

# **Annotating with SACR and exploiting annotations**

10 May 2019

Bruno Oberle

Workshop at  
Cardiff University



# Outline

- **Introduction**

- annotation process
- overview of the workshop

- **Practice:**

- annotation guidelines and scheme
- marking the text
- exploiting annotations (CRViewer and a spreadsheet program like Excel)



# Introduction

# Annotation with SACR

- **Script d'Annotation des Chaînes de Référence**
- **Coreference Chain Annotation Script**

### Coreference Annotation Script 5.0.0 ([help](#))

#title:Le Laboureur et l'Aigle (Ésope, Fables, traduction Chambry)

[#1] **laboureur** Un laboureur, **laboureur** ∅ ayant trouvé **aigle** un aigle pris **filet** au filet fut si frappé de **beauteAigle** **aigle** sa beauté qu'**laboureur** il **aigle** le délivra et **aigle** lui **laboureur** ∅ donna **liberteAigle** la liberté. **aigle** L'aigle ne **aigle** se montra pas ingrat envers **laboureur** **aigle** son bienfaiteur ; mais **laboureur** le **aigle** ∅ voyant **mur** un mur **mur** qui menaçait ruine, **aigle** il vola vers **laboureur** lui **aigle** ses griffes **aigle** ses griffes **bandeau** le bandeau **bandeau** qui **laboureur** se levait tête **laboureur** la tête. **laboureur** L'homme **laboureur** se leva

n nom propre  
t nom sans déterminant  
d nom défini  
i nom indéfini  
p nom possessif  
m nom démonstratif  
s pronom personnel  
f pronom réfléchi  
r pronom relatif  
o pronom démonstratif  
z pronom zéro  
e déterminant possessif

Edit Mode | i nom indéfini | v compl (verbe) | 1: aigle - v

### Chains and Links

**laboureur**

**aigle**  
un aigle pris au filet  
sa  
le  
lui  
L'aigle  
se  
son  
∅  
il  
∅  
ses  
sa  
L'aigle

**filet**  
au filet

**beauteAigle**  
sa beauté

**liberteAigle**  
la liberté

# Annotation process

- **Preparing:**
  - *what* to annotate (annotation guidelines)
  - *how* to annotate (annotation scheme)
- **Marking the text**
- **Exploiting:**
  - visualizing annotations in the text
  - using statistics and graphs

# Types of annotations

- **SACR can be used to:**
  - **group** several **segments** of text together,
  - annotate **properties** for each segment
- **For example:**
  - semantic types of verbs
  - argument structures
  - types of adverbials
  - etc.

# Coreference

- ***Referring expression (or mention):***
  - an expression that refers to an entity of the extra-linguistic world
- ***Coreference chain:***
  - all the referring expressions that refer to the same entity
- **Examples:**
  - [The cat]** is walking. **[It]** is happy.
  - [The computer]** is not working properly: **[it]** is overheating. **[This machine]** should be replaced.

# Marking coreference chains

- **Segments:**

finding referring expressions

- **Groups:**

building chains

- **Properties:**

characteristics of referring expressions:

- part of speech of the syntactic head
- grammatical function
- gender, number
- modifiers
- semantic information
- etc.



# Segments: referring expressions

## Coreference Annotation Script 5.0.0 ([help](#))

[#1] M1 A Peasant found M2 an Eagle captured in M3 a trap , and much admiring M4 the bird , set M5 him free. M6 The Eagle did not prove ungrateful to M8 M7 his deliverer , for seeing M9 the Peasant sitting under M10 a wall M11 which was not safe , M12 he flew toward M13 him and with M14 M15 his talons snatched M16 a bundle from M17 M18 his head . When M19 the Peasant rose in pursuit, M20 the Eagle let M21 the bundle fall again. Taking M22 it up, M23 the man returned to M24 the same place , to find that M25 the wall under M26 which M27 he had been sitting had fallen to pieces; and M28 he marveled at

# Groups: coreference chains

## Coreference Annotation Script 5.0.0 ([help](#))

[#1] Peasant A Peasant found Eagle an Eagle captured in M3 a trap , and much admiring

Eagle the bird , set Peasant him free. Eagle The Eagle did not prove ungrateful to

Peasant Eagle his deliverer , for seeing Peasant the Peasant sitting under Wall a wall

Wall which was not safe , Eagle he flew toward Peasant him and with M14 Eagle his talons

snatched Bundle a bundle from M17 Peasant his head . When Peasant the Peasant rose in

pursuit, Eagle the Eagle let Bundle the bundle fall again. Taking Bundle it up, Peasant the

# Properties of each referring expression

## Coreference Annotation Script 5.0.0 ([help](#))

[#1] Peasant A Peasant found Eagle an Eagle captured in M3 a trap , and much admiring

Eagle the bird , set Peasant him free. Eagle The Eagle did not prove ungrateful to

Peasant Eagle his deliverer , for seeing Peasant the Peasant sitting under Wall a wall

Wall , Eagle he flew toward Peasant him and with M14 Eagle his talons

snatch le from M17 Peasant his head . When Peasant the Peasant rose in

- d noun with definite article
- i noun with indefinite article
- n noun with determiner
- t noun without determiner
- s personal pronoun
- r relative pronoun
- e possessive adjective

