tag set is to be applied; ensure consistency across different annotators.

Part-of-speech (POS) annotation

Part-of-speech (POS) tagging is the most basic kind of linguistic annotation.

Each linguistic token is assigned a code indicating its *part of speech*, i.e., basic grammatical status.

Examples of POS information:

- singular common noun;
- comparative adjective;
- past participle.

POS tagging forms a basic first step in the disambiguation of homographs.

E.g., it distinguishes between the verb “boot” and the noun “boot”.

But it does not distiguish between “boot” meaning “kick” and “boot” as in “boot a computer”, both of which are transitive verbs.

Example POS tag sets

- CLAWS tag set (used for BNC): 62 tags;
- Brown tag set (used for Brown corpus): 87 tags:
- Penn tag set (used for the Penn Treebank): 45 tags.

Category	Examples	CLAWS	Brown	Penn
Adjective	happy, bad	AJ0	JJ	JJ
Adverb	often, badly	PNI	CD	CD
Determiner	this, each	DT0	DT	DT
Noun	aircraft, data	NN0	NN	NN
Noun singular	woman, book	NN1	NN	NN
Noun plural	women, books	NN2	NN	NN
Noun proper singular	London, Michael	NP0	NP	NNP
Noun proper plural	Australians, Methodists	NP0	NPS	NNPS

POS Tagging

Idea: Automate POS tagging: look up the POS of a word in a dictionary.

Problem: POS ambiguity: words can have several possible POS's; e.g.:

Time flies like an arrow. (1)

time: singular noun or a verb;
flies: plural noun or a verb;
like: singular noun, verb, preposition.

Combinatorial explosion: (1) can be assigned $\mathbf{2 \times 2 \times 3 = 12}$ different POS sequences.

Need to take sentential context into account to get POS right!

Probabilistic POS tagging

Observation: words can have more than one POS, but one of them is more frequent than the others.

Idea: assign each word its most frequent POS (get frequencies from manually annotated training data). Accuracy: around 90%.

State-of-the-art POS taggers take the context into account; often use Hidden Markov Models. Accuracy: 96–98%.

Example output from a POS tagger (not XML format!):

```
Our/PRP$ enemies/NNS are/VBP innovative/JJ and/CC
resourceful/JJ ,/, and/CC so/RB are/VB we/PRP ./. They/PRP
never/RB stop/VB thinking/VBG about/IN new/JJ ways/NNS
to/TO harm/VB our/PRP$ country/NN and/CC our/PRP$
people/NN, and/CC neither/DT do/VB we/PRP ./.
```

Use of markup languages

An important general application of markup languages, such as XML, is to separate *data* from *metadata*.

In a corpus, this serves to keep different types of information apart;

- *Data* is just the raw data.

  In a corpus this is the text itself.

- *Metadata* is data about the data.

  In a corpus this is the various annotations.

Nowadays, XML is the most widely used markup language for corpora.

The example on the next slide is taken from the BNC XML Edition, which was released only in 2007.
(The previous BNC World Edition was formatted in SGML.)

```
<wtext type="FICTION">
  <div level="1">
   <head> <s n="1">
     <w c5="NN1" hw="chapter" pos="SUBST">CHAPTER </w>
     <w c5="CRD" hw="1" pos="ADJ">1</w>
   </s> </head>
   <p> <s n="2">
     <c c5="PUQ">˜</c>
     <w c5="CJC" hw="but" pos="CONJ">But</w>
     <c c5="PUN">,</c> <c c5="PUQ">˜ </c>
     <w c5="VVD" hw="say" pos="VERB">said </w>
     <w c5="NP0" hw="owen" pos="SUBST">Owen</w>
     <c c5="PUN">,</c> <c c5="PUQ">˜</c>
     <w c5="AVQ" hw="where" pos="ADV">where </w>
     <w c5="VBZ" hw="be" pos="VERB">is </w>
     <w c5="AT0" hw="the" pos="ART">the </w>
     <w c5="NN1" hw="body" pos="SUBST">body</w>
     <c c5="PUN">?</c> <c c5="PUQ">˜</c>
   </s> </p>
   ....
  </div>
</wtext>
```

Aspects of this example

The example is the opening text of J10, a novel by Michael Pearce.

