

Introduction to Corpus Linguistics: Course Handbook

Behrang QasemiZadeh

zadeh@phil.hhu.de

Computational linguistics department, HHU

October 2017–January 2018; Draft 1.1

Preface

This material provides instructions for the Corpus Linguistics course offered at the Winter Semester of the Philosophy Faculty of HHU. In this course we discuss and share ideas on a wide range of topics related to the data-driven analyses of languages, particularly, written natural language utterances.

This course is organized in three chapters. In Chapter 1, in the first 5 sessions, we review parts of the following text books (as well as the provided references) to make sure that we are all familiar with the terminology that is used in our communications:

- ▶ Corpus Linguistics; Tony McEnery and Andrew Wilson [McEnery and Wilson, 2001].
- ▶ Corpus Linguistics: Investigating Language Structure and Use; Biber, Conrad, and Reppen [Biber et al., 1998].

The ultimate goal of these sessions is to enable us to get into constructive conversations (e.g., in the form of essays and class discussions) related to "corpus linguistics".

In the second chapter and the remaining sessions, we focus on practical applications and tools that are often used in 'corpus-based' methods. Particularly, we walk through building 'corpora', and preparing them ('annotation') for an application (e.g., 'concordance views' in a lexicography application), which includes some basics 'corpus query language' usage.

Chapter 3 is devoted to the evaluation of the course participants. In the last two sessions, we organize administrative matters, such as the homework assessment and preparing for the final examination (if applicable).

Goals of the course I

This is an introduction course and as stated above, the goals of the course are:

- ▶ Introducing the terminology that is used in corpus linguistics.
- ▶ Introducing essential methods and tools used for corpus-based studies, with a balance between theory and practice.

More detailed list can be found in [Biber et al., 1998] too.

Homework and grading I

Students are asked to do some homework (often, short essays and practices) to pass the course without marking. Although attending the course is not mandatory, it may be required for choosing a homework and to deliver it.

If a student requires grading, then homework can be reused (for the maximum of 2 credits). The title of the homework essay is chosen from the table of contents in this document, collaborations and discussions during the seminar hours, on an individual or group basis.

Homework and grading II

Essays are supposed to provide a comparison of discussions in the introduced text books for this course (see references).

Students who wish to use this course for more than 2 credits, please contact me directly (e.g., Erasmus Students). We can arrange extra work for you!

Apart from essays, students can contribute to activities such as preparation and annotation of corpora.

I kindly request you ... I

If I am not clear, e.g., you feel that I keep things behind a door,
please let me know.

Preface

Course Semester Plan

Goals of the course

Homework and grading

Language, What is it? Why is it important?

Rational approach for language analyses

Corpus linguistics: Empiricist approaches to language analyses

On Rationalism versus Empiricism

Summary

What is a corpus?

Why Machine Readable?

What is sampling and representativeness?

Example 1

Example 2

How to define sampling and representativeness in mathematical terms?

Corpus Ethics, Ethics in Corpus Linguistics, and legal matters.

What is a corpus in Machine-readable form?

Text Encoding Initiative (TEI)

Example of a Machine Readable Corpus in TEI

Alternatives to SGML–XML-Based Mark-ups  8/215

Annotation

What is annotation and annotated corpus?

Leech's Maxims of Annotation

What is 'Standard Methods' for Encoding and Representing Annotated Corpora?

Types of Annotation

Textual and extra-textual annotations

Textual and extra-textual annotations

Homework 2

Concordancer systems

Corpus query language (CQL)

Corpus-based Problem Solving Methodology?

Quantitative Methods for Corpus Linguistics: Overview

Using Simple Counts, Proportional Counts, and Tests of Significance

Collocations and Measurement of Collocational Strength: Extracting common collocations.

Multi-way contingency tables and multivariate analysis.

The **k**-Nearest Neighbours Algorithm

How to install NoSKE corpus management system?

How to compile a new corpus?

How to install R?

Language, What is it? Why is it important? I

Language is an essential element of an intellectual process (thinking). We need language to communicate our thoughts, if not with someone else but with our self, to perform reasoning. For example, how often do you engage in a conversation with yourself? E.g., when you coordinate your body movements unconsciously, or when you explain an event to yourself consciously, etc.

Language can be studied based on fundamentals of rationalism and empiricism. In Philosophy, specifically Epistemology, these two are distinguished based on the methodology they use to build a model of, or to perform some systematic study on a subject matter to provide an answer to a question.

Language, What is it? Why is it important? II

What are the answers for the question “what is language?” in rationalism and empiricism school of thoughts?

Rational approach to language analyses I

Maybe rational methods for analyses of languages can be exemplified using discussions around Formal Language and Automata Theory.

For instance, a formal language L can be defined over an alphabet Σ as a subset of Σ^* (the set of all finite **strings** over Σ).

In general, in rationalism we put more emphasis on reason than experience in order to assess the truthfulness of what comes out of our reasoning (with reference to the concept of certainty in knowledge).

Rational approach to language analyses II

The rational method defines language using another language (e.g., Logic). It borrows tools and methods of reasoning from this language (e.g., predicate calculus) to define the language they are used to study (e.g., a natural language such as English).

Colloquially, in rational methods, the focus is on what is 'theoretically' possible in a language [Biber et al., 1998].

Corpus linguistics: Empiricist approaches to language analyses I

In empirical methods of language analyses, our knowledge about the language comes primarily from its usage and real examples. As a result, as put by [McEnery and Wilson, 2001], **corpus linguistics** can be defined as the study of a language based on examples from its 'real life' usage.

The 'corpus' is a large body of examples (linguistic evidences) from 'real life' usages of a language.

Each study requires its own corpus as we discuss later.

Corpus linguistics: Empiricist approaches to language analyses II

These methods have become increasingly popular given their usage in information systems, particularly in human language technology applications, i.e., automated analysis of natural language.

On Rationalism versus Empiricism I

Rationalism vs. Empiricism is a common theme in Philosophical studies, see e.g., [Markie, 2017]. The debate is on how **A Priori** and **A Posteriori** are employed to solve a problem. A priori knowledge is assumed to be true and known independent of experience, e.g., as expressed in tautologies ("all lads are male"), or used in deduction. On the other hand, a posteriori knowledge is justified by experience and empirical evidence, e.g., as in drug-testing.

This debate has found its way in linguistics; e.g., as used by Chomsky to criticize corpus based linguistics.

On Rationalism versus Empiricism II

Disregarding the method used for investigating language, paradoxes are unavoidable by the very nature of Language: There are questions that we cannot yet answer.

We use and need both rational and empirical methods in our problem solving processes: For doing a scientific (or, scientifically reasonable) corpus-based work we need to build a hypothesis (an educated guess), to establish some framework for our problems, to define some questions, etc. — we do this rationally. Once we formulated a problem, we use a formal model, e.g., a statistical model, to form a summary of our empirical findings (e.g., word counts).

On Rationalism versus Empiricism III

In summary, we are empirically investigating questions that are formed rationally.

Therefore, rational and empirical techniques are used as complementary tools for each other.

Summary I

We define corpus linguistics as the rational use of linguistic evidence (i.e., corpora) to answer questions related to certain aspects of a language.

Corpus linguistics is not a branch of linguistics (e.g., as syntax, semantics, etc are) but a set of techniques and methods used in linguistics [McEnery and Wilson, 2001]. However, linguistic research methods can be classified methodologically as corpus-based and non-corpus based, e.g., to identify areas such as corpus-based/data-driven syntax, etc.

Summary II

Corpus-based methods are used in disciplines other than linguistics, e.g., journalism and social sciences.

What is a corpus? I

In modern corpus linguistics, Corpus, plural corpora, refers to a large collection of linguistic evidences/manifestations of any medium, which is recorded and represented in a 'machine readable' format. The most common medium are natural language text and transcriptions of recorded speech.

Certain constraints are imposed when collecting data and compiling it as a corpus. These constraints ensure that the analysis of the corpus will yield to a meaningful and rational outcome for the application that it is designed for.

What is a corpus? II

Simply put, not all collections (of, e.g., text) are corpora. They are built for a purpose, e.g., an application, or a study. Corpora are machine accessible, i.e., they can be analyzed by computers.

Question: How to assess truthfulness of results: what is the most common subject for the verb 'google' in English?

What is a corpus? Our definition. I

[McEnery and Wilson, 2001]: A finite-sized body of machine-readable text, sampled in order to maximally representative of the language variety under consideration.

[Sinclair, 1996]:¹ A corpus is a collection of pieces of language that are selected and ordered according to [explicit linguistic criteria] in order to be used as a **sample** of the language.

We replace linguistics with "any collection of explicit criteria". In the above definition, the word 'sample' is very important; what does sample mean?!

What is a corpus? Our definition. II

a small part or quantity intended to show what the whole is like.

a portion drawn from a population, the study of which is intended to lead to statistical estimates of the attributes of the whole population

'sample' is a keyword in our definition. For instance, in our definition, replace 'sample' with the word 'representation' or 'model', how this change changes our definition of corpus?

¹<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.28.1988&rep=rep1&type=pdf>

What is a corpus? characteristics. I

[McEnery and Wilson, 2001] list four main characteristics:

- ▶ Machine-readable form
- ▶ Sampling and representativeness
- ▶ Finite size
- ▶ A standard reference

Before we discuss any of these characteristics, let's think about a few examples.

Why Machine Readable? I

Mostly, for automation. Machines can be used to analyze large corpora fast and accurately.

In corpus-based data driven methods, after understanding the question we would like to answer, we need to find and collect examples that help us to answer the question. Computers can be used for a fast search and retrieval of these examples, much faster than man (but, remember humankind has done corpus-based study long time before the birth of machines).

Concordance programs are the tool most often implemented in corpus linguistics to examine corpora. Put simply, a concordance

Why Machine Readable? II

program builds a structured representation of a corpus that can be searched and retrieved fast, to serve our purpose.

What is sampling and representativeness? Example 1 I

What are common frames from Berkeley's FrameNet that are often used in English?

Can we find them for German, Italian, Persian, Turkish, etc.?

What is the nature of these frames?

What are the effects of sampling and representativeness?

What is sampling and representativeness? Example 2 I

What are the most frequently used German phrases?

What are the most frequently used German phrases in scientific writing?

What are the most frequently used German phrases to teach to foreigners?

What are morphological properties of English proper nouns for male and female?

Are there patterns in English proper nouns for places?

What is sampling and representativeness? Example 2 II

What morphological patterns can be seen in English uncountable nouns?

What are the most common syntactic structures in English?

What is the best method to compile a corpus for building an “automatic syntactic tagger”?

What is the best method to compile a corpus for analyzing morphosyntactic structure of nouns?

What is sampling and representativeness? Example 2 III

Do "fake news" that are published in social media have linguistic/extra-linguistic characteristics which can be used to distinguish them from reliable ones?

What is sampling and representativeness? A Statistical Perspective I

“In statistics, sampling refers to the selection of a subset of individuals from within a statistical population to estimate characteristics of the whole population. Two advantages of sampling are that the cost is lower and data collection is faster than measuring the entire population (Wikipedia).”

What if we cannot count our population? Is this situation possible?

For example, what is the relative frequency of the word **the** to the word **that** in English? To study this, we need to build a corpus of

What is sampling and representativeness? A Statistical Perspective II

English, is this going to be a finite corpus or an infinite one? Can we really count all the usages of the words 'the' and 'that'? We have a countably infinite set. Since it is countable we can report some stat. But, What are the effects of sampling when answering this question (the relative frequency of 'the' to 'that')?

If P is the ideal corpus (perhaps, a countably infinite set — i.e., a huge corpus), how did we sample it when creating it? If we want to sample a F sub-corpus from P , how does the sampling method affect our outcomes?

Does it have to do anything with the size of F ?

What is sampling and representativeness? A Statistical Perspective III

What about building a corpus from the web? What is a 'web corpus'?

We will get back to this topic later.

Corpus Ethics and Legal Matters I

Some ethical questions cannot be avoided in corpus linguistics and when building corpora. The least and obvious one is that we respect copyright and licenses: we use corpora that we have rights for, we cite corpora/resources that we use, etc. This is complicated enough, yet there are more issues to face when building a corpus.

What if your corpus contains sensitive personal information? Can one be persecuted based on what is in a corpus? Would you like a corpus that you have built to be used for mass surveillance programs?

What is a corpus in Machine-readable form? I

Let's emphasize that you can still do corpus linguistics without machines. But, why not using machines when they are available, reliable, fast, and cheap?

The characteristic of 'Machine-readable form' implies the requirement for a set of guidelines (how-tos), and even better, standards (and open standards) for representing contents in computers.

Plenty of such recommendations and standards are available. For example, Text Encoding Initiative (TEI) consortium is a respected working group which provides guidelines and recommendations for

What is a corpus in Machine-readable form? II

converting/maintaining/representing corpora in machine-readable forms.

Text Encoding Initiative (TEI) I

TEI maintains a standard for the representation of texts in digital form which has been used in libraries, museums, publishers, etc.