Edit Mode

i noun with indefinite article

o object

1: Eagle

# Exploiting annotations

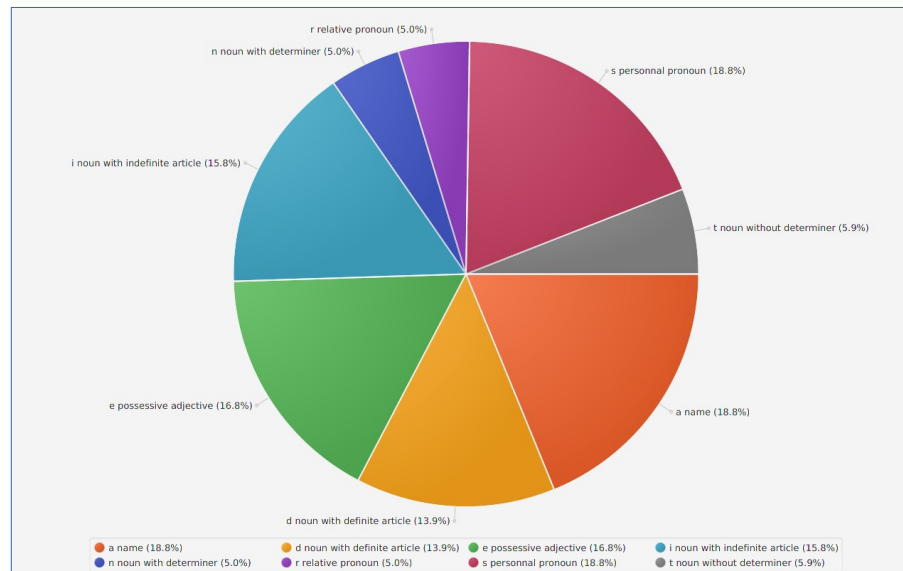
- **Concordance:**

Left	Text
	*A Peasant* found an Eagle captured in
A Peasant found	*an Eagle captured in a trap*, and much admiring the bird,
a trap, and much admiring	*the bird*, set him free. The Eagle
much admiring the bird, set	*him* free. The Eagle did not
the bird, set him free.	*The Eagle* did not prove ungrateful to
did not prove ungrateful to	*his deliverer*, for seeing the Peasant sitting
did not prove ungrateful to	*his* deliverer, for seeing the Peasant
to his deliverer, for seeing	*the Peasant sitting under a wall which was not safe*, he flew toward him and
seeing the Peasant sitting under	*a wall which was not safe*, he flew toward him and
Peasant sitting under a wall	*which* was not safe, he flew
wall which was not safe,	*he* flew toward him and with
not safe, he flew toward	*him* and with his talons snatched
flew toward him and with	*his* talons snatched a bundle from
and with his talons snatched	*a bundle* from his head. When the
talons snatched a bundle from	*his* head. When the Peasant rose
bundle from his head. When	*the Peasant* rose in pursuit, the Eagle
the Peasant rose in pursuit,	*the Eagle* let the bundle fall again.
in pursuit, the Eagle let	*the bundle* fall again. Taking it up,

# Exploiting annotations

- **Statistics:**

Id	function: s subject	function: o object	function: a adverbial	function: t other	function: m noun modifier	partofspeech: i noun with indefinite ar...	partofspeech: s personal pron...
aesop.sacr	10	9	4	3	0	4	7
caesar.sacr	11	7	1	0	11	1	7
cicero.sacr	5	11	0	0	1	6	3
pliny.sacr	5	8	1	2	12	5	2



# Studying coreference

- **Study how entities are expressed in the text:**
  - main characters vs secondary characters
  - types of referents (people vs animal vs objects vs abstract concepts)
  - evolution of null anaphora from Old French to Modern French
  - relations between coreference chains and textual structure (sections of texts, paragraphs...)
  - etc.

Using SACR

to make coreference chain annotations

# Guidelines

- **What is a “referring expression”?**
  - physical entities vs abstract entities?
  - adverbials, subject complements, etc.?
- **What are the limits of a referring expression?**
- **What is a coreference relation?**
  - “climate change” vs “global warming”
  - “water of Paris” vs “water of London”



# Guidelines

- **Different annotation campaigns, different choices**
- **Most often:**
  - noun, pronoun and possessive determiner
  - the whole noun phrase, including all the modifiers

# Annotation scheme

- **List of properties to be annotated for each expression:**
  - morphology
  - syntax
  - semantics
  - etc.
- **For today:**
  - part of speech of the syntactic head
  - grammatical function

# Beginning with SACR

- go to [boberle.com/projects/sacr](http://boberle.com/projects/sacr)

## SACR - Coreference Chain Annotation Tool - 5.2.0

Please use [Firefox](#), or at least [Chromium](#) or [Google Chrome](#)!

You will find a user guide [here](#), and some video tutorials (in French) [here](#).

Write or paste the **text** in the text area below, or use one of these options:

- load a file:  No file selected.
- use a sample text:  ▾

Write or paste the **properties** in the text area below, or use one of these options:

- load a file:  No file selected.
- use a sample schema:  ▾

# Import text

- **How:**

- copy and paste a text in the text area
  - e.g. from Wikipedia
- import a text from a file
- use one of the sample texts (for training)

- **Format:**

- a white line between each paragraph
- you may add directives:
  - `#textid: name of text`
  - `#textmetadata:property=value` (e.g. the genre or the topic of a text, in order to compare different genres of texts; or the author, if the corpus contains several texts by different authors and you want compare them by authors; etc.)

# Import a scheme

- **How:**
  - copy and past to the text area
  - import a text from a file
  - use one of the sample schemes (for training)
- **Best practice:**
  - define the scheme in advance
  - save it into a file
  - load the same scheme file for all the texts of the corpus

# Format of the scheme

**PROP:name=name\_of\_property**

**\$\$\$** # for an “empty value”

**a name** # value 1

**d noun with definite article** # value 2

**i noun with indefinite article** # value 3

...

**different properties are separated by a white line**

# Format of the scheme

- **other possible parameters**
  - showname=true|false
  - type=text|head
  - newline=true|false

# Format of the scheme

- **example:**

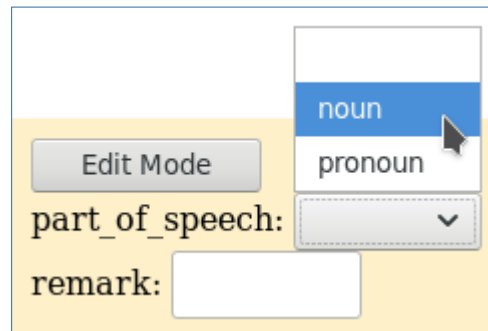
- PROP:name=partofspeech,showname=true,newline=true

\$\$\$

noun

pronoun

PROP:name=remark,type=text,showname=true,newline=true



The screenshot shows a form with a yellow background. At the top left is a grey button labeled "Edit Mode". Below it is the label "part\_of\_speech:" followed by a dropdown menu. The dropdown menu is open, showing two options: "noun" (highlighted in blue) and "pronoun". Below the dropdown is the label "remark:" followed by a text input field.



# For today...

- **Choose:**
  - the “Peasant” by Aesop, without predefined annotations
  - the sample scheme for English

# Step 1: marking referring expressions

- **To mark an expression:**
  - if one word: double-click on that word
  - if two words or more: click on the first and last word
- **To remove an expression:**
  - select an expression
  - use the “delete” key or “backspace”

[#1] M1 A Peasant found an Eagle

# Step 1: marking referring expressions

- **Nested expressions:**

ungrateful to M3 M2 his deliverer ,

- **A selected expression:**

M1 A Peasant

- **A non-selected expression:**

M1 A Peasant

# At the end of step 1...

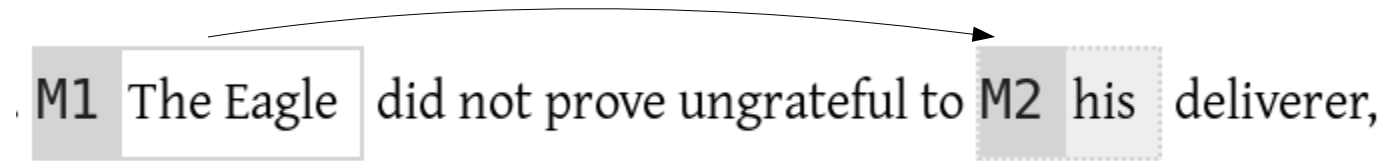
## Coreference Annotation Script 5.0.0 ([help](#))