Some aspects of the tagging:

- The **`wtext`** element stands for *written text*. The attribute **`type`** indicates the genre.
- The **`head`** element tags a portion of header text (in this case a chapter heading).
- The **`s`** element tags sentences. (N.B., a chapter heading counts as a sentence.) Sentences are numbered via the attribute **`n`**.
- The **`w`** element tags words. The attribute **`pos`** is a POS tag, with more detailed POS information given by the **`c5`** attribute, which contains the CLAWS code. The attribute **`hw`** represents the *root form* of the word (e.g., the root form of “said” is “say”).
- The **`c`** element tags punctuation.

Syntactic annotation (parsing)

Syntactic annotation: information about the structure of sentences. Prerequisite for computing meaning.

Linguists use phrase markers to indicates which parts of a sentence belong together:

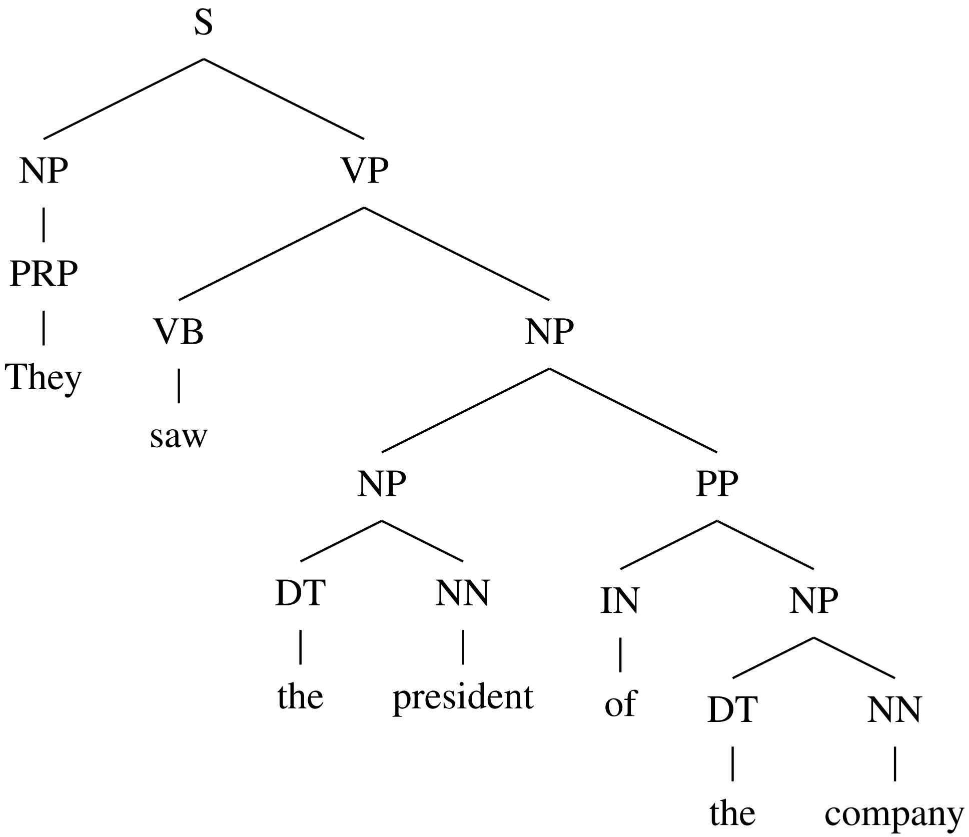
- noun phrase (NP): noun and its adjectives, determiners, etc.
- verb phrase (VP): verb and its objects;
- prepositional phrase (PP): preposition and its NP;
- sentence (S): VP and its subject.

Phrase markers group hierarchically in a *syntax tree*.

Syntactic annotation can be automated. Accuracy: around 90%.