Note that using TEI does not guarantee that you have a corpus. It just tells you how to encode/structure certain content for machines. For instance, TEI tells you:

*“strings that are referring to people, places, and organizations, can be encoded by placing them in **elements** such as <rs> and <name>.”*

Text Encoding Initiative (TEI) II

These tutorials are a good place to start learning things about TEI:
<http://www.tei-c.org/Support/Learn/tutorials.xml>. Do not let the technicality of the content scare you from seeing what TEI consortium does.

About TEI I

Initially, TEI used **SGML** (The Standard Generalized Markup Language (SGML; ISO 8879:1986)).

Later **XML** replaced SGML, and TEI became a complex application of XML for **annotating** in a wide variety of **language resources**.

We define the terms annotation and language resources in the later slides.

Extensible Markup Language (XML) I

Extensible Markup Language (XML) is a markup language (HTML is another example — metalanguage).

A markup language, itself, is a system for storing 'contents about things'/documents.

Markup languages such as XML are processed with respect to references that are specified in other texts.

Reference documents for a mark-up language are maintained by an authority, e.g., a person, a group of people, an organization, etc.

Extensible Markup Language (XML) II

For instance, the World Wide Web Consortium (W3C) maintains reference documents for a number of mark-languages such as XML.

TEI P5 and Beyond I

Since TEI (version 5) uses XML, it embraces some key standards and concepts that XML offer (picking Unicode, definitions of well-formedness and validity, and DTDs) and obviously some XML-related recommendations: e.g., how to use **schema**, or **namespace** (and then to use tools such as **XPath** and **XSLT** to retrieve data using them).

The advent of web has changed the TEI encoding to an even more complex system. TEI has become part of the Semantic Web, and the open linked data movement, for instance by employing technologies such as Dublin Core, RDF, RDFS, and OWL.

TEI P5 and Beyond II

Note that other recommendations can be built on top of TEI: Music Encoding Initiative (MEI) consortium adapt TEI and customize it for presenting content in Music applications and musicology research.

Example of a Machine Readable Corpus in TEI

Let's say we want to work on George Orwell's Nineteen Eighty-Four.

Listing 1: Example of TEI-encoded corpus

```
1 <TEI xmlns="http://www.tei-c.org/ns/1.0">
2   <teiHeader>
3   </teiHeader>
4   <text>
5   </text>
6 </TEI>
```

An Example of a Machine Readable Corpus in TEI II

Let's continue with a simple example.

Listing 2: Example of TEI-encoded corpus

```
1 <TEI xmlns="http://www.tei-c.org/ns/1.0">
2   <teiHeader>
3     <fileDesc>
4       <titleStmt>
5         <title>TEI Example</title>
6       </titleStmt>
7       <publicationStmt>
8         <distributor>HHU</distributor>
9         <date value="2017-10-23">October
10        23rd, 2017</date>
        </publicationStmt>
```

An Example of a Machine Readable Corpus in TEI I II

```
11         </fileDesc>
12     </teiHeader>
13     <text lang="en">
14         <body>
15             <div id="1">
16                 <head>Minimal Example</head>
17                 <p>
18                     <emph>TEI</emph> is not
19                         complicated, said <name>
20                         John Doe</name>.
21                 </p>
22             </div>
23             <div id="2">
24                 <head>
25                     Looking for more?
26                 </head>
```

An Example of a Machine Readable Corpus in TEI I III

```
25         <p>  
26         Check TEI P5 guidelines.  
27         </p>  
28     </div>  
29 </body>  
30 </text>  
31 </TEI>
```

An Example of a Machine Readable Corpus in TEI I

The TEI P5 guidelines have most answers to your questions: <http://www.tei-c.org/Guidelines/P5/>, for example, the following question:

- ▶ What are the semantics for the elements used in the above example? Are they used properly? I want to mark logical segments such as sentences, phrases, and words in this corpus, what elements TEI recommends? Do I go beyond elements used proposed for Linguistic Segment Categories? What are Feature Structures?

An Example of a Machine Readable Corpus in TEI II

A simple example that illustrates how TEI can be used to build annotated multilingual corpora and feature structures is the MULTEXT-East project: <http://nl.ijs.si/ME/>.

Machine Readable Corpora: Alternatives to SGML–XML-Based Mark-ups I

Do I have to use XML and TEI for making a corpus machine readable? The answer is 'no'. For example, you have a machine readable corpus as soon as you collect and save some text in a 'text file' (based on the rational which is defined by your problem).

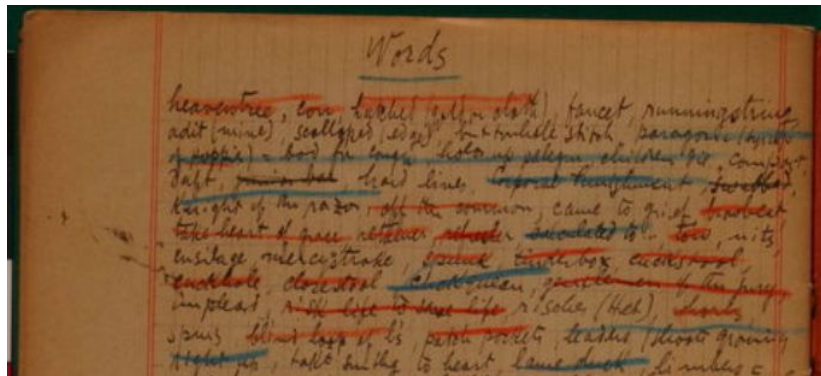
There are many corpora which are widely used but not yet encoded in TEI, e.g., Corpora (treebanks) in Universal Dependencies Project are made machine readable using the so-called CoNLL-U format:
<http://universaldependencies.org/v2/conll-u.html>.

Machine Readable Corpora: Alternatives to SGML–XML-Based Mark-ups II

```
1 1 Mary _ _ _ _ 2 nsubj 2:nsubj _
2 2 won _ _ _ _ 0 root 0:root _
3 3 silver _ _ _ _ 2 obj 2:obj _
4 4 and _ _ _ _ 5 cc E5.1:cc _
5 5 Sue _ _ _ _ 2 conj E5.1:nsubj _
6 5.1 _ _ _ _ 2 conj 2:conj _
7 6 bronze _ _ _ _ 5 orphan E5.1:dobj _
```


Machine Readable Corpora: Alternatives to SGML–XML-Based Mark-ups Broadening Horizons

What if we want to make and study a corpus of hand written notes? Can we still use tab-separated values (TSV) file? Often, the TEI system is used in these scenarios.



What is annotation and annotated corpus? I

Corpora are often classified as **unannotated** and **annotated**. The content of an unannotated corpus has a very close representation of the raw material from which it is constructed (e.g., the plain text). However, the content of an annotated corpus is 'enhanced' with various types of information.

Annotations add explicit information about certain attributes of the content of a corpus to the corpus or another corpus (annotation layers). For instance, TEI structural mark-ups such as `<p></p>` add information about the boundaries of logical text segments (paragraphs) in raw text files.

What is annotation and annotated corpus? II

The verb "to annotate" refers to the process of adding annotations to a certain content according to a particular **annotation scheme and guidelines**. From the very first moment we start to make corpora, we use an annotation system, explicitly or implicitly. Similarly, for what we discussed for encoding data into computers, the process of annotation is carried out according to a reference: an annotation scheme and guidelines that specify the meaning of what we have asserted as annotation and how this must be represented in the machine readable corpus.

Implicit guidelines are those fundamentals that you do not document but use when building a corpus. Rarely, a produced corpus can be self explanatory. We need to explicitly, preferably in

What is annotation and annotated corpus? III

a machine readable form, document the process, but are there limits in 'explicitness'?

Sometimes, the meaning of annotations can be represented accurately (e.g., the start and the end of the file/document that contains a corpus, or categorization labels (vocabulary) that we could define formally). In these cases, we can use a scheme, a machine readable reference document that defines our vocabulary. For instance, TEI uses DTDs (a document type definition file): a set of markup declarations that define a document type for SGML-family markup language such as SGML and XML. Some other times, the meaning in our annotation vocabulary cannot be

What is annotation and annotated corpus? IV

defined formally or distinguished decisively (e.g., group words according to their meanings – word sense grouping).

We often define several layers of annotations for a corpus. The layers can be cascaded in several ways, they can be dependant on each other or independent.

The assumptions that we have not explicitly encoded in our scheme, or justifications for the decisions that we made when creating an annotation schema, often goes to a document called annotation guidelines.

What is annotation and annotated corpus? V

The ultimate goal of annotations (mark-ups) is to enable us to efficiently locate, search and retrieve information we need in our corpus based study. The information that annotations provide usually cannot be accessed from the raw data (e.g., text) that we use to build a corpus.

For example, how to find verbs that have an organization as their subject in a corpus of 10,000 sentences. To do so, one may create and use two annotation layers: one annotation layer to mark all the verbs and another layer to mark organization.

Leech's Maxims of Annotation I

To maximize the usability and interchangeability of annotated corpora, Geoffrey Leech suggests the following seven principles:

1. It should be possible to remove the annotation from an annotated corpus in order to revert to the raw corpus. The complexity of this process depends on the annotation scheme and method for representing it.

E.g., if we use an underscore+part-of-speech-tag to annotate corpus such as "Claire_NP1 collects_VVZ shoes_NN2", the first maxima can be achieved by removing underscores, i.e., we can generate the original input string of "Claire collects shoes".

Leech's Maxims of Annotation II

However, sometimes recovering the original input from the corpus annotations is not that easy (e.g., corpora of transcriptions of speech such as the London-Lund Corpus of Spoken English).

2. It should be possible to extract the annotations by themselves from the text.

This is the flip side of maxim 1. Taking points 1 and 2 together, the annotated corpus should allow the maximum flexibility for manipulation by the user.

3. The annotation scheme should be based on guidelines which are available to the end user. As discussed, this document details the

Leech's Maxims of Annotation III

annotation scheme and guidelines used by the annotators. This helps to remove ambiguity and justifies an annotation decision when more than one/or no annotation was possible. We can use guidelines published for MULTEXT-East, PARSEME, Penn Treebank, ACL RD-TEC, etc. as an example.

4. It should be made clear how and by whom the annotation was carried out. E.g., if the corpus is annotated by more than one person or automatically.
5. The end user should be made aware that the corpus annotation is not infallible, but simply a potentially useful tool. Asserting annotations in a corpus implies an interpretation of the input, and

Leech's Maxims of Annotation IV

that can be erroneous or incomplete. Methods for measuring certain aspect of this feature are available and we introduce them later.

6. Annotation schemes should be based as far as possible on widely agreed and theory-neutral principles. For example, parsed corpora often adopt a basic context-free phrase structure grammar rather than implementing a narrower specific grammatical theory such as Chomsky's Principals and Parameters framework.

7. No annotation scheme has the a priori right to be considered as a standard. Standards emerge through practical consensus.

A Side-note on Language Resources and Corpora

Any corpus is a **language resource**. But there are other types of language resources, too: Grammar, Language Model, Dictionary, (Polarity lexicon, Sub-categorization dictionary, Translation lexicon, etc.), Gazetteers and Stop-words lists, Terminologies, Ontologies, Affixes list in language X, List of morphemes in language Y, etc.

Note that depending on the problem you investigate, any of these language resources may be used as a corpus in your study.

What is 'Standard Methods' for Encoding and Representing Annotated Corpora?

In fact, there is no standard format for encoding and representing annotations and annotated corpora. Numerous methods have been proposed and applied for building and representing annotated corpora (for instance, see examples for machine readability). However, as implied earlier some initiatives (e.g., TEI and its sponsored/affiliated projects in 'digital humanities') have been more successful in terms of being 'a standard through practical consensus'.

As a rule of thumb, if you are about to construct an annotated corpus, always look for what has been done before, and check the suitability of previously proposed formats and methods for your own specific application. The ultimate goal in this process is to ensure the usability of and interchangeability of the annotated corpus that you build.

Types of Annotation

What types of information are typically asserted as annotation?
[McEnery and Wilson, 2001] list some categories:

- ▶ Textual and extra-textual annotations
- ▶ Orthography annotations
- ▶ Linguistic annotations
 - ▶ Part-of-speech annotation
 - ▶ Lemmatization
 - ▶ Syntactic annotation
 - ▶ Semantic annotations
 - ▶ Those that are asserted between text units.
 - ▶ Those that are asserted about text units.
 - ▶ Discourse annotations
- ▶ Annotations for Speech Corpora
- ▶ **Problem-oriented annotations**

Textual and extra-textual annotations I

[McEnery and Wilson, 2001] use the term 'textual and extra-textual annotations' to refer to the basic annotation type that provides general information about the 'external attributes' of text, e.g., the type of content, the title of text, information about its author (name, gender, etc.) and so on. As example, [McEnery and Wilson, 2001] list the following annotation mark-ups from the **CELT** corpus:

- ▶ `<Q></Q>`: to mark questions;
- ▶ `<EX></EX>`: to mark expansions of abbreviations in text'
- ▶ `<LB>`: to indicate line breaks;

Textual and extra-textual annotations II

- ▶ `<FRN Lang="x"></FRN>`: marks foreign words in language 'x';
- ▶ `<PN></PN>`: mark names of places;
- ▶ ...