[#1] M1 A Peasant found M2 an Eagle captured in M3 a trap , and much admiring M4 the bird , set M5 him free. M6 The Eagle did not prove ungrateful to M8 M7 his deliverer , for seeing M9 the Peasant sitting under M10 a wall M11 which was not safe , M12 he flew toward M13 him and with M14 M15 his talons snatched M16 a bundle from M17 M18 his head . When M19 the Peasant rose in pursuit, M20 the Eagle let M21 the bundle fall again. Taking M22 it up, M23 the man returned to M24 the same place , to find that M25 the wall under M26 which M27 he had been sitting had fallen to pieces; and M28 he marveled at

## Step 2: Building chains

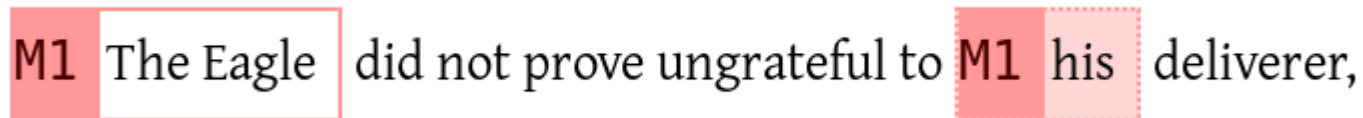
- **drag an expression and drop it on another one**

M1 The Eagle did not prove ungrateful to M2 his deliverer,



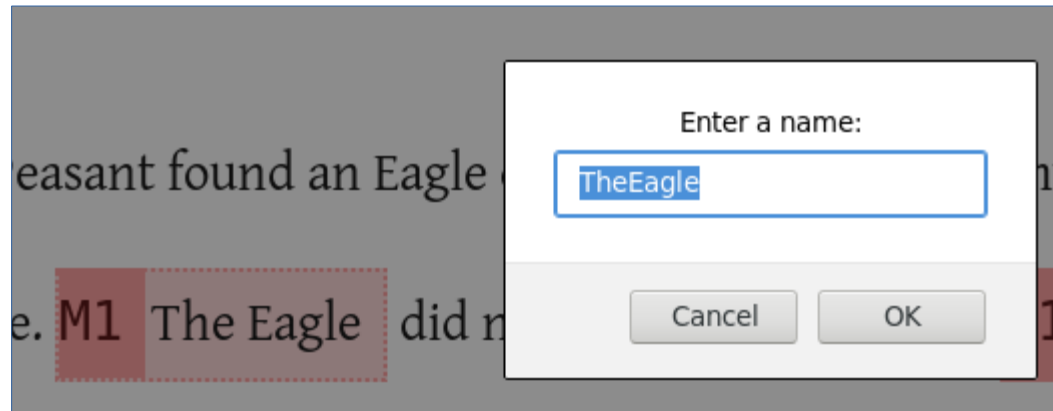
- **a coreference relation is made:**
  - the color changes
  - the name of the chain changes

M1 The Eagle did not prove ungrateful to M1 his deliverer,



## Step 2: building chains

- **Change the color with “c”**
- **Change the name with “m”**
  - select an expression with a meaningful content
  - press “m”



e. TheEagle The Eagle did not prove ungrateful to TheEagle his

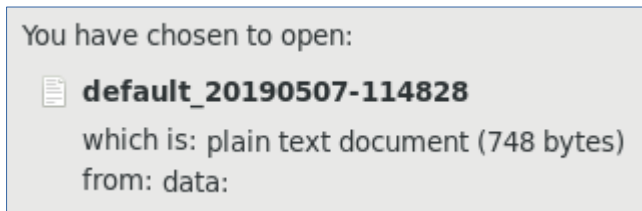
## Step 2: building chains

- **More commands:**

- “n” to *rename* (current name provided by default)
- “shift” to *detach* an expression from its chain and make it a singleton
- drag and drop a whole chain on another one to *merge* the chains

# Saving the annotations

- **Press “w” (write) to save the annotations**
  - Firefox will offer you to open or download the file
  - a name with a *timestamp* is proposed



- each time you press “w”, the timestamp is updated, so you can save different stages of the annotation process



# Saving the scheme

- **SACR saves the text together with the annotations in one file**
- **The scheme is saved separately, so it can be used for different texts**
  - save it by pressing the “x” key
  - but it is usually defined and saved before annotation begins so you don’t have to save when annotating a text
- **To show the scheme in a dialog box, press the “X” key**

# Getting help

- Press “h” to show all the available command:

## Coreference Chain Annotation Tool 5.2.0 ([help](#))

#title:The Peasant and the Eagle

[#1] A Peasant found a

ungrateful to his deliv

his talons snatched a

Taking it up, the man

pieces; and he marveled at the service rendered him by the Eagle.

Create links by clicking on tokens:

- to create a link with one token: double click on that token
- to create a link with more than one token: click on the first and last token
- by default, a default name will be used
  - hold shift to be asked for a name
  - hold ctrl to automatically attach the link to the last selected chain

Create chain with drag and drop:

- source is singleton, target is singleton:
  - target will be attached to source chain
- source is singleton and target is not singleton:
  - source will be attached to target chain
- hold ctrl for the reverse- source is not singleton and target is singleton:
  - target will be attached to source chain
- source is not singleton and target is not singleton:
  - the chains will be merged (you will be asked to confirm)