Example syntax tree

Sentence from the Penn Treebank corpus:



The same syntax tree in XML:

```
<s>
  <np><w pos="PRP">They</w></np>
  <vp><w pos="VB">saw</w>
    <np>
      <np><w pos="DT">the</w> <w pos="NN">president</w></np>
      <pp><w pos="NN">of</w>
        <np><w pos="DT">the</w> <w pos="NN">company</w></np>
      </pp>
    </np>
  </vp>
</s>
```

Note the conventions used in the above document: phrase markers are represented as elements; whereas POS tags are given as attribute values.

N.B. The tree on the previous slide is *not* the XML element tree generated by this document.

Other Types of Annotation

- Edited text is comparatively easy to annotate;
- unscripted dialog is much harder (hesitations, false starts, slips of the tongue, cross talk);
- example for a corpus of unscripted dialog: HCRC MapTask corpus;
- rich annotation: dialog moves, disfluencies, gaze, parts of speech, syntax;
- we could also annotate prosodic structure, named entities, co-references, etc.

Part III — Corpora

III.1 Introduction to corpora

III.2 Building a corpus

III.3 Querying a corpus

Topics

- how to do something useful with corpus data and its annotation;
- how to extract statistics that are useful for linguistic questions or NLP applications;
- how to use regular expressions for queries, obtain concordances, extract collocations from corpora.

Concordances

Concordance: all occurrences of a given word, displayed in context.

More generally, one looks for all occurrences of matches for a given query expression.

- generated by concordance programs based on a user keyword;
- keyword (search query) can specify word, annotation (POS, etc.) or more complex information (e.g.,using regular expressions);
- output displayed as keyword in context: matched keyword in the middle of the line, predefined context to left and right.

Example

A concordance for all forms of the word *“remember”* in the Dickens corpus (used in tutorial 6).

```
 ’s cellar . Scrooge then <remembered> to have heard that ghost
, for your own sake , you <remember> what has passed between
e-quarters more , when he <remembered> , on a sudden , that the
corroborated everything , <remembered> everything , enjoyed eve
urned from them , that he <remembered> the Ghost , and became c
ht be pleasant to them to <remember> upon Christmas Day , who
its festivities ; and had <remembered> those he cared for at a
wn that they delighted to <remember> him . It was a great sur
ke ceased to vibrate , he <remembered> the prediction of old Ja
as present myself , and I <remember> to have felt quite uncom
```

Concordance programs

Concordances are generated automatically by concordance programs, such as the *Corpus Query Processor (CQP)* used in tutorial 6.

CQP s query engine searches corpora based on user queries over words, parts of speech, or other markup.

Regular expressions make the CQP's query language powerful.

N.B. This is the second time we have found an application for regular expressions in Data & Analysis.

CQP syntax for regular expressions

CQP makes use of the following format for regular expressions.

- **`exp1 exp2`** : first **`exp1`** then **`exp2`** in sequence.
- **`exp*`** : zero or more occurrences of exp.
- **`exp?`** : zero or one occurrences of exp.
- **`exp+`** : one or more occurrences of exp.
- **`exp1|exp2`** : either exp1 or exp2.

Question: What is the one difference here from the regular expression syntax used in DTD's (see slide II: 30)?

Example CQP query

The query:

- **`[word="remember|remembers|remembered|remembering"];`**

Returns all forms of the word “remember”, as on slide III: 50.

Here **`word`** is a *positional attribute* looking for tokens that have been marked up as words.

The value of the attribute is matched against the right-hand side of the query (here: all forms of remember).

N.B., In this case the right-hand side is a (very simple) regular expression.

Other operators

CQP offers additional regular expression operators.

The *dot operator* matches any character, e.g.

- `[word="s.ng"];`

matches **`sing`**, **`sang`**, **`sung`**, but also **`song`**, **`szng`**, **`s6ng`** etc.

The *list operator* `[...]` matches all characters in the list, e.g.