Orthography I

We briefly discussed the importance of character encoding when building machine-readable corpora: imagine the horror of encoding text with non-Roman character set before the wide spread use of Unicode!

Annotation of 'orthography information' can simply go beyond mere text encoding: Remember the example of annotating hand-written texts? How much information should we assert about the typography used in the origin, what about layout information (e.g., line and page breaks)? If annotating such information is necessary, then TEI guidelines are, perhaps, the most reliable start point.

Orthography II

The orthography annotation in corpora made from media other than text can be more challenging. For instance, for speech corpora, asserting any punctuation or logical segmentation can be seen as an interpretation of the source data (i.e., speech) by the annotator/annotators. As suggested by [McEnery and Wilson, 2001], a basic decision when building and transcribing speech corpora is about the orthography (e.g., transcriptions in the form of orthographic sentences or ‘intonation’ units?).

Linguistic Annotations I

Linguistic annotation often involves the attachment of special symbols/codes to words and other text units (e.g., phrases) to indicate certain linguistic features. These symbols/codes are sometimes called **tags**, and the annotation process may be also referred to as **tagging**.

Linguistic Annotations: Part-of-speech Tagging I

Part-of-speech annotations are one of the most commonly used linguistic annotations. The aim of part-of-speech tagging/annotation is to assign each word (lexical units) in the text a 'code' that indicates the **part-of-speech category** the word belongs to.

Traditionally, a part of speech (often abbreviated as PoS) is a group of words that have similar grammatical properties. In other words, words from the same part of speech category usually exhibit similar 'linguistics behavior' at various levels (e.g., morphology, syntax, and semantics). E.g., at the morphology level, words of similar PoS often share similar 'inflectional' patterns. These

Linguistic Annotations: Part-of-speech Tagging II

common patterns can be seen also at the syntactic and semantic levels.

For instance, one may classify words in English as **noun**, **verb**, **adjective**, **adverb**, **pronoun**, **preposition**, **conjunction**, **interjection**, **article**, **number**, and so on.

For the part-of-speech category of verbs: commonly they can inflected by adding suffixes such as **-ed**, and **-ing** to their base form; they are often the head of verb phrases, and subject/object grammatical relations to other words or phrases. These words often convey meanings related to events or processes, and so on.

Linguistic Annotations: Part-of-speech Tagging III

Part-of-speech tags used to annotate corpora are often finer than the coarse categories mentioned above. For instance, for nouns, the tags often indicate whether a noun is ‘singular’ or ‘plural’, or whether it is ‘common’ or ‘proper’.

The part-of-speech annotation task preceded by designing the inventory of part-of-speech tags, which is not a trivial task. The complexity of the proposed tag sets often differs from one project/corpus to another one. For instance, for English, we can compare PoS tags used in Penn Treebank project²—an inventory of 36 tags (which are very popular in practical natural language processing applications), with the UCREL CLAWS tagset³, and the

Linguistic Annotations: Part-of-speech Tagging IV

tagset used in the English section of MULTEXT-East project⁴ with more than 135 different categories.

²https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

³<http://ucrel.lancs.ac.uk/claws/>

⁴<http://nl.ijs.si/ME/V5/msd/html/msd-en.html#msd.msds-en>

Linguistic Annotations: Lemmatization I

Lemmatization is the process of mapping word forms to their respective 'lemmas'—as put by [McEnery and Wilson, 2001], the head word form that one would look up if one were looking for the word in a dictionary. For instance, for words forms **went, go, going, gone, goes** belong to the lemma **go**. In other words, lemmatization can be seen as the process of grouping inflected forms of words that are originated from the same source.

Lemmatization is particularly important in lexicography applications, to allow researchers to extract and examine all variants of a lexeme with a single query instead of several ones.

Linguistic Annotations: Lemmatization II

Note that lemmatization must not be confused with the so-called **stemming** process.

Linguistic Annotations: Syntax and Parsing I

We can go beyond morphosynactic categorization of words and their annotation in a corpus by asserting syntactic annotations in a corpus, which is often called parsing.

Corpora which are parsed and annotated with syntactic relationships between words are often called **Treebanks**. Treebanks can be categorized by the underlying grammatical formalism used for their annotation. Well known examples are constituent treebanks and dependency treebanks.

Linguistic Annotations: Syntax and Parsing II

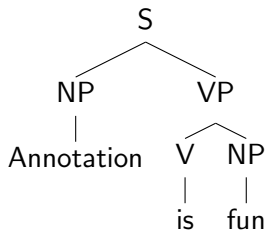


Figure: Label for a

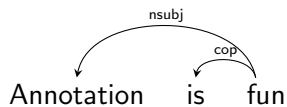


Figure: Dependency Parse

Syntax and Parsing: Example of Dependency Annotations

1	Pierre	Pierre	PROPN	NNP	-	2	compound	-	-
2	Vinken	Vinken	PROPN	NNP	-	9	nsubj	-	-
3	,	,	PUNCT	,	-	2	punct	-	-
4	61	61	NUM	CD	-	5	nummod	-	-
5	years	year	NOUN	NNS	-	6	nmod:npmmod	-	-
6	old	old	ADJ	JJ	-	2	amod	-	-
7	,	,	PUNCT	,	-	2	punct	-	-
8	will	will	AUX	MD	-	9	aux	-	-
9	join	join	VERB	VB	-	0	root	-	-
10	the	the	DET	DT	-	11	det	-	-
11	board	board	NOUN	NN	-	9	dobj	-	-
12	as	as	ADP	IN	-	15	case	-	-
13	a	a	DET	DT	-	15	det	-	-
14	nonexecutive	nonexecutive	ADJ	JJ	-	15	amod	-	-
15	director	director	NOUN	NN	-	9	nmod	-	-
16	Nov.	Nov.	PROPN	NNP	-	9	nmod:tmod	-	-
17	29	29	NUM	CD	-	16	nummod	-	-
18	.	.	PUNCT	.	-	9	punct	-	-

Syntax and Parsing: Example of Constituent Annotations

```
( (S (NP-SBJ (NP Pierre Vinken)
      ,
      (ADJP (NP 61 years)
            old)
      ,)
  (VP will
      (VP join
          (NP the board)
          (PP-CLR as
              (NP a nonexecutive director)))
      (NP-TMP Nov. 29)))
.))
```

Linguistic Annotations: Semantics I

[McEnery and Wilson, 2001] categorize semantic annotations into two broad categories:

- ▶ Annotations for marking semantic features of words in text, which are mostly about ‘word senses’;
- ▶ Annotations for marking semantic relationships between text units, e.g., a word with other words.

Linguistic Annotations: Semantics II

Word sense induction and disambiguation has been a research topic in corpus linguistics and lexical semantics for a long time. Put simply, the aim is to distinguish different meanings of words and classify them in certain way. For example, to understand and express that the work form 'bank' has at least two different meanings, one being a financial institute and another 'the land alongside a river'. Additionally, in its first sense, bank can be considered 'synonym' with bound, edge, etc. while in its second sense it is synonym with finance company or building.

There are several word-sense-tagged corpora, i.e., corpora in which word senses/meanings are annotated: Brown corpus is a classic example while corpora such as OntoNotes is a more recent one.

Linguistic Annotations: Semantics III

On the other hand, the well-known example of corpora with annotations of semantic relationships between text units are those that are developed for the so-called semantic parsing tasks. Likewise syntax treebanks, these corpora come in different forms and formalism.

Linguistic Annotations: Semantics (Example)

1	Pierre	Pierre	NNP	-	-	-	NE	-	-	-	-	-
2	Vinken	vinken	NNP	-	+	-	-	-	-	ACT-arg	-	-
3	,	,	,	-	-	-	-	-	-	-	-	-
4	61	61	CD	-	-	-	-	RSTR	-	-	-	-
5	years	year	NNS	-	+	-	-	-	EXT	-	-	-
6	old	old	JJ	-	+	-	DESCR	-	-	-	-	-
7	,	,	,	-	-	-	-	-	-	-	-	-
8	will	will	MD	-	-	-	-	-	-	-	-	-
9	join	join	VB	+	+	ev-w1777f1	-	-	-	-	-	-
10	the	the	DT	-	-	-	-	-	-	-	-	-
11	board	board	NN	-	-	-	-	-	-	PAT-arg	-	-
12	as	as	IN	-	-	-	-	-	-	-	-	-
13	a	a	DT	-	-	-	-	-	-	-	-	-
14	nonexecutive	nonexecutive	JJ	-	-	-	-	-	-	-	RSTR	-
15	director	director	NN	-	+	-	-	-	-	COMPL	-	-
16	Nov.	nov.	NNP	-	+	-	-	-	-	TWHEN	-	-
17	29	29	CD	-	-	-	-	-	-	-	-	RSTR
18	.	.	.	-	-	-	-	-	-	-	-	-

Multilingual Corpora I

So far, the corpora and examples that we discussed were limited to one language. However, in many applications we require multilingual corpora, which contain texts in several different languages. Several types of multilingual corpora exist:

- ▶ Parallel Corpora, which contains translations of a text in several languages. Parallel corpora can be **aligned** at different level of granularity (often sentence level): MULTEXT-East, the OPUS corpus, the Europarl corpus, etc.

Multilingual Corpora II

- ▶ Comparable Corpora, which contains texts from different languages. These texts come from the same domain, however, they are not necessarily parallel (or exact tr).
- ▶ ** Translation Corpora, which represent L1 texts in different languages and not translations, e.g., PAROLE corpora.

Types of Corpora I

It is possible to categorize corpora based on some of their common features and applications.

Evidently, corpora can be categorized as annotated and unannotated, and by the language(s) that they represent: English, French, German, Zaza, etc..

Corpora can be Monolingual, Bilingual, or Multilingual.

One can classify corpora and language resources primarily by their modalities (Spoken, Written, Multimodal/Multimedia).

Types of Corpora II

Another important feature is the temporal nature of our corpora. For example text documents can be mapped onto a time line by the data they are generated, e.g., publications from a series of conferences. Particularly, you often hear about diachronic and synchronic (mostly in the sense of being time-agnostic) corpora. If temporal data about text documents in your corpora are available (time they are generated, published, or even the temporal frame for their contents), then you can sort/partition your corpus using this data. Also, have a look at monitor corpora (as named by Sinclair).

Tools for Corpus Annotation I

To build and annotate corpora in real-world applications, we often use an annotation tools (or set of tools).

- ▶ Annotation tools can be simple and generic software such as text editors, spreadsheets software (such as MS Excel, OpenOffice, etc.).
- ▶ They can be a generic “annotation tool”:
 - ▶ brat rapid annotation tool: <http://brat.nlplab.org/>
 - ▶ FLAT: <https://github.com/proycon/flat>
 - ▶ WebAnno: <https://webanno.github.io/webanno/>
 - ▶ ANNIS: <http://corpus-tools.org/annis/>
 - ▶ and many many more ...

Tools for Corpus Annotation II

These annotation tools often cover a range of annotation tasks, from a simple part-of-speech annotation task to more complicated syntactic and semantic annotation. For a long list of available annotation tools, have a look at:

- ▶ <https://corpus-analysis.com/>
- ▶ http://annotation.exmaralda.org/index.php?title=Linguistic_Annotation
- ▶ <http://corpus-tools.org>

Each of these tools, expect your unannotated input corpus be in a specific format. Some of these annotation tools are hosted over the CLARIN infrastructure.⁵ For example, WebAnno is hosted by CLARIN-D and can be used freely and without the effort for installation at <https://webanno.sfs.uni-tuebingen.de/>.

Tools for Corpus Annotation III

- ▶ Alternatively, an annotation software can be developed solely for the purpose of a building a particular corpus.
 - ▶ ParsemeBot: <https://github.com/kercos/ParsemeBot>

Evidently, the nature of your annotation task, resources available to you, and the size/duration of your project is an important factor in choosing one of the above mentioned category of tools. No need to say that each category has its own pros and cons.

Manual, Automatic, or Semi-Automatic Annotation I

As many of you suggested, some annotation tasks can be automated. But, should we automate the task? What are the pros and cons of automation?

There are a number of good reasons to automate an annotation, mainly to save time and thus money.

In cases that a pre-annotated corpus and automatic tools for annotation are available, the automatically generated tags can be used, almost free of charge!

Unfortunately, in many scenarios automation of the task is not feasible due to the lack of a pre-annotated corpus for the task.