OK

agle did not prove

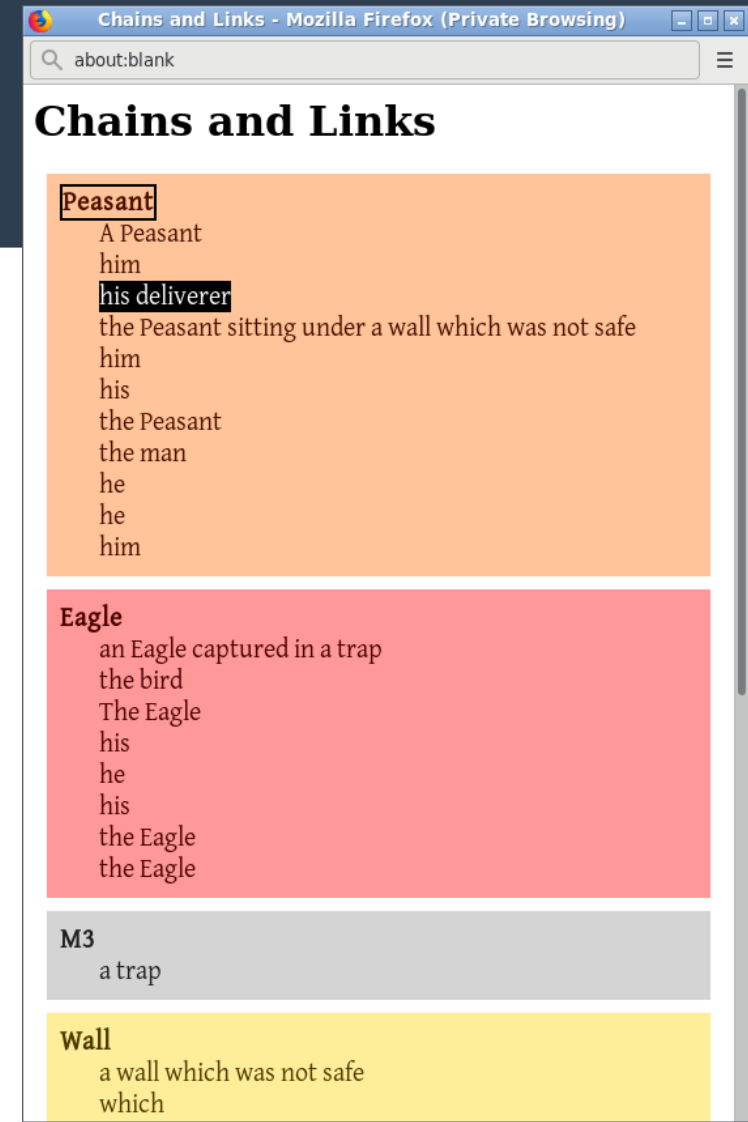
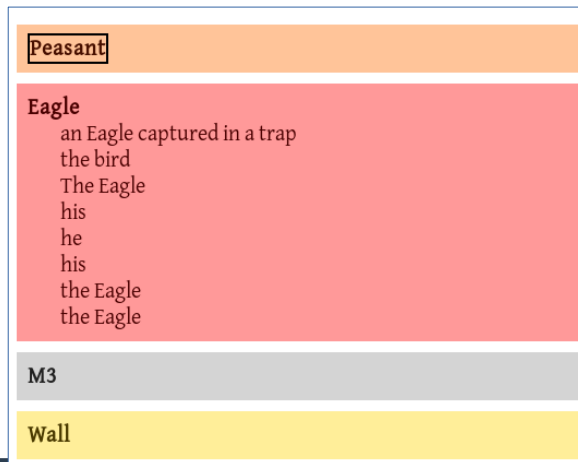
toward him and with

ndle fall again.

sitting had fallen to

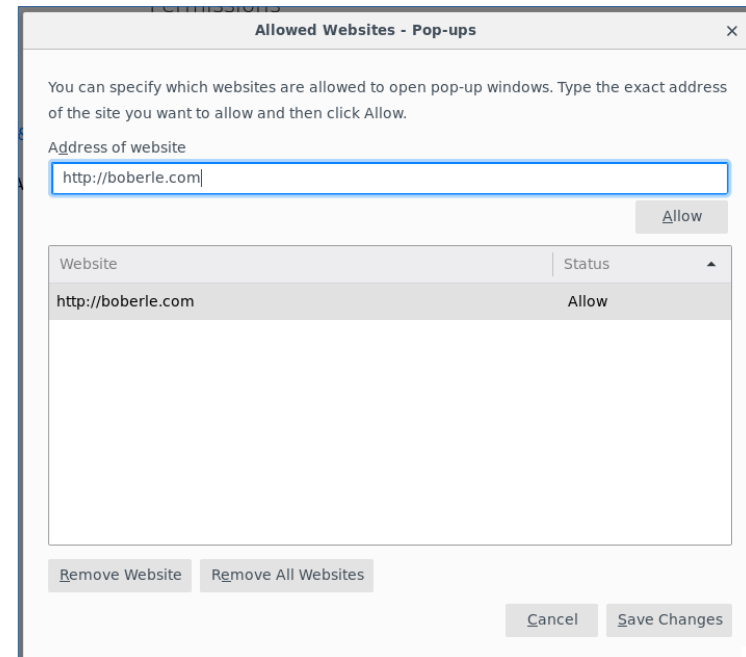
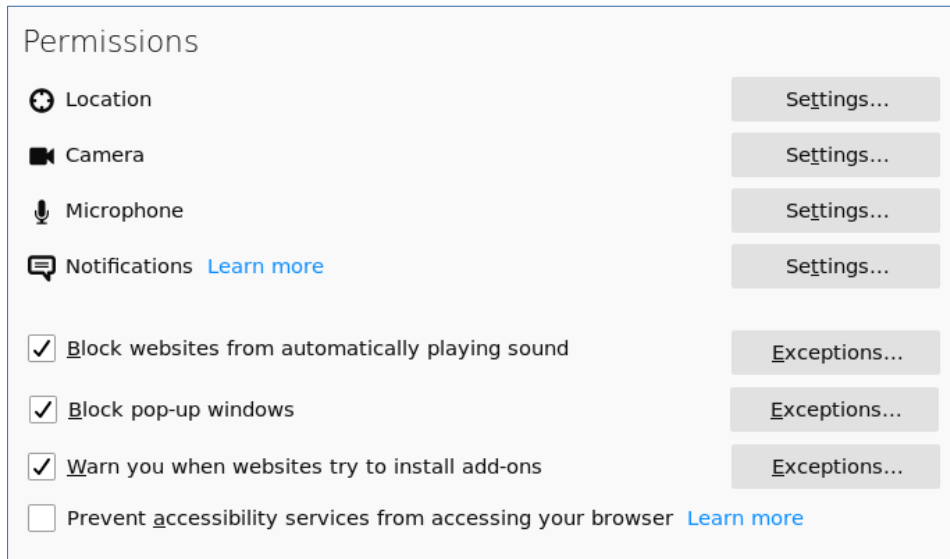
# The popup

- Press “p”
- Drag and drop from the popup to the main widow and vice versa
- Collapse and expand with “E” and “e”



# The popup

- **Enabling popup on Firefox:**



# At the end of step 2...

## Coreference Annotation Script 5.0.0 ([help](#))

[#1] Peasant A Peasant found Eagle an Eagle captured in M3 a trap , and much admiring Eagle the bird , set Peasant him free. Eagle The Eagle did not prove ungrateful to Peasant Eagle his deliverer , for seeing Peasant the Peasant sitting under Wall a wall Wall which was not safe , Eagle he flew toward Peasant him and with M14 Eagle his talons snatched Bundle a bundle from M17 Peasant his head . When Peasant the Peasant rose in pursuit, Eagle the Eagle let Bundle the bundle fall again. Taking Bundle it up, Peasant the