- `[word="s[iau]ng"];`

Abbreviations for subsets are allowed, e.g., `[a-d]` or `[1-6]`.

POS information and boolean expressions

The positional attribute **`word`** is available (in one form or other) in every corpus.

Most corpora contain additional annotation, e.g., part of speech information. In CQP this is given by the attribute **`pos`**.

- **`[pos="NN.*"];`**

This returns all nouns: **`NN.*`** matches **`NN`** for regular nouns, **`NNP`** and **`NNPS`** for singular and plural proper nouns, etc.

Regexes (regular expressions) can be combined using Boolean operators **`&`** (and), **`|`** (or), and **`!`** (not):

- **`[(word="like.*") & (pos!="NN.*")];`**

returns all words starting with “like” not tagged as noun.

Sequences

Queries can refer to *sequences of words*

- **`[pos="JJ.*"] [word="tea"];`**

matches all instances of the word “tea” preceded by an adjective (i.e. a word with pos value **`JJ`**).

```
now , notwithstanding the <hot tea> they had given me before
.' ' Shall I put a little <more tea> in the pot afore I go ,
o moisten a box-full with <cold tea> , stir it up on a piece
tween eating , drinking , <hot tea> , devilled grill , muffi
e , handed round a little <stronger tea> . The harp was there ; t
e so repentant over their <early tea> , at home , that by eigh
rs. Sparsit took a little <more tea> ; and , as she bent her
s illness ! Dry toast and <warm tea> offered him every night
of robing , after which , <strong tea> and brandy were administ
rsty . You may give him a <little tea> , ma'am , and some dry t
```

Collocations

Collocation: a sequence of words that occurs ‘atypically often’ in language usage

Examples:

- *run amok:* the verb “run” can occur on its own, but “amok” can’t.
- *strong tea:* sounds much better than “powerful tea” although the literal meanings are much the same.
- Phrasal verbs such as *make up* or *make off* or *make out* (but not, for example, “make in”).
- *rancid butter*, *bitter sweet*, *over and above*, etc.

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

Identifying collocations

Task: automatically identify collocations in a large corpus.

For example collocations with the word *tea* (see III: 56).

- *strong tea* occurs in the corpus.

  This is a collocation.

- *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 an *atypically* common frequency.

Problem: How do we detect when a bigram (or n-gram) is a collocation?

Finding bigrams in CQP

Use CQP to compute *bigram frequencies* for all words that occur with *strong* and *powerful*.

-
  ```
  Q1 = [word="strong"] [];
  Q2 = [word="powerful"] [];
  ```

Use the **`group`** command to obtain frequencies:

-
  ```
  group Q1 matchend word by match word;
  group Q2 matchend word by match word;
  ```

This groups together the values of word at the position **`matchend`** (the end of the matching sequence) and sorts result by word at position **`match`** (number of matches).

strong	,	52	powerful	,	5
	and	31		effect	3
	enough	16		sight	3
	.	16		enough	3
	in	15		mind	3
	man	14		for	3
	emphasis	11		and	3
	desire	10		with	3
	upon	10		enchanter	2
	interest	8		displeasure	2
	a	8		motives	2
	as	8		impulse	2
	inclination	7		struggle	2
	tide	7		grasp	2
	beer	7		friends	2

Filtering collocations

The bigram table shows:

- Neither *strong tea* nor *powerful tea* are frequent enough to make it into the top 15.
- Potential collocations for *strong*: e.g., *strong desire*, *strong inclination*, and *strong beer*;
- Potential collocations for *powerful*: e.g., *powerful effect*, *powerful motives*, and *powerful struggle*;
- Problem: The bigrams *strong and*, *strong enough*, *powerful for*, are highly frequent. These are not collocations.
- To distinguish collocations from non-collocations, we need to filter out 'noise'.

The need for statistics

Problem: Words like *for* and *and* are highly frequent on their own: they occur with *tea* by chance.

Solution: use statistical testing to detect when the frequency of a bigram is atypically high given the frequencies of its constituant words.

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 in Part V of the course.