Manual, Automatic, or Semi-Automatic Annotation II

Assume you want to do part-of-speech tagging for the “old Valyrian language”. For those who does not know Valyrian language, it is a fictional language family in the series of fantasy novels by George R. R. Martin, and in their television adaptation Game of Thrones:

https://en.wikipedia.org/wiki/Valyrian_languages

As to my knowledge, there is not part-of-speech tagged corpus of old Valyrian and therefore, it is not possible to develop an automatic part-of-speech tagger for this language. Consequently, it is required that an old Valyrian speaker (e.g., a Dothraki) to be hired to do the manual annotation task!

Manual, Automatic, or Semi-Automatic Annotation III

In the case that a pre-annotated corpus is not available, one may adapt an Iterative and Incremental development strategy:

1. Annotate a portion of corpus
2. Use the annotated portion to develop an automatic tagger
3. Use the automatic tagger to annotate the rest of corpus
4. Go through automatically annotated data and correct its mistakes
5. Repeat from 2 until !!!

Manual, Automatic, or Semi-Automatic Annotation IV

The above mentioned process is also used, e.g., for the adaptation of automated taggers, i.e. to reduce errors in their output. For instance, an automatic part-of-speech tagger 'trained' using annotations from the Wall Street Journal text is not really useful for annotating informal English texts.

Then what are the cons?!

With regard to automatic tagging with no adaptation, erroneous output (in best case with a **systematic** error pattern) is unavoidable. In this case, analysis which are followed the tagging can be biased or in worst case, some interesting phenomena can be missed or even worst the result of analysis can be invalid all together.

Manual, Automatic, or Semi-Automatic Annotation V

But what is the problem of the iterative methods?

Simply, it can influence decisions that are made by annotators during the correction phase and therefore introduce bias during annotation, and thus lead to erroneous analysis.

Conclusion: be cautious, Automate Responsibly!

Quality Assessment for Manual Annotation Tasks I

How to assess the quality of manual annotations? is a question that is often pops up in corpus based studies. The simple answer is, perhaps, to annotate responsibly.

Likewise drinking responsibly motto that “the first thing you should do is avoid drinking alone, drink with people you know”, to annotate responsibly implies that the manual annotation task often is carried out by more than one person.

Often, a corpus is annotated by several annotators. Or, at least a portion of the corpus is annotated by at least two person. The asserted annotations in the shared portion are then checked against each other to measure what we call “annotation agreement” or “inter-annotator agreement” .

Quality Assessment for Manual Annotation Tasks II

In some scenarios, specially when the aim is to develop a so-called gold benchmark, if there are inconsistencies between annotators, they discuss them with each other and resolve them (whether it is disagreement or simple human error). In some other applications, resolving disagreement may not be feasible or desirable. In any case, a “measure of inter-annotator agreement” is used and it is reported for reporting the quality of annotations (e.g., how accurate are the annotators), or to indicate the difficulty of the task.

There are a number of methods to compute “measure of inter-annotator agreement”. In most of these methods, first we need to make a confusion matrix from the asserted annotations:

Quality Assessment for Manual Annotation Tasks III

Table: A confusion matrix (contingency table) for computing inter-annotator agreement

	Annotator 1: Yes	Annotator 1: NO	
Annotator 2: Yes	TN=50	FP=10	60
Annotator 2: No	FN=5	TP=100	105
	55	110	165

A naive way of comparing the annotators' agreement is to use measures such as Accuracy: Overall, how often these two annotators are in an agreement?

$$\blacktriangleright \text{Accuracy} = \frac{(TP+TN)}{\text{total}} = \frac{(100+50)}{165} \approx 0.91$$

Quality Assessment for Manual Annotation Tasks IV

The problem with accuracy is that we disregard a lot of other factors, such as chance!

A more elegant measure is the Cohen's Kappa κ measure:

$\kappa = \frac{p_o - p_e}{1 - p_e}$, in which

- ▶ p_o is the observed proportionate agreement, i.e.,

$$p_o = \frac{TN+TP}{total} \approx 0.91$$

- ▶ p_e is the probability of random agreement, i.e.,

$$p_e = p_e^{YES} + p_e^{NO}. \text{ In turn,}$$

- ▶ $p_e^{YES} = \frac{TN+FP}{TOTAL} \times \frac{TN+FN}{TOTAL}$, which in our case is $\frac{60}{165} \times \frac{55}{165} \approx 0.12$;

- ▶ $p_e^{NO} = \frac{FP+TP}{TOTAL} \times \frac{FN+TP}{TOTAL}$, which in our case is $\frac{110}{165} \times \frac{105}{165} \approx 0.42$;

- ▶ $p_e = 0.12 + 0.42 = 0.54$

Quality Assessment for Manual Annotation Tasks V

Putting all the numbers above together, in our simple example

$$\kappa = \frac{0.91 - 0.54}{1 - 0.54} = \frac{0.36}{0.45} \approx 0.80$$

Depending on the task, the computed κ score is interpreted differently. In certain annotation tasks, κ score greater than 0.30 is often considered as an acceptable measure. In general, $\kappa < .25$ interpreted as no agreement, $0.25 < \kappa < 0.45$ is an OK score, $0.6 < \kappa < 0.8$ is assumed to be a measure of substantial agreement, and anything more than 0.80 is said to be a perfect agreement.

The inter-annotator agreement is a challenging topic with renowned interest, particularly with the popularity of using social networks and mechanical turks for annotation tasks.

Quality Assessment for Manual Annotation Tasks VI

Evidently, the independence assumptions in Cohen's κ can be altered to have a measure which is better suited for an application.

Homework II: What is the Cohen's κ for this annotation task?

What is the inter annotator agreement in my corpus study?

Table: Contingency table for an annotation task

	Annotator 1: Yes	Annotator 1: NO	Annotator 1: Maybe
Annotator 2: Yes	yy=50	ny=10	my=10
Annotator 2: No	yn=5	nn=100	mn=10
Annotator 2: Maybe	ym=10	nm=10	mm=22

Tip: follow the instruction for computing p_e^{yes} , p_e^{no} in the previous slide and generalize it to p_e^{maybe} ; then use the κ definition.

Searching Corpora and Collecting Data I

So far, we have discussed how to build machine readable annotated corpora. As discussed, a major outcome of this step of process is the conversion of unstructured raw text data to a format that is some-how **structured** and can be read and processed by machines in an effective way. We define this structure using a mark-up language and/or by storing the raw data in structured files. We discussed that annotations can be used in order to explicitly mark and store implicit linguistics/extra-linguistics features of text so that it can be found without much of "effort" or even without the need for understanding the text.

As part of our activities, we all created our own toy corpus, which has markup and annotations for text segments, part-of-speech tags, and lemmas.

Searching Corpora and Collecting Data II

Once we have a corpus, e.g., our toy corpus, we would like to use it for answering some questions, e.g., what is the proportion of verbs and nouns in our toy corpus? How many copulas with lemma "be" exists in our corpus? What are the most common part-of-speech bi-grams in our toy corpus? And so on.

Luckily, we can use computers to answer these questions without much of efforts. Otherwise, we had to sit down and do this manually!!!

One of the most important outcomes of converting our corpora to structured data is that we can define or use a so-called meta language to search and retrieve the data that we are looking for. This meta language can be really simple and tell the computer to do a simple thing, e.g., implement a exact search functionality.

Searching Corpora and Collecting Data III

However, this meta-language can be more complicated so that it can be used to tell computers to run more sophisticated functions/operations on our corpus. For instance, the exact text search can be replaced by more powerful **Wildcard Searches**: Most text editors support search in which two 'wildcard characters' * and ? can be used to search and find strings that match combinations of characters and wildcards. Wildcard searches are not simply exact string matches, but are based on **character pattern matching** between the characters specified in a **query** and words in documents that we search.

- ▶ * matches zero or more non-space characters.
- ▶ ? matches exactly one non-space character
- ▶ ...

Searching Corpora and Collecting Data IV

For example, `hel*` will match any word starting with `hel`, such as `hell`, `help`, `hello`, and so on. On the other hand, `hel?` will only match four-letter words starting with `hel`, such as `help`, `hell`, and so on.

As many of you know, the notion of searching wildcard queries is generalized to what is known as matching/searching for regular expression patterns in text.

Similarly, if we have some well-defined structure for our corpus file, e.g., if we have them in XML format, or in a relational database, then we can use queries in the form of XSLT or SQL.

In a nutshell, structure of our corpus and its annotation and the meta-language that we can use to query it are the two sides of the

Searching Corpora and Collecting Data V

same "coin" and one affect the other. Development and using this "coin" requires skills which are often not possessed by users of corpora. Instead, likewise annotation tools, there exist tools which can be used to search and retrieve data from your corpus. Two things you need to know about these tools:

- ▶ These tools expect your corpus to have a specific format/structure. Therefore, to use one of them, you need to convert your corpus into the format which is defined/expected by the chosen tool.
- ▶ Each tool support/understand its own meta-language (aka. **query language**). Therefore, to use a tool, you need to learn the query language that the tool understand.

Searching Corpora and Collecting Data VI

We can categorize the search and retrieval tools similar to what we did for annotation tools. However, here it is probably more helpful to look at them by the type of annotation structure that they support (see listings from the last session).

Concordance Systems I

Concordancers are one of the tools that is often used in corpus linguistics. These are computer programs that search your corpus and generate **concordance** views or lists based on your input query.

Put simply, a concordance lists every instance of an entity (usually words) with its neighbouring context. The entity and context are defined by your query. The most simple (yet very powerful and helpful) example of a concordance is a word and its neighbouring words.

Concordance Systems II

Table: Example of a concordance for the word "corpora"

models trained on the Web parallel	corpora	in CUR . We conducted CLIR experiments
is the lack of large parallel	corpora	. In this paper we first describe
is based on a manually tagged	corpus	of Czech texts (mostly from
generated Chinese-English parallel	corpus	is used to train a probabilistic
probabilistic models from parallel	corpora	. Based on one of the statistical
approach is the lack of parallel	corpora	for model training . Only a few

As you will see, a concordance can be seen as a multifaceted index of results that are returned from your query.

There are several concordance systems, some of them are quiet old, e.g., the Wordsmith Tool. However, the IMS Open Corpus Workbench (CWB)⁶ (and further the introduction of the Corpus Query Language) is a noteworthy one. The main feature of the

Concordance Systems III

CWB and similar systems to it (such as the NoSke that we use in our course) is that with the help of CQL queries you can search and retrieve complex structures and patterns in a flexible manner which would not be that easy using other means such as a graphical use interface.

⁶For history, introduction, etc. see

Corpus query language (CQL) I

CQL is a query language. To learn CQL, we need to learn its syntax. [Kovar, 2017] has a very nice summary for basic CQL. Text in colour red is CQL. If your concordancer understands CQL, then the following queries have the following meaning for your system (concatenate "please retrieve ... " to the beginning of the text after "-").

Tokens and restrictions

- ▶ [] – any token
- ▶ [lemma="cat"] – all tokens where lemma is cat

Corpus query language (CQL) II

- ▶ `[tag="V.*"]` – all verbs (tokens whose tag matches V.*)
- ▶ `[tag!="V.*"]` – all **none** verbs (tokens whose tag is **not** V.*)

In general

- ▶ `[attribute="value"]`
- ▶ attributes: word, lemma, lc, lemma_lc, tag, ...

More restrictions

- ▶ `[lemma="help" & tag="N.*"]` – all occurrences of lemma “help”, as a noun
- ▶ `[lemma="help" — tag="N.*"]` – all occurrences of lemma “help” and all tokens tagged as noun

Corpus query language (CQL) III

- ▶ `[lemma="help—aid"]` – all occurrences of lemmas “help” and “aid”

More tokens

- ▶ `[tag="J.*"] [tag="N.*"]` – all adjective-noun bigrams

Optional tokens

- ▶ `[tag="J.*"] [tag="J.*"]? [tag="N.*"]` – one or two adjectives + noun
- ▶ `[tag="J.*"] [tag="J.*"]{0,3} [tag="N.*"]` – 1 to 4 adjectives + noun

Extending CQL

In theory and practice, you can extend the syntax of CQL to accommodate processes other than search and retrieval of concordance views. For this, you must study the user guides provided by the creator of the system you use. The most common ones are those which are found in SQL, that is count, distinct, etc. Note that the concordance system is particularly flexible: just see it as a text annotation friendly inverted index system.

Most concordance systems support a kind of application programming interface, i.e., if you want you can query the system pragmatically.

Corpus-based Problem Solving Methodology? I

Can we answer all questions using corpus based methods? You must be able to answer this question yourself. What we suggest is that they are useful and practical and can, at least, help some studies, e.g., to cross check the validity of a theory or the other way round to inspire a new one.

What is the common setup of a corpus-based analysis? The answer is known: the steps that are used in all empirical sciences. These are:

Corpus-based Problem Solving Methodology? II

- ▶ **Hypothesis formation** (an educated guess often triggered by an observation which cannot be explained precisely (e.g., the complexity of its behavioral system; you need a question. Hypotheses often are expressed in a natural language.)
- ▶ **Hypothesis formalization** and representation: you need to formalize your hypothesis in order to make it understandable for others and possibly machines (this step often involves translation of a natural language utterance to a "formal model", e.g., using set theory and numerical theory; in general the formalized model must bring some capability for reasoning and decision making.
- ▶ **Hypothesis testing/evaluation**: This involves using the tools that are provided by your formalized model.