# Step 3: Feature annotation

- **Select an expression**
- **Fill out the boxes at the bottom of the screen**

The screenshot shows a text annotation interface. The text being annotated is: "snatched from M17 Peasant his head . in pursuit, Eagle the Eagle let B bundle it up, Peasant the man". The word "Peasant" is highlighted in orange. A dropdown menu is open over "Peasant", listing the following options: "a name", "d noun with definite article" (highlighted in blue), "i noun with indefinite article", "n noun with determiner", "t noun without determiner", "s personal pronoun", "r relative pronoun", and "e possessive adjective". At the bottom of the interface, there are three input fields: "Edit Mode" (a button), "d noun with definite article" (a dropdown menu), "s subject" (a dropdown menu), and "1: Eagle" (a dropdown menu).

# Step 3: Feature annotation

- **To go faster:**
  - in “edit mode”, select “auto-edit”
  - select the first expression of the text
  - each time you set a value for a feature, the script will automatically go to the next expression

## Edit mode

Choose the edit mode: normal: html elements behave normally ▼

only links on the same chain  only visible links

save Cancel

# Exploiting annotations



# Exploiting annotations

- **Four ways:**
  - a quick look with SACR
  - CRViewer: a easy-to-use tool for coreference exploration
  - exporting tables to use in a spreadsheet program or a specialized statistic program
  - using an online interface

# A quick look at annotations with SACR

- **The “search” function in SACR:**
  - press “s” and enter a value
  - all matching expressions will show up in the text

## Coreference Annotation Script 5.0.0 ([help](#))

[#1] A Peasant found an Eagle captured in a trap, and much admiring the bird, set Peasant him free. The Eagle did not prove ungrateful to his deliverer, for seeing the Peasant sitting under a wall which was not safe, Eagle he flew toward Peasant him and with his talons snatched a bundle from his head. When the Peasant rose in pursuit, the Eagle let the bundle fall again. Taking Bundle it up, the man returned to the same place, to find that the

Edit Mode s personal pronoun o object 0: him

### Search box

use regex

partofspeech is equal to s personal pronoun

search

Cancel

# CRViewer

# CRViewer

- **For this activity, download all the pre-annotated sample texts from SACR (or [boberle.com/projects/sacr/preannotated\\_texts.zip](http://boberle.com/projects/sacr/preannotated_texts.zip))**
- **Go to [boberle.com/projects/crviewer](http://boberle.com/projects/crviewer) and download the “zip” file**
- **Decompress it and run the “jar” file**

# CRViewer: interface

- **Select options:**
  - minimum chain size
  - filter
    - by text id
    - by chain name
    - features

=== SELECT OPTIONS ===

Min size:

Refname:

Refname:

Full id:

Split by:

# CRViewer: interface

- **Display option:**
  - type of display
    - concordancer
    - table
    - graph
  - property/statistic to show
  - context width for concordancer

=== DISPLAY OPTIONS ===

Display:	concordancer ▼
Unit:	corpus ▼
X Property:	function ▼
Y Stat:	getAverageLinkToLinkDis... ▼
Display Filter:	<input type="text"/>
Stab coeff p...	function ▼
Stab coeff v...	<input type="text"/>
Context wid...	5

# CRViewer: Concordancer

- Choose concordancer and click on “Update!”

Left	Text
	*A Peasant* found an Eagle captured in
A Peasant found	*an Eagle captured in a trap*, and much admiring the bird,
a trap, and much admiring	*the bird*, set him free. The Eagle
much admiring the bird, set	*him* free. The Eagle did not
the bird, set him free.	*The Eagle* did not prove ungrateful to
did not prove ungrateful to	*his deliverer*, for seeing the Peasant sitting
did not prove ungrateful to	*his* deliverer, for seeing the Peasant
to his deliverer, for seeing	*the Peasant sitting under a wall which was not safe*, he flew toward him and
seeing the Peasant sitting under	*a wall which was not safe*, he flew toward him and
Peasant sitting under a wall	*which* was not safe, he flew
wall which was not safe,	*he* flew toward him and with
not safe, he flew toward	*him* and with his talons snatched
flew toward him and with	*his* talons snatched a bundle from
and with his talons snatched	*a bundle* from his head. When the
talons snatched a bundle from	*his* head. When the Peasant rose
bundle from his head. When	*the Peasant* rose in pursuit, the Eagle
the Peasant rose in pursuit,	*the Eagle* let the bundle fall again.
in pursuit, the Eagle let	*the bundle* fall again. Taking it up,

# CRViewer: Concordancer

- **Filter:**

- choose the referent “Caesar”
- select the part of speech “a name”
- note: property values are *regular expressions*

Refname: Caesar

Full id:

partofspeech: a name

	Text and Right	Text and Chain
	*Gaius Julius Caesar* (12 or 13 July 100	caesar.sacr:Caesar
In 60 BC,	*Caesar*, Crassus and Pompey formed the	caesar.sacr:Caesar
the frequent support of Cicero.	*Caesar* rose to become one of	caesar.sacr:Caesar
51 BC. During this time,	*Caesar* became the first Roman general	caesar.sacr:Caesar
the Channel to invade Britain.	*Caesar*'s wars extended Rome's	caesar.sacr:Caesar
Wars concluded, the Senate ordered	*Caesar* to step down from his	caesar.sacr:Caesar
unsanctioned wars. As a result,	*Caesar* found himself with no other	caesar.sacr:Caesar
Italy under arms. This began	*Caesar*'s civil war, and his	caesar.sacr:Caesar



# CRViewer: Filtering

- **Filter by chain size:**

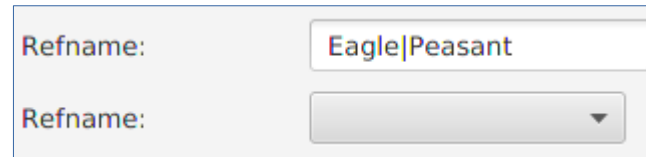
- how many mentions (at least) must a chain have?
- enter 1 to get all chains, including singletons
- enter 2 to get chains with 2 or more mentions
- enter 3 to get chains with 3 or more mentions
- etc.