Corpus-based Problem Solving Methodology? III

- ▶ Finally, **Translation**: translate your test to the original language; i.e., do some "reality" check (or, sometimes, cross validate) for your formal model and its answers – at this stage, I would like to refer you back to our discussion about "a priori" and a "posteriori". Two things are common: a) the result from the hypothesis testing is taken as evidence to verify the validity of your original hypothesis (note the model and its outcome cannot be used to reject the validity of your hypothesis — at least in my school of thought / philosophy). Other times, the model and the validity of its hypothesis are taken as granted, as philosophers say, we are a priori justified in believing what our model tells is the truth. By assuming that the model is "right", we start to judge about new

Corpus-based Problem Solving Methodology? IV

observations, which is the basis for many automated language analysis systems. We often define and build a new system to evaluate it. In many cases, the evaluation is based on an empirical approach and the evaluation system itself uses a corpus based method (so we need the full stack again ^_^).

Perhaps the list can be modified; specially the last two steps overlap a lot.

Example: Annotation Quality Assessment: We already did one complete round of these processes.

Corpus-based Problem Solving Methodology? V

- ▶ **Hypothesis formation:** Provided that we have some double annotations from annotators that can understand and follow our guidelines (it means from cooperative annotators; in other words, annotations are originated from Reliable Users), then we can have a measure of quality by validating and cross checking annotations from different annotators; the quality is the ratio of agreement between them.
- ▶ **Hypothesis formalization:** Let's use statistics, particularly a categorical model. The first step is to define a model, i.e., to define a contingency table (our intention is then to use tools for analysis of contingency tables to suggest or simply calculate a measure of quality).

Corpus-based Problem Solving Methodology? VI

- ▶ **Hypothesis testing/evaluation:** We choose Cohen's κ for this purpose (we may use other correlation/similarity measures for later studies, too).
- ▶ Finally, **Translation:** Now it is time to translate our Cohen's κ to an understandable utterance/answer, most likely in natural language, e.g., $\kappa = 0.35$. This can be tricky!

Quantitative Methods for Corpus Linguistics: Overview I

Quantitative and numerical analyses are ingrained in modern corpus linguistics in as much as one can perceive many corpus based research as the study of mathematical structures in corpora.

Let's emphasize that here our aim is no to provide a comprehensive and step-by-step guide to statistics and mathematical methods that are used in corpus based studies. Instead, we provide a set of examples from basic numerical methods, mainly to help you understand the general idea behind using these numerical methods for analysis of text data. A comprehensive picture of these

Quantitative Methods for Corpus Linguistics: Overview II

methods can be found in books that discuss statistics for corpus based linguistics, e.g.:

- ▶ Foundations of Statistical Natural Language Processing [Manning and Schütze, 1999]
- ▶ Speech and Language Processing [Jurafsky and Martin, 2000]

Quantitative Methods for Corpus Linguistics:

Using Simple and Proportional Counts, and Test of Significance |

- ▶ **Using simple frequency counts:** Perhaps the most straightforward quantitative analysis of a corpus is to do some simple frequency counts of words or their classes. Finding the most frequent "word" or "part-of-speech category" is an example that we worked on earlier.
Put simply, a basic method for analyzing this data consists of forming a list of linguistic structures (e.g., words) alongside their frequencies (we call this list a one-dimensional frequency table), e.g.:

Quantitative Methods for Corpus Linguistics:

Using Simple and Proportional Counts, and Test of Significance II

Table: A one dimensional frequency table

Word	freq.
this	401,239
that	460,051
the	656,3782
is	780,000
...	...

Quantitative Methods for Corpus Linguistics:

Using Simple and Proportional Counts, and Test of Significance III

We then use a baseline (e.g., random or some other frequency counts from another corpus) to compute expected frequencies. We then compare the expected counts with the observed ones, e.g., to see their deviation. We go through details in the following slides.

Quantitative Methods for Corpus Linguistics:

Using Simple and Proportional Counts, and Test of Significance IV

- ▶ **Working with Proportions:** Although simple frequency counts can be helpful in some applications, they are inherently disadvantageous when we wish to compare our observations in one corpus (sub corpus) to another one. That is to say, simple counts do not provide information about the prevalence of a count (e.g., a token) with respect to the total counts (e.g., the size of corpus). Obviously, this is problematic when we want to compare our frequency counts across corpora of different sizes. Therefore, we often **normalize** simple frequency counts as a percentage of the total number of observations. For instance, if we are working on token counts, we can divide

Quantitative Methods for Corpus Linguistics:

Using Simple and Proportional Counts, and Test of Significance V

them by the total number of tokens in our corpus and thus normalize them to the percentage of their occurrences in our corpus. Given that the acquired proportional numbers are often small, we often **scale up** our percentages using a method, in its simplest form we can multiply these proportions to a constant value (e.g., to 1,000 to have per mille or 1,000,000 to have proportion in parts per million). However, in certain applications, we are supposed to adapt a more sophisticated scaling problem, e.g., to use logarithmic scales, to avoid errors due to numerical analysis (see scaling methods

Quantitative Methods for Corpus Linguistics:

Using Simple and Proportional Counts, and Test of Significance VI

for numerical analysis). Normalized counts are often reported together with a **statistical dispersion** measure.

- ▶ **Test of Significance:** Sometimes we are interested to have a comparison of statistics that we have collected in a study and assign a degree of confidence or certainty to our findings from this comparison. [McEnery and Wilson, 2001] provide an example of the usage of the verb form "dicit" (*to say*) in the *Gospel of Matthew* and the *Gospel of John*; particularly, how often the present (i.e., *dicit*) and the perfect (i.e., *dixit*) tenses of this verb are used in these corpora.

Quantitative Methods for Corpus Linguistics:

Using Simple and Proportional Counts, and Test of Significance VII

	dicit	dixit
Gospel of Matthew	46	107
Gospel of John	118	119

Table: Usages of DICIT verb in Gospel of Matthew and John

Quantitative Methods for Corpus Linguistics:

Using Simple and Proportional Counts, and Test of Significance VIII

At first glance, we can say that John uses the present form proportionally more often than Matthew. But how confident are we about this conclusion? One way to answer this question is to use a **statistical significance test**, i.e., to determine the effect of chance in our answer and collected frequencies. There are numerous significance tests that can be/are used, each has its own merits and drawbacks. The Chi squared test (χ^2) is a well-known example. Put simply,

Quantitative Methods for Corpus Linguistics:

Using Simple and Proportional Counts, and Test of Significance IX

likewise many other tests, the χ^2 test compares the difference between the observed frequencies and the expected ones, i.e.:

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^c \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

Above, e_{ij} s stand for the **Expected Frequencies**, which are calculated by the estimated probabilities for each category (remember the discussion we had on the IAA computation):

Quantitative Methods for Corpus Linguistics:

Using Simple and Proportional Counts, and Test of Significance X

First find the sum of rows (i.e., t_{11} and t_{21}) and columns (i.e., t_{21} and t_{22}) as well as the total sum of counted observations (i.e., t).

	dicit	dixit	
Gospel of Matthew	46	107	$t_{11} = 46 + 107 = 153$
Gospel of John	118	119	$t_{21} = 118 + 119 = 237$
	$t_{12} = 46 + 118 = 164$	$t_{22} = 107 + 119 = 226$	$t = 46 + 107 + 118 + 119 = 390$

Table: Computing expected frequencies (1): first compute the required sums of observed counts.

Quantitative Methods for Corpus Linguistics:

Using Simple and Proportional Counts, and Test of Significance XI

Now use t_{ij} s and t to compute probabilities of the observed values (similar to what we did for P_e in the IAA computation), e.g., $P_{dicit} = \frac{t_{12}}{t} = \frac{164}{390}$, and so on. Now, to compute the expected count for, let's say 'dicit' in the Gospel of Metthew, simply multiply the probability P_{dicit} to the total number of observations in the Gospel of Metthew, i.e., $\frac{164}{390} \times 153$. The summary formula for the expected counts in each cell is given below:

Quantitative Methods for Corpus Linguistics:

Using Simple and Proportional Counts, and Test of Significance XII

	dicit	dixit
Gospel of Matthew	$e_{11} = \frac{t_{11} \times t_{12}}{t}$	$e_{12} = \frac{t_{11} \times t_{22}}{t}$
Gospel of John	$e_{21} = \frac{t_{21} \times t_{12}}{t}$	$e_{22} = \frac{t_{21} \times t_{22}}{t}$

Table: Computing expected frequencies: multiply the obtained probabilities to the respective sums of observed values for each cell (formula summary).

Quantitative Methods for Corpus Linguistics:

Using Simple and Proportional Counts, and Test of Significance XIII

If the difference between the expected and the observed frequencies is small, the chance that the observed frequencies are a result of chance is higher. The computed χ^2 can be compared against a reference table in order to decide how *significant* is our result. To do so, we need an additional value called the **degree of freedom** (usually abbreviated as d.f.). D.f. is simply (number of col in the freq. table - 1) \times (number of rows in the freq. table - 1). In our example, the d.f. is 1. Now we have complete data to look into the χ^2 probability table. We find the relevant d.f. row and find the value that is closest to our compute χ^2 and read the assigned probability

Quantitative Methods for Corpus Linguistics:

Using Simple and Proportional Counts, and Test of Significance XIV

for that column. A probability close to 0 means that the difference is significant, i.e., very unlikely to be a result of chance, and vice versa, the probability close to 1 means that our observation is certainly due to the chance. In our example, the $\chi^2 = 14.8432$ (d.f.=1), which gives us the p -value of approximately 0.0001, in turn, we can claim that the result is significant at $p < 0.05$ (or we are more than 95% confident about our hypothesis about John and Matthew's use of different tenses of dicit — in this case, we are in-fact more than 99% confident).

Quantitative Methods for Corpus Linguistics:

Using Simple and Proportional Counts, and Test of Significance XV

Table: Chi-Square Distribution Table

df	Probability (p)									
	0.995	0.99	0.975	0.95	0.90	0.10	0.05	0.025	0.01	0.005
1	—	—	0.001	0.004	0.016	2.706	3.841	5.024	6.635	7.879
2	0.010	0.020	0.051	0.103	0.211	4.605	5.991	7.378	9.210	10.597
3	0.072	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.345	12.838
4	0.207	0.297	0.484	0.711	1.064	7.779	9.488	11.143	13.277	14.860
5	0.412	0.554	0.831	1.145	1.610	9.236	11.070	12.833	15.086	16.750
6	0.676	0.872	1.237	1.635	2.204	10.645	12.592	14.449	16.812	18.548
7	0.989	1.239	1.690	2.167	2.833	12.017	14.067	16.013	18.475	20.278

Quantitative Methods for Corpus Linguistics:

Using Simple and Proportional Counts, and Test of Significance XVI

Note that using the χ^2 test is reliable if only you respect assumptions behind the test. This is very important (e.g., χ^2 won't work for proportional counts and when the counted observations are small) (See http://www.basic.northwestern.edu/statguidefiles/gf-dist_ass_viol.html for a short summary as well [McEney and Wilson, 2001] for additional information). Depending on your problem, other tests of significance (such as t -test, z -test, etc.) can replace χ^2 .

The “null hypothesis”: Using a more precise mathematical term, here, we use χ^2 as a tool for null hypothesis testing. In statistics,

Quantitative Methods for Corpus Linguistics:

Using Simple and Proportional Counts, and Test of Significance XVII

"null hypothesis" is a general term to state that there is no relationship between two measured phenomena, i.e., to say that there no association among groups of observations (or simply, independence). Statistical tests such as χ^2 provide a procedural method and criteria for rejecting a null hypothesis (e.g., as used in dicit and dixit example). In mathematical terms, rejecting a null hypothesis is used as a basis for believing that there is a relationship between two phenomena.

This being said, we can use the following general steps for solving problems using the test of significance:

Quantitative Methods for Corpus Linguistics:

Using Simple and Proportional Counts, and Test of Significance XVIII

- ▶ Describe the Research Hypothesis
- ▶ Translate the Research Hypothesis into a Null Hypothesis (there is no relationship between the two variables dicit and dixit in the two Gospels)
- ▶ Choose an appropriate error level (i.e., p value, often $p = 0.05$)
- ▶ Compute the test for statistical significance (we used χ^2 , which had the value 14.84 and according to that and $p = .05$, the null hypothesis was correct)

Quantitative Methods for Corpus Linguistics:

Using Simple and Proportional Counts, and Test of Significance XIX

- ▶ Based on the obtained numerical values interpret the results (there is no meaningful correlation between the usage of different tenses of *dicit* in John and Matthew's gospels, in other word, John and Matthew use different tenses of the verb "*dicit*")

Quantitative Methods for Corpus Linguistics:

Collocations and Measurement of Collocational Strength |

The notion of collocation has been a major topic in corpus linguistics. Given the broad meaning that it bears in corpus linguistics (collocation=co-occurrence, right?!), unfortunately, we cannot give a definite answer to the question of "what a collocation is". In a sense (and in simple language), collocations are sequence of words (or in general, linguistic entities) that co-occur with each other more than it is expected by chance. The simplest example of collocations are phrases and contiguous

Quantitative Methods for Corpus Linguistics:

Collocations and Measurement of Collocational Strength II

sequences of tokens such “fast food”, “nice day”, “New York”, “far away”, and so on.

However, non-contiguous sequences can be also analyzed as collocations, e.g., pair of verbs and their subjects.

In general, the notion of collocation“ can be extended with respect to the “context” from which co-occurrences are collected from. For instance, the context can be limited to adjacent words (such as bi-grams in the following examples) or the context can be documents, i.e., we can count word co-occurrences within

Quantitative Methods for Corpus Linguistics:

Collocations and Measurement of Collocational Strength III

documents (such as used in information retrieval systems); even, the context can be structured, e.g., we can count co-occurrences between words that are in a particular grammatical relationship (as in the above example for the non-contiguous collocations).

Defining a notion of **collocational strength** and the identification of collocations, thus, can be helpful in many applications. For instance, **multiword expressions**, such as idioms and terminologies, are collocations that are interesting for linguistic investigations given that their characteristics are often not predictable from characteristics of their constituent words.

Quantitative Methods for Corpus Linguistics:

Collocations and Measurement of Collocational Strength IV

To identify collocations and to quantify collocation strength, we use **association measures**. For example, in this context, the χ^2 can be used as an association measure (more popular choices are **mutual information**, z-score, ϕ score, etc.). In this case, we expect that word sequences with strong collocational affinity are assigned to large association values (such as χ^2). To make sense of what is stated here, let's compute the collocational strength for the two bi-grams of "far away" and "far for" using the χ^2 measure. Evidently, "far away" is in fact an English collocation and thus we expect the χ^2 for it will be larger than "far for" in a representative corpus for English such as BNC.

Quantitative Methods for Corpus Linguistics:

Collocations and Measurement of Collocational Strength V

To compute the collocational strength between a pair of words such as "far away" we first need to collect sufficient statistical information, as shown in the following table.

Table: Contingency tables for far+away from BNC corpus

	away	\neg away
far	1,239	36,345
\neg far	46,051	112,214,788

Quantitative Methods for Corpus Linguistics:

Collocations and Measurement of Collocational Strength VI

The key is the assigned labels to the rows and columns of this table, i.e., the type of statistical information we collect from our corpus (note its difference to Table 5 that we used for the null hypothesis testing). In this table, \neg is the logical complement (the negation marker — for instance, the cell for the row \neg far and the column away $(\neg\text{far},\text{away})=46,051$ gives the number of occurrences of all the bi-grams in which the second word is "away" but the first word is **not** "far"). Replacing the numbers from the above table to the χ^2 formula, we arrive to $\chi^2(\text{far} + \text{away}) = 94526$

For instance, the computation can be done using **R**:

Quantitative Methods for Corpus Linguistics:

Collocations and Measurement of Collocational Strength VII

```
1 > mat=matrix(c(1239, 46051,36345, 112214788)
  ,ncol=2, nrow=2) # initialize a 2x2 matrix
  (note the order of numbers)
2 > mat # print the matrix
3      [,1]      [,2]
4 [1,]  1239      36345
5 [2,] 46051 112214788
6
7 > chisq.test(mat) # compute the X-square
8 X-squared = 94526, df = 1, p-value < 2.2e-16
```

Quantitative Methods for Corpus Linguistics:

Collocations and Measurement of Collocational Strength VIII

if we repeat the above process for the bigram "far+for"

Table: Contingency tables for far+for from BNC corpus

	for	¬ for
far	82	37,502
¬ far	830,086	111,430,753

```
1 > mat=matrix(c(82, 830086,37502, 111430753) ,ncol=2, nrow=2)
2 > chisq.test(mat)
3 X-squared = 138.41, df = 1, p-value < 2.2e-16
```

Quantitative Methods for Corpus Linguistics:

Collocations and Measurement of Collocational Strength IX

As expected, the χ^2 value for the habitual bigram of "far+away" is much higher than the computed χ^2 for "far+for" (i.e., $94526 > 138.41$).

As mentioned, we can choose association measures other than χ^2 . For example, below is the *R* script for computing mutual information (MI) for "far+away" and "far+for". As shown, the MI measure also confirms suggests that the collocation strength for "far+away" is much larger than for "far+for".

Quantitative Methods for Corpus Linguistics:

Collocations and Measurement of Collocational Strength X

```
1 install.packages(c("entropy")) #if you have not installed it
2 > mat=matrix(c(1239, 46051,36345, 112214788) ,ncol=2, nrow=2)
3 > library("entropy") # load the entropy library
4 > mi.plugin(mat)
5 [1] 3.753742e-05
6 > mat=matrix(c(82, 830086,37502, 111430753) ,ncol=2, nrow=2)
7 > mi.plugin(mat)
8 [1] 8.576249e-07
9 > 3.753742e-05 > 8.576249e-07
10 [1] TRUE
11 >
```

Similarly, we can repeat the process for other bigrams such as "far+from" (which we know is idiom):

Quantitative Methods for Corpus Linguistics:

Collocations and Measurement of Collocational Strength XI

Table: Contingency tables for far+from from BNC corpus

	from	¬ from
far	3222	34362
¬ far	406241	111854598

```
1 > mat=matrix(c(3222, 406241 ,34362 , 111854598),ncol=2, nrow =2)
2 > chisq.test(mat)
3 X-squared = 69702, df = 1, p-value < 2.2e-16
4 > mi.plugin(mat)
5 [1] 6.438995e-05
```

Quantitative Methods for Corpus Linguistics:

Collocations and Measurement of Collocational Strength XII

What about "far+away" vs "far+from"? As seen, $\chi^2(\text{far+away}) > \chi^2(\text{far+from})$ but $mi(\text{far+away}) < mi(\text{far+from})$. Does this mean MI is a better choice than χ^2 (as recommended in many publications) for measuring collocation strength?

In real world applications, we often compute collocation strength for a list of candidates, and the computed measure is used to sort and manipulate the output into a set of groups or clusters. For instance, for terminology extraction, we often compute collocation strength for all noun phrases in the corpus. We choose a threshold δ : all noun phrases with collocational strength greater than δ are

Quantitative Methods for Corpus Linguistics:

Collocations and Measurement of Collocational Strength XIII

assumed to be terms (keywords) and the remaining to be none-terms (i.e., the noun phrases are grouped/classified/clustered into two categories).⁷

⁷Obviously this is an overly simplified solution. > < ☰ > < ☰ > ☰ 🔍 ↻

Quantitative Methods for Corpus Linguistics:

Three-way contingency tables (Multivariate analysis) I

Previously, we look into calculating collocation strength for bigrams. What about collocations of longer length? for example, "far from it", "far from the madding crowd", etc. How to compute a collocational strength for these cases?

Contingency tables for sequences of length n ($n > 2$) tokens are a little more complicated to form and visualize (since they are n -dimensional; i.e., we need to deal with superordinate columns).

Quantitative Methods for Corpus Linguistics:

Three-way contingency tables (Multivariate analysis) II

Below is one way to write down a contingency table for computing collocation strength for trigram $x+y+z$:

Table: An example of a contingency table for a trigram $x+y+z$

		z	$\neg z$
x	y	f_{111}	f_{110}
x	$\neg y$	f_{101}	f_{100}
$\neg x$	y	f_{011}	f_{010}
$\neg x$	$\neg y$	f_{001}	f_{000}

Quantitative Methods for Corpus Linguistics:

Three-way contingency tables (Multivariate analysis) III

Alternatively, you the above table can be split into two two-dimensional tables, one for keep tracking the count of tri-grams that end with z and another for $\neg z$ (known as partial tables):

Quantitative Methods for Corpus Linguistics:

Three-way contingency tables (Multivariate analysis) IV

Table: ends with z, i.e., for f_{**1}

	y	$\neg y$	
x	f_{111}	f_{101}	t_{1+1}
$\neg x$	f_{011}	f_{001}	t_{0+1}
	t_{+11}	t_{+01}	t_{++1}

Table: ends with $\neg z$, i.e., for f_{**0}

	y	$\neg y$	
x	f_{110}	f_{100}	t_{1+0}
$\neg x$	f_{010}	f_{000}	t_{0+0}
	t_{+10}	t_{+00}	t_{++0}

and, respectively, defining values such as $t_{1++} = t_{1+0} + t_{1+1}$ and so on (note the pattern in indices), we can compute expected/estimated counts as follows:

Quantitative Methods for Corpus Linguistics:

Three-way contingency tables (Multivariate analysis) V

$$e_{111} = \frac{t_{1++} \times t_{+1+} \times t_{++1}}{t_{+++}^2}$$

$$e_{110} = \frac{t_{1++} \times t_{+1+} \times t_{++0}}{t_{+++}^2}$$

$$e_{100} = \frac{t_{1++} \times t_{+0+} \times t_{++0}}{t_{+++}^2}$$

...

$$e_{010} = \frac{t_{0++} \times t_{+1+} \times t_{++0}}{t_{+++}^2}$$

$$e_{001} = \frac{t_{0++} \times t_{+0+} \times t_{++1}}{t_{+++}^2}$$

Again please pay attention to patterns in the indices for e_{xxx} s and t_{xxx} s.

Quantitative Methods for Corpus Linguistics:

Three-way contingency tables (Multivariate analysis) VI

Given f s and e s, now we can compute the True Mutual Information (TMI) weight [Manning and Schütze, 1999] for each trigram using the formula [Lyse and Andersen, 2012]:

$$TMI = \sum_{ijk} \frac{f_{ijk}}{t_{+++}} \times \ln\left(\frac{f_{ijk}}{e_{ijk}}\right).$$

Marginals for the two-way partial tables (i.e., t s) can be expressed as conditional proportions, similar to the simple conditional proportion in the two-way case, e.g., $p_{ij|k} = \frac{n_{ijk}}{t_{++k}}$.

Quantitative Methods for Corpus Linguistics:

Three-way contingency tables (Multivariate analysis) VII

This three-way table can be used for extracting collocational bi-grams in which we can assert the effect of an additional **feature**/condition when computing an association measure. You can read more on this under topic "Multivariate analysis".

Evidently, the idea behind the three-way contingency tables can be extended to n -way tables. These n -way tables are used in "multivariate/multifactorial" analysis in which we take into account the interdependence relationships between variables, e.g., to assert information about syntactic conditions that rule over word co-occurrences, and/or to consider **latent (e.g., semantic) variables**.

Quantitative Methods for Corpus Linguistics:

Other applications of collocation measures I

The techniques introduced for extracting habitual/idiomatic n-gram English collocations can be used in other applications, too. [McEnery and Wilson, 2001] suggest two (among others):

- ▶ Word Sense Discrimination (word sense disambiguation and induction): Let's assume we want to find different meanings of a word w (e.g., book). In this context, we can look for all words in the corpus that show a strong collocational association to the word w , e.g., for $w = \text{book}$, we will arrive to

Quantitative Methods for Corpus Linguistics:

Other applications of collocation measures II

a list such as *Collocations* = { read, resort, hotel, science, school, ... }. Next, by grouping words in *Collocations*, we can grasp an idea about different meanings of w (as you can see intuitively from this list for the word book).

- ▶ Translation: If we have a corpus of sentence-aligned translations, then the proposed association measures can be used for further alignment (translation) of words in these sentences.

Quantitative Methods for Corpus Linguistics:

Other applications of collocation measures III

Similar applications such as the above listed ones can be found under the topic of **discriminatory feature extraction** in pattern recognition and machine learning text books.

Quantitative Methods for Corpus Linguistics:

Other frequently used techniques |

You may have heard (or will hear often) about techniques such as

- ▶ Matrix Factorization, e.g., Principal Component Analysis (PCA), Singular Value Decomposition (SVD), t-Distributed Stochastic Neighbor Embedding (t-SNE), etc.
 - ▶ These methods are often used for visualization of word collocations, to perform dimension reduction, to build distributed representation of words (i.e., word vectors, aka. word embeddings), etc.

Quantitative Methods for Corpus Linguistics:

Other frequently used techniques II

- ▶ Log-linear analysis, a technique which is used for finding relationship between more than two categorical variables, particularly, to find most important features (or their contribution), etc.
- ▶ Probabilistic language modeling (and related techniques such as Hidden Markov Models, Expectation Maximization, etc.).
- ▶ And many many other methods ...

Quantitative Methods for Corpus Linguistics:

Other frequently used techniques III

Once the suitable contingency table for a problem is devised, the above mentioned methods can be applied to the cross-tabulated data (e.g., with a few lines of codes in an environment such as R). How these methods work and mathematical principles behind them are discussed in other CL courses with focus of statistics.