Min size:

# CRViewer: Filtering

- **Filter by chain name:**

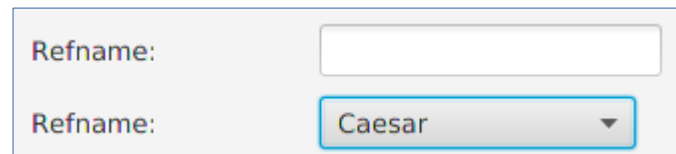
- use the box with a regular expression
  - you can select several chains with “|”



Refname:

Refname:

- or enter any regular expression
- or select **one** chain in the drop down list



Refname:

Refname:

# CRViewer: Filtering

- **The concordance varies accordingly when you press the “Update!” button**

Left	Text and Right	Text and Chain
	*A Peasant* found an Eagle captured in	aesop.sacr:Peasant
A Peasant found	*an Eagle captured in a trap*, and much admiring the bird,	aesop.sacr:Eagle
a trap, and much admiring	*the bird*, set him free. The Eagle	aesop.sacr:Eagle
much admiring the bird, set	*him* free. The Eagle did not	aesop.sacr:Peasant
the bird, set him free.	*The Eagle* did not prove ungrateful to	aesop.sacr:Eagle
did not prove ungrateful to	*his deliverer*, for seeing the Peasant sitting	aesop.sacr:Peasant
did not prove ungrateful to	*his* deliverer, for seeing the Peasant	aesop.sacr:Eagle
to his deliverer, for seeing	*the Peasant sitting under a wall which was not safe*, he flew toward him and	aesop.sacr:Peasant
wall which was not safe,	*he* flew toward him and with	aesop.sacr:Eagle
not safe, he flew toward	*him* and with his talons snatched	aesop.sacr:Peasant
flew toward him and with	*his* talons snatched a bundle from	aesop.sacr:Eagle
talons snatched a bundle from	*his* head. When the Peasant rose	aesop.sacr:Peasant
bundle from his head. When	*the Peasant* rose in pursuit, the Eagle	aesop.sacr:Peasant
the Peasant rose in pursuit,	*the Eagle* let the bundle fall again.	aesop.sacr:Eagle
fall again. Taking it up,	*the man* returned to the same place,	aesop.sacr:Peasant
that the wall under which	*he* had been sitting had fallen	aesop.sacr:Peasant
had fallen to pieces; and	*he* marveled at the service rendered	aesop.sacr:Peasant
marveled at the service rendered	*him* by the Eagle.	aesop.sacr:Peasant
the service rendered him by	*the Eagle*.	aesop.sacr:Eagle

# CRViewer: Filtering

- **Filtering by text id to show only chains from some texts**
- **Again, a regular expression**

Full id:

- **Text id is:**
  - the file name
  - or the #textid directive

# CRViewer: Filtering

- **Filter by property values:**
  - regular expressions
  - boxes are linked with an “and” operator
- **Ex. to show mentions that are:**
  - name
  - **AND** subject or object:

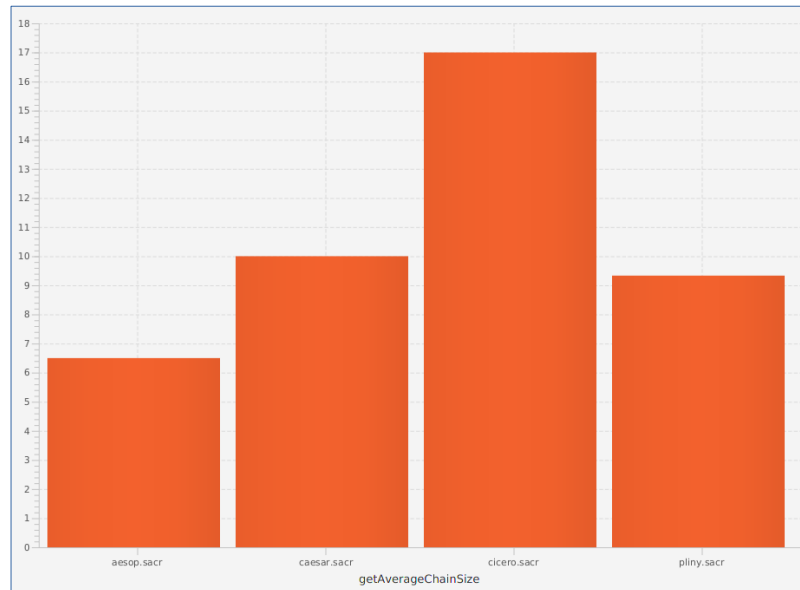
partofspeech ▼	a name
function ▼	s subject o object

# CRViewer: Filtering

- **These filters are applied whatever the display is:**
  - concordance
  - table
  - graph

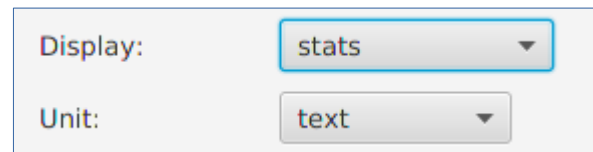
# CRViewer: Statistics

Id	AvgL2LDist	AvgLinkLength	AvgChainSize	ChainCount	LinkCount
aesop.sacr	11.75	2.25	6.5	4	26
caesar.sacr	40.333333333333336	2.6666666666666665	10.0	3	30
cicero.sacr	7.0	2.0	17.0	1	17
pliny.sacr	18.666666666666668	1.6666666666666667	9.333333333333334	3	28



# CRViewer: Statistics

- **To get a table**
  - choose “stats”
  - choose a unit:
    - “corpus” for averages for the whole corpus
    - “text” to get details by texts
    - “chain” to get details by chains



The image shows a screenshot of a user interface for CRViewer. It features two dropdown menus. The first is labeled 'Display:' and has 'stats' selected. The second is labeled 'Unit:' and has 'text' selected. Both dropdown menus have a small downward-pointing arrow on the right side.



# CRViewer: Statistics

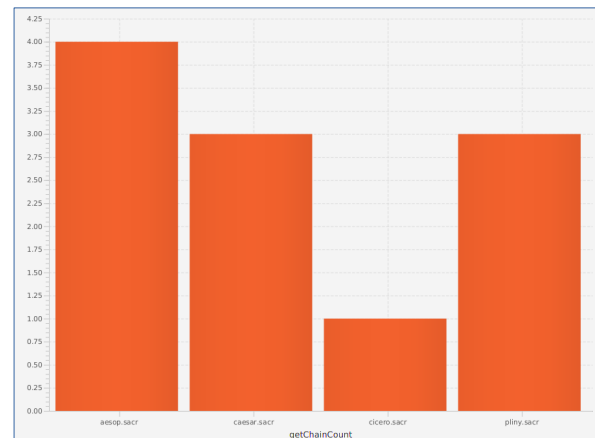
- **Results:**

Id	AvgL2LDist	AvgLinkLength	AvgChainSize	ChainCount	LinkCount	
aesop.sacr	11.75	2.25	6.5	4	26	0.5595
caesar.sacr	40.333333333333336	2.6666666666666665	10.0	3	30	0.6565
cicero.sacr	7.0	2.0	17.0	1	17	0.25
pliny.sacr	18.666666666666668	1.6666666666666667	9.333333333333334	3	28	0.3562