Quantitative Methods for Corpus Linguistics:

Examining relations between many variables (Example) I

Here, we have an an example other than the use of significant tests. Let's look at the distribution of modal verbs (**can, could, may, might, shall, should**) in different text genres in the Susanne corpus, which are (specified in doc.id structural attributes):

A : press reportage

G : belles lettres, biography, memoirs

J : learned (mainly scientific and technical) writing

N : adventure and Western fiction

Quantitative Methods for Corpus Linguistics:

Examining relations between many variables (Example) II

Table: Contingency Table of modal verbs across genres

	Genre			
modal	A	G	J	N
can	33	79	76	29
could	29	42	23	125
may	16	58	48	1
might	10	21	15	31
shall	1	5	1	0
should	36	27	53	10

Quantitative Methods for Corpus Linguistics:

Examining relations between many variables (Example) III

To investigate distributional/statistical similarities between these modal pairs (aka **variables**) in different genres (aka **samples**), e.g., to find the most similar pairs of modal verbs, we can use an **intercorrelation matrix** (aka similarity matrix):

Table: Inter-correlation Matrix: What do you reckon?

	can	could	may	might	shall	should
can	1.0	-0.582	0.979	-0.205	0.694	0.555
could	-0.582	1.0	-0.684	0.914	-0.384	-0.881
may	0.979	-0.684	1.0	-0.33	0.78	0.571
might	-0.205	0.914	-0.33	1.0	-0.079	-0.81
shall	0.694	-0.384	0.78	-0.079	1.0	0.029
should	0.555	-0.881	0.571	-0.81	0.029	1.0

Quantitative Methods for Corpus Linguistics:

Examining relations between many variables (Example) IV

Before building the inter-correlation matrix, we could apply some statistical method to cancel the effect of noise (chance and other things...). For example, we replace the raw frequencies in cells with the Odds Ratio (association strength):

Table: Odds-Ratio-weighted Contingency Table

	Genre			
modal	A	G	J	N
can	0.0	0.187904096765896	0.22064854856735056	0.0
could	0.0	0.0	0.0	0.806218317760927
may	0.0	0.44661225301333585	0.3288292173569525	0.0
might	0.0	0.0	0.0	0.4571580929065891
shall	0.0	0.861881361218121	0.0	0.0
should	0.5640142637079749	0.0	0.40382256842212283	0.0

Quantitative Methods for Corpus Linguistics:

Examining relations between many variables (Example) \checkmark

Now, we can compute correlations based on this weighted contingency table:

Table: Inter-correlation matrix for the odds-ratio-weighted table.

	can	could	may	might	shall	should
can	1.0	-0.574	0.948	-0.574	0.482	-0.095
could	-0.574	1.0	-0.564	1.0	-0.333	-0.562
may	0.948	-0.564	1.0	-0.564	0.736	-0.278
might	-0.574	1.0	-0.564	1.0	-0.333	-0.562
shall	0.482	-0.333	0.736	-0.333	1.0	-0.562
should	-0.095	-0.562	-0.278	-0.562	-0.562	1.0

Quantitative Methods for Corpus Linguistics:

Examining relations between many variables (Example) VI

What is the effect of weighting using an association measure such as Odds Ratio?

Here, the values in the inter-correlations matrix are computed using Pearson's r ; however, other correlation measures can be used, too.

Also, note that that you can imagine the transposed contingency table and interpret your analysis from a new perspective (e.g., what are the most similar genres).

Given inter-correlation matrix (or, similarly, equivalently a distance matrix), we can a **clustering technique** to reduce the number of our variables (i.e., our modal verbs) into a **group** of modal verbs.

Quantitative Methods for Corpus Linguistics:

Examining relations between many variables (Example) VII

Modal verbs in each group are similar in a sense (remember, sometimes, finding the common similar feature/sense between items in the group is not easy). Following are clusterings and their visualization (**heat map** and **dendrogram**) for our example. **We do this for both modal verbs and text genres.**

Quantitative Methods for Corpus Linguistics:

Clustering Visualization I

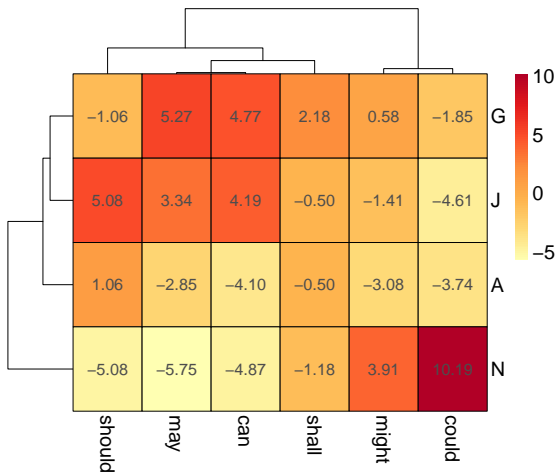


Figure: Single linkage clustering based on Pearson's r correlations.

Quantitative Methods for Corpus Linguistics:

Clustering Visualization II

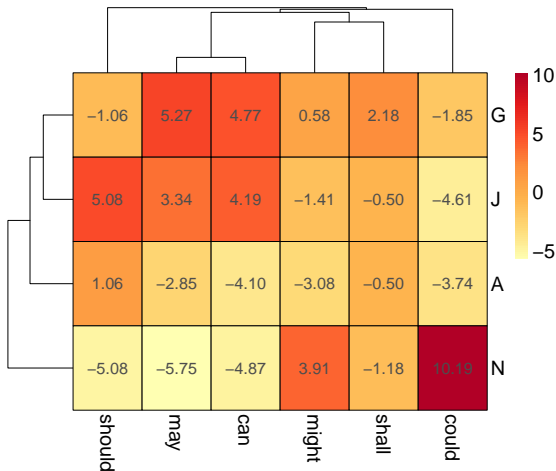


Figure: Single linkage clustering based on Euclidean distances.

Quantitative Methods for Corpus Linguistics:

Clustering Visualization III

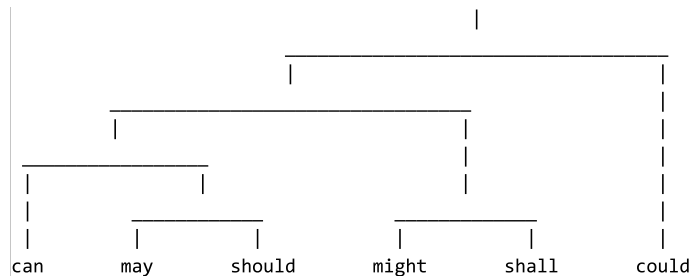


Figure: A simple dendrogram. Can this result imply some relation between modal verbs' tense and the genres of text (in the Susanne corpus)?!

Quantitative Methods for Corpus Linguistics: Exploratory Techniques Side Note

Side note: In the corpus linguistics community, methods such as Hypothesis testing and Clustering are often known as Exploratory Techniques.

Quantitative Methods for Corpus Linguistics:

Classification and Clustering I

There is a constantly increasing interact between computational linguistics and machine learning (e.g., see research published by the Association for Computational Linguistics). Almost all machine learning techniques have been applied to computational linguistics, and vice versa, the challenge of understanding human language (as one of the main goals of artificial intelligence) has constantly influenced research in machine learning.

But what is classification?

Quantitative Methods for Corpus Linguistics:

Classification and Clustering II

A **class** is a set of entities that can be identified by characteristics that all its members share, e.g., words such as can, should, must, shall can be of class modal verb.

Classification is the task of automatic assignment of entities to classes. However, if the classes are not known prior to the assignment task, then the task is called **clustering** (as discussed in the previous session).

Clustering is therefore the task of grouping entities by their mutual characteristics in such a way that the members of a group, called a

Quantitative Methods for Corpus Linguistics:

Classification and Clustering III

cluster, are more similar to each other than to the members of other clusters in a sense. The classification task is usually referred to as **supervised learning**, whereas the clustering task is known as **unsupervised learning**.

Familiar examples of such tasks are document classification and clustering. For example, documents can be categorized by their subject areas. In this example, if the subject areas are known beforehand—for example, the subject areas are limited to science and art—the task is called document classification. However, if the subject areas are not known beforehand, then the task is called

Quantitative Methods for Corpus Linguistics:

Classification and Clustering IV

document clustering and it organizes the documents, for this given example, into groups that give a sense of the subject areas (it will be all about what you count in your contingency table, e.g., documents can be classified, instead of by subjects area, by their relatedness, style, theme, sentiment, author characteristics, etc).

A classification task—that is, supervised learning—can be formalised by a mapping function f . For a feature space V (i.e., your count table) and an output space L made of a finite set of category labels l , the classification process is given by $f : V \mapsto L$.

Quantitative Methods for Corpus Linguistics:

Classification and Clustering V

The mapping function f is **learned** by a machine learning algorithm during a process called **training**. The training process chooses a function that **best** estimates the relationship between the input feature values (vectors) and the output labels from a given set of instances $T \in V \times L$, which is called the **training dataset**.

If $L = \mathbb{R}$, then the classification task is called regression. For $|L| = 2$, the task is called **binary** classification. If $|L| > 2$, then the task is called **multi-class** or **multi-way** classification. In a clustering task—that is, unsupervised learning—the T and L are not presented explicitly. Instead, criteria—such as the cardinality of L ,

Quantitative Methods for Corpus Linguistics:

Classification and Clustering VI

the way similarities are compared, and a relationship between members of clusters—are given.

These learning algorithms are the subject of vibrant scientific research in a framework known as **statistical learning theory**. The comprehensive study of these methods, therefore, requires dedicated course. Here we just scratch the surface.

Quantitative Methods for Corpus Linguistics:

Classification and Clustering VII

Table: Example of training data: annotations as class labels

modal	A	G	J	N	CLASS LABEL
can	33	79	76	29	Present
could	29	42	23	125	Past/preterite
may	16	58	48	1	Present
might	10	21	15	31	Past/preterite
shall	1	5	1	0	Present
should	36	27	53	10	Past/preterite

Quantitative Methods for Corpus Linguistics:

Classification and Clustering VIII

In statistical learning theory, **learning procedure**, itself, is formalised using a mapping function $(V \times L)^n \mapsto \mathcal{F}$. In this definition, \mathcal{F} , which is called the hypothesis space, is a space of functions $f_m : V \mapsto L$, where V and L are the input feature space and the output label space, respectively. The learning algorithm searches in \mathcal{F} for a function that best approximates the relationship implied between the vectors and the labels by the set of n samples from $(V \times L)^n$.

This formalisation is based on two assumptions.

Quantitative Methods for Corpus Linguistics:

Classification and Clustering IX

- ▶ First, it is assumed that the data is being classified, that is, the set of n tuples $\langle \vec{v}, l \rangle$, are drawn independently and identically from a fixed but unknown joint probability distribution $p(\vec{v}, l)$.
- ▶ Second, in order to assess the quality of learning, it is assumed that there is a notion of **loss** or error that can determine, for a given input vector, the discrepancy between the expected label and the label predicted by a f_m . This is indicated by a **Loss function** $loss : L \times L \mapsto \mathbb{R}$. For a given vector \vec{v} and the expected label l , $Loss(l, f_m(\vec{v}))$ gives the error of f_m .

Quantitative Methods for Corpus Linguistics:

Classification and Clustering X

By these assumptions, the goal of the learning process is to find a $f_0 \in \mathcal{F}$ that minimises the average error. For $f \in \mathcal{F}$, the average error, which is also called the **risk** of f $R(f)$ is given by:

$$R(f) = \int_{V \times L} \text{Loss}(l, f(\vec{v})) dp(\vec{v}, l). \quad (1)$$

However, $R(f)$ cannot be computed directly because the probability distribution $p(\vec{v}, l)$ is unknown. The learning problem formalised above can be solved using a variety of approaches.

Quantitative Methods for Corpus Linguistics:

Classification and Clustering XI

In the probability-based approaches, two major methods to approximate $R(f)$ can be recognised. In the first group of methods, it is assumed that the type of the distribution of data is known; thus, a probability model with a number of fixed parameters can be used to estimate $p(\vec{v}, l)$. Consequently, the training dataset T is used to estimate the value of the model's parameters. For instance, assuming the data has a Gaussian distribution, the joint probability is estimated using the mean and variance of the data samples in T . The familiar algorithm in this group is the naïve Bayes classifier.

Quantitative Methods for Corpus Linguistics:

Classification and Clustering XII

The second group of probability-based methods, in contrast to the former methods, do not assume prior knowledge of the type of data distribution. These techniques estimate $p(\vec{v}, l)$ by the observation of the data samples provided in T . E.g., the latent Dirichlet allocation for uncovering **topic models** is a well-known example of these methods. Both category of methods listed above can exploit the learned joint distribution in a reverse fashion; that is, given a class label l , they can synthesise examples of context elements related to l . Hence, the probability-based methods are often known as **generative** approaches.