# CRViewer: Statistics

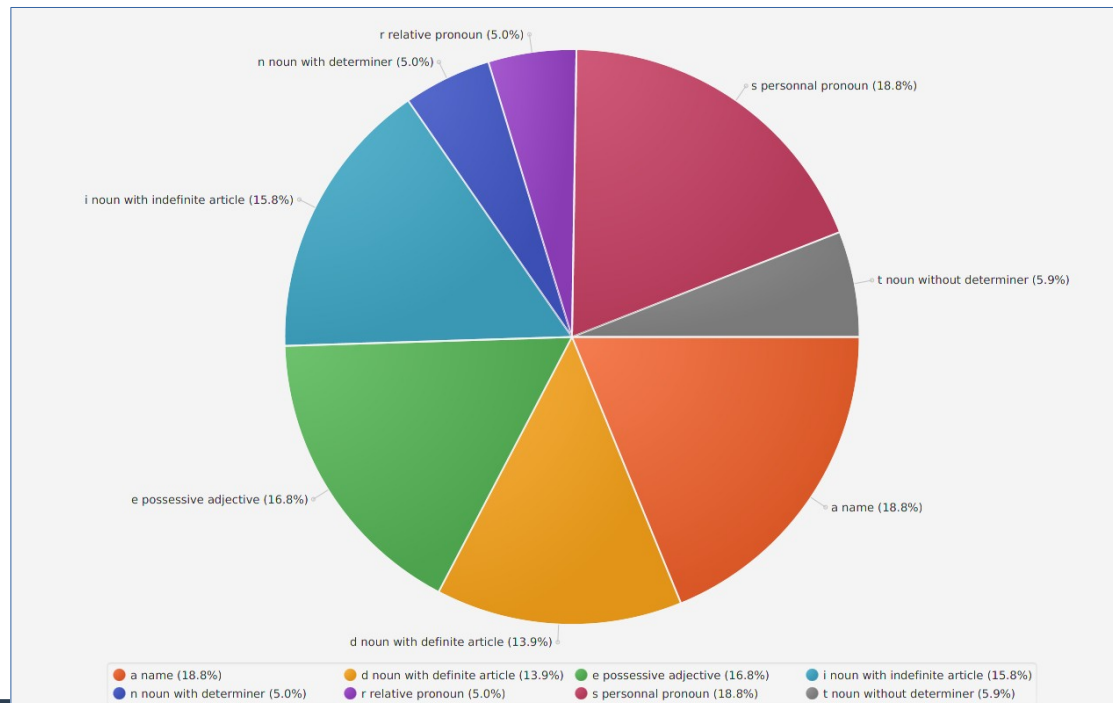
- **To get a graph:**
  - choose “statsBar”
  - choose “text”
  - choose a value in the “Y Stat” field, for example “getChainCount” to get the graph for chain counts
  - ignore “X Property”

Display:	statsBar
Unit:	text
X Property:	function
Y Stat:	getChainCount



# CRViewer: Frequencies

Id	function: s subject	function: o object	function: a adverbial	function: t other	function: m noun modifier	partofspeech: i noun with indefinite ar...	partofspeech: s personal pron...
aesop.sacr	10	9	4	3	0	4	7
caesar.sacr	11	7	1	0	11	1	7
cicero.sacr	5	11	0	0	1	6	3
pliny.sacr	5	8	1	2	12	5	2



# CRViewer: Frequencies

- **To get a table of frequencies:**

- choose “frequencies”
- choose a unit, as before

Display:

Unit:

- **The result is a table with the frequencies for all the properties**

Id	function: s subject	function: o object	function: a adverbial	function: t other	function: m noun modifier	partofspeech: i noun with indefinite ar...	partofspeech: s personal pr
aesop.sacr	10	9	4	3	0	4	7
caesar.sacr	11	7	1	0	11	1	7
cicero.sacr	5	11	0	0	1	6	3
pliny.sacr	5	8	1	2	12	5	2