Quantitative Methods for Corpus Linguistics:

Classification and Clustering XIII

On the other side, one category of learning techniques—often named as **discriminative** methods—bypasses the probability estimation and approximates $R(f)$ directly. A subcategory of these methods adopt a **geometric approach** in the sense that they reformulate a learning task as the construction of decision boundaries in a metric space. The support vector machine algorithm and the k -nearest-neighbours technique are the familiar examples in this category. These methods approximate $R(f)$ from the training set T using an **induction principle** such as **empirical**

Quantitative Methods for Corpus Linguistics:

Classification and Clustering XIV

risk minimisation (ERM). Given n samples $\langle \vec{v}_i, l_i \rangle$ in T , the **empirical risk of function** f over T is given by:

$$R_{\text{emp}}(f) = \frac{1}{n} \sum_{i=1}^n \text{Loss}(f(\mathbf{v}_i), l_i). \quad (2)$$

It is expected that the function f that has a small empirical risk (i.e., $R_{\text{emp}}(f)$) will also have a small risk (i.e., $R(f)$). It is proved that for f of **finite complexity**, $R_{\text{emp}}(f)$ converges to $R(f)$ when $n \rightarrow \infty$. Therefore, it is assumed that the goal of a learning task

Quantitative Methods for Corpus Linguistics:

Classification and Clustering XV

can be achieved—that is, finding the $f_o \in \mathcal{F}$ that minimises the risk $R(f)$ —by finding the f_o that minimises the empirical risk $R_{\text{emp}}(f)$:

$$f_o = \arg \min_{f \in \mathcal{F}} R_{\text{emp}}(f) = \arg \min_{f \in \mathcal{F}} \left(\frac{1}{n} \sum_{i=1}^n \ell(f(\mathbf{v}_i), l_i) \right). \quad (3)$$

Accordingly, $R_{\text{emp}}(f)$ is employed as a quantifiable method for the assessment of the **generalisation** ability of f_o —that is, it is assumed that if f_o has a small $R_{\text{emp}}(f)$, then it also has a high generalisation ability.⁸ Whereas research in machine learning

Quantitative Methods for Corpus Linguistics:

Classification and Clustering XVI

investigates developing algorithms by suggesting induction principles other than ERM, and imposing restriction on the complexity of \mathcal{F} ⁹, here we introduce a simple yet effective method called memory-based k -nearest neighbours (k -nn) algorithm. The k -nn algorithm assumes that the f_o which minimizes R_{emp} is the function that determines class labels by taking an average of the class labels of instances in T that are close to input \vec{v} .

⁸Although in real-world applications, this assumption does not hold. If the training dataset is small or the hypothesis space \mathcal{F} is large, then there are many functions that can satisfy Equation 3. Under these conditions, however, using ERM may not necessarily result in a function that has a high generalisation ability. Under such circumstances, a function f_o that shows a high performance during the learning procedure shows a poor performance when dealing with data samples other than T . This is often called **overfitting**.

⁹For example, using the assumption that the target function f_o is in the form of a *linear discriminant function*.

k -Nearest Neighbours Algorithm I

The k -nearest neighbours (k -nn) algorithm is a learning technique that is explained by the geometry of vectors in space. In k -nn, instances of data—that is, vectors—are classified based on the class of their nearest neighbours. It is a two-step process:

- ▶ in the first step, the k closest vectors to the data item being classified are located;
- ▶ in the second step, the class label of the data item is determined using the class label of these nearest neighbours.

k -Nearest Neighbours Algorithm II

Given a vector space V and a training dataset $T \in V \times L$, where L is a finite set of class labels, it is assumed that there exists a distance function $d : V \times V \rightarrow \mathbb{R}$ that assigns a distance value $d(\vec{v}, \vec{t})$ to each pair of vectors $\vec{v} \in V$ and $\vec{t} \in T$. In its simplest form, when $k = 1$, for an input vector $\vec{v} \in V$, T is searched for the \vec{t} that has the least distance to the \vec{v} and its class label is assigned to the \vec{v} . This classification task can be formalised by the mapping function nn that returns corresponding label $l \in L$ of vector \vec{t} such that:

$$nn(\vec{v}) = l_{\vec{t}}, \text{ where } \vec{t} = \arg \min_{\vec{y} \in T} d(\vec{v}, \vec{y}). \quad (4)$$

k -Nearest Neighbours Algorithm III

By the same token, the $nn(\vec{v})$ can be generalised to k neighbours. After finding the k closest instances in T to \vec{v} , that is $\{t_1 \cdots t_k\}$, the most straightforward approach—known as **unweighted voting**—is to assign the majority class label among the k nearest neighbours to the data item being classified:

$$k\text{-}nn(\vec{v}) = l_y, \text{ where } l_y = \arg \max_{l \in L} \sum_{i=1}^k \delta(l, f(\vec{t}_i)), \quad (5)$$

k-Nearest Neighbours Algorithm IV

where $f(\vec{t}_i)$ denotes the class label of $\vec{t}_i \in T$, and $\delta(x, y)$ is a function that compares the two class labels x and y , that is:

$$\delta(x, y) = \begin{cases} 1 & x = y \\ 0 & x \neq y \end{cases}. \quad (6)$$

However, a **distance weighted** method can replace the unweighted sum of labels:

$$k\text{-nn}(\vec{v}) = l_y, \text{ where } l_y = \arg \max_{l \in L} \sum_i^k w_i \delta(l, f(\vec{t}_i)), \quad (7)$$

k -Nearest Neighbours Algorithm V

where w_i is real valued function on the distance between \vec{v} and instances from the training set. For example, the weight function can be defined as an inverse of the distances between \vec{v} and $\vec{t}_i \in T$, that is:

$$w_i = \begin{cases} 1 & x = y \\ \frac{1}{d(\vec{v}, \vec{t}_i)} & x \neq y \end{cases}. \quad (8)$$

Similarly, w_i can be defined using an exponential function:

$$w_i = e^{-\alpha d(\vec{v}, \vec{t}_i)^\beta}, \quad (9)$$

where α and β are constant, often $\alpha, \beta = 1$, that are used to control the power of exponential decay factor. The k -nn algorithm,

k-Nearest Neighbours Algorithm VI

thus, can be alternated by adopting different approaches for assigning class labels through definitions of δ and w .

The *k*-nn algorithm is known to be a **lazy-learning** technique, which means that it does not require a training procedure prior to the classification task. The induction takes place during run-time and using training data samples that are presented explicitly. The main computation in the learning and classification task is the scoring of training vectors against an input vector in order to find the *k* nearest neighbours. The *k*-nn, therefore, is also known as an **example-based** or **case-based** learning technique. It is a simple yet

k -Nearest Neighbours Algorithm VII

effective method of classification that has been widely used in many applications.

However, the application of k -nn requires selecting the k value where it is dependent on the distribution of the data is being classified, the distribution of training samples, and the metric that is used to find the nearest neighbours. The value for k is usually selected by a heuristic technique such as cross-validation. In general, larger values of k are believed to reduce the effect of noise; however, this makes class boundaries less distinct. For small values of k , the k -nn method is also known to be sensitive to the

k -Nearest Neighbours Algorithm VIII

presence of noisy or irrelevant data. In addition, when the number of training instances increases, the performance of k -nn reduces.

How to Install NoSke Concordance System? I

The NoSketch Engine can be downloaded from:

<https://nlp.fi.muni.cz/trac/noske/wiki/Downloads>. You can follow the instruction in the web page, or follow the instructions below. If you are a Microsoft Windows user, install WSL and then Ubuntu from the Store. Once you have access to Ubuntu (v. 16.04, 64bit),¹⁰ please follow the following instructions or those in the aforementioned URL.

1. Install Apache server on your machine using the following command:

```
sudo apt-get install apache2
```

How to Install NoSke Concordance System? II

2. Download and install NoSketchEngine required packages:

2.1 Download all the required packages using:

```
wget http://corpora.fi.muni.cz/noske/deb/1604 -r  
-l 2
```

The result of this process is a directory named `corpora.fi.muni.cz` downloaded on your local directory.

2.2 Change your directory using

```
cd corpora.fi.muni.cz/noske/deb/1604
```

2.3 Install Python's `signalfd`:

```
cd python-signalfd/  
sudo dpkg -i  
python-signalfd_0.1-1ubuntu1_amd64.deb  
cd ..
```

You may be get an error message asking you to install Python, please do so.

How to Install NoSke Concordance System? III

2.4 Install **finlib**:

```
cd finlib
sudo dpkg -i finlib_2.36.5-1_amd64.deb
cd ..
```

2.5 Install **manatee-open**:

```
cd manatee-open
```

To ensure you do not face some technical problem regarding dependencies do as follow:

```
sudo dpkg -i
manatee-open-dbg_2.151.5-1ubuntu1_amd64.deb
sudo dpkg -i
manatee-open-dev_2.151.5-1ubuntu1_amd64.deb
sudo dpkg -i
manatee-open_2.151.5-1ubuntu1_amd64.deb
```

How to Install NoSke Concordance System? IV

At this stage, you have installed `manatee-open`, which is the heart of the NoSke and does processes regarding indexing and querying corpora. There are several programming interfaces for manatee, such as for Python, Java, etc.. Among them, we have to install the Python API (for the Bonito interface):

```
sudo dpkg -i  
manatee-open-python_2.151.5-1ubuntu1_amd64.deb
```

Lastly, we install the Susanne demo corpus:

```
sudo dpkg -i  
manatee-open-susanne_2.151.5-1ubuntu1_amd64.deb
```

How to Install NoSke Concordance System? V

2.6 Finally, install the **Bonito** interface:

```
cd ..  
cd bonito-open  
sudo dpkg -i bonito-open_3.99.9-1_all.deb  
sudo dpkg -i bonito-open-www_3.99.9-1_all.deb  
sudo service apache2 restart
```

After this process, open your web browser and navigate to this address:

http://127.0.0.1/bonito/run.cgi/first_form

You must be able to see the first page of the NoSke with the **Susanne** corpus loaded in it.

How to Install NoSke Concordance System? VI

These are important directories after your installation (default settings):

- ▶ `/var/lib/manatee/vert`: this contains all your corpora in the 'vertical' format. We collect these files in this directories.
- ▶ `/var/lib/manatee/registry`: this folder contains reference documents that explain the structure and annotation schema of a vertical corpus file. The registry documents follow certain formats, specified by the NoSke developers.
- ▶ `/var/lib/manatee/data`: this directory contains all the internal index files that are used by the NoSke system.

How to Install NoSke Concordance System? VII

Given that you have a vertical file and its registry, you can compile (upload) it to your NoSke using the following command:

```
sudo compilecorp --no-sk  
/var/lib/manatee/registry/corpus-registry
```

where `/var/lib/manatee/registry/corpus-registry` is the registry file for your corpus.

Once you compile a corpus and index it using manatee, you can add it for use in the Bonito interface. Navigate to Bonito's `run.cgi` file (`/var/www/bonito`) and edit it.

How to Install R? and some basics I

R is a free tool for statistical analysis which covers a range of methods for numerical analysis, and also for graphical representation of the outcome. For further information about the R project see <https://www.r-project.org/>.

To install R, simply download and run a precompiled binary distributions from <https://cran.r-project.org/>.

Stefan Evert and Marco Baroni provide a tutorial:
http://www.stefan-evert.de/SIGIL/sigil_R/

How to Install R? and some basics II




For a shorter version, Markus Dickinson provides a simple step by step introduction to R for corpus linguistics: <http://cl.indiana.edu/~md7/13/615/slides/08-r/08-r.pdf>.

Also, look for books by




- ▶ Stefan Thomas Gries: Quantitative Corpus Linguistics with R (<https://katalog.ulb.hhu.de/Record/003677837>);
- ▶ Guillaume Desagulier: Corpus Linguistics and Statistics with R (<https://katalog.ulb.hhu.de/Record/003989431>).

Both books are available through ULB (both in print and electronic format – see the link above).

Bibliography I

-  Biber, D., Conrad, S., and Reppen, R. (1998).
Corpus Linguistics: Investigating Language Structure and Use.
Cambridge Approaches to Linguistics. Cambridge University Press.
-  Jurafsky, D. and Martin, J. H. (2000).
Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.
Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition.
-  Kovar, V. (2017).
Corpus querying: Corpus query language.

Bibliography II

-  Lyse, G. I. and Andersen, G. (2012).
Collocations and statistical analysis of n-grams.
Studies in Corpus Linguistics.
-  Manning, C. D. and Schütze, H. (1999).
Foundations of Statistical Natural Language Processing.
MIT Press, Cambridge, MA, USA.
-  Markie, P. (2017).
Rationalism vs. empiricism.
In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2017 edition.

Bibliography III



McEnery, A. M. and Wilson, A. (2001).

Corpus linguistics: an introduction.

Edinburgh University Press.



Sinclair, J. (1996).

Preliminary recommendations on corpus typology.

Technical Report Document EAG-TCWG-CTYP/P, EAGLES.

<http://www.ilc.cnr.it/EAGLES96/corpusstyp/corpusstyp.html>.