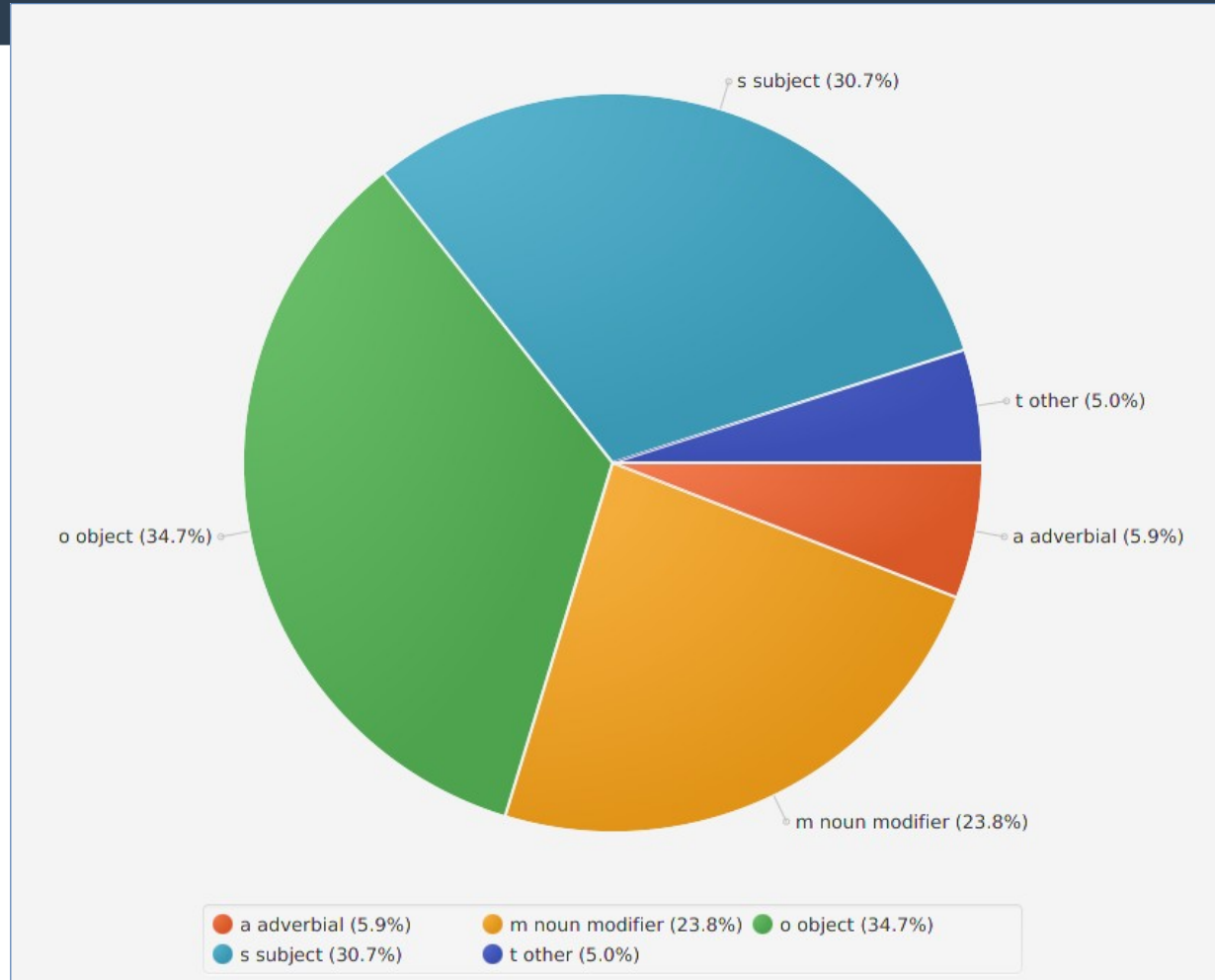
# CRViewer: Frequencies

- **To get a pie:**
  - choose “pie”
  - choose a property in the “X Property” field
  - ignore “Y Stat” and “Unit”

Display:	pie
Unit:	text
X Property:	function
Y Stat:	getChainCount

# CRViewer: Frequencies

- **Result:**



# CRViewer: Frequencies and Filter

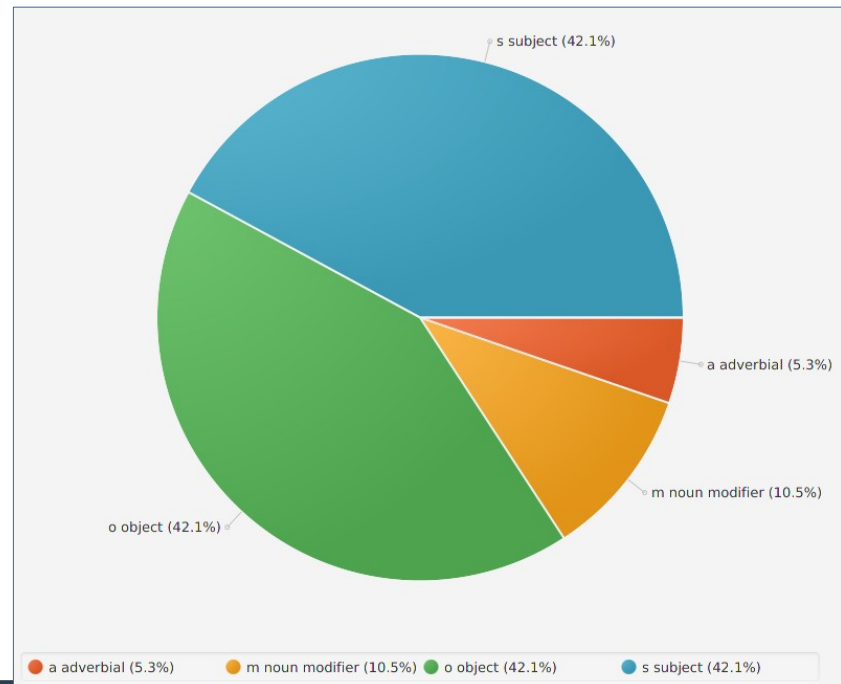
- **To show a distribution (pie) of function for personal pronouns only:**

- set a filter:

partofspeech ▼ s personal pronoun

- and apply as before:

Display: pie ▼  
Unit: chain ▼  
X Property: function ▼

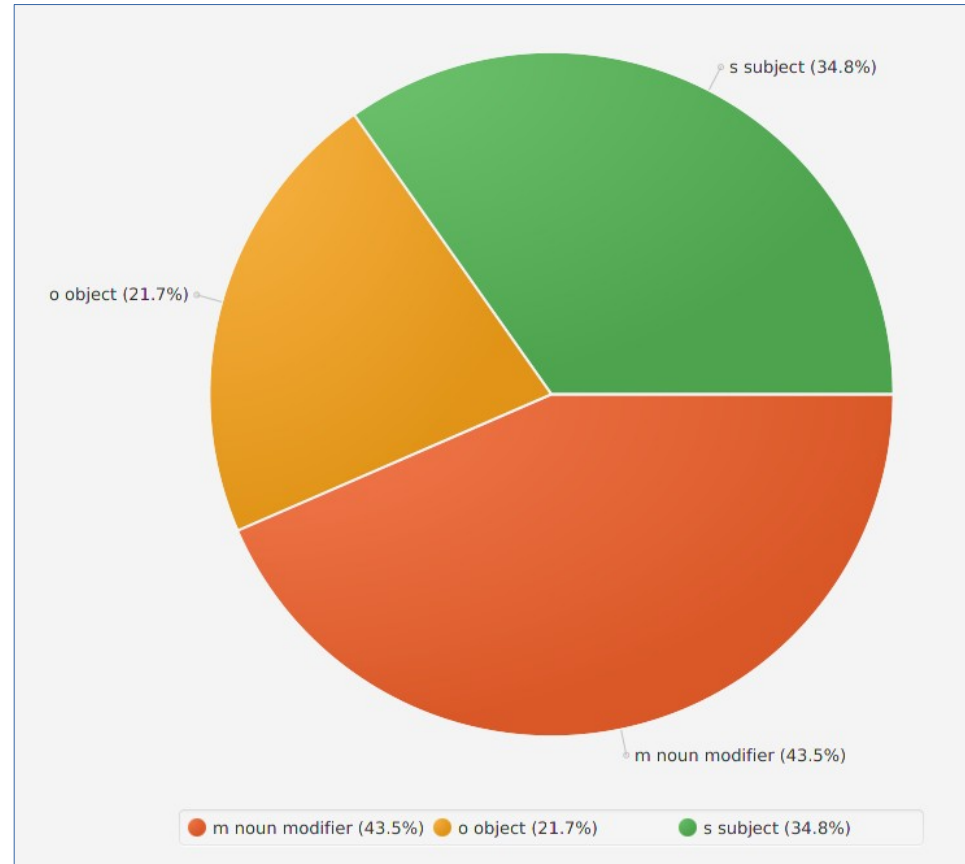


# CRViewer: Frequencies and Filter

- **For only Caesar:**
  - set the filter:

Refname:

Refname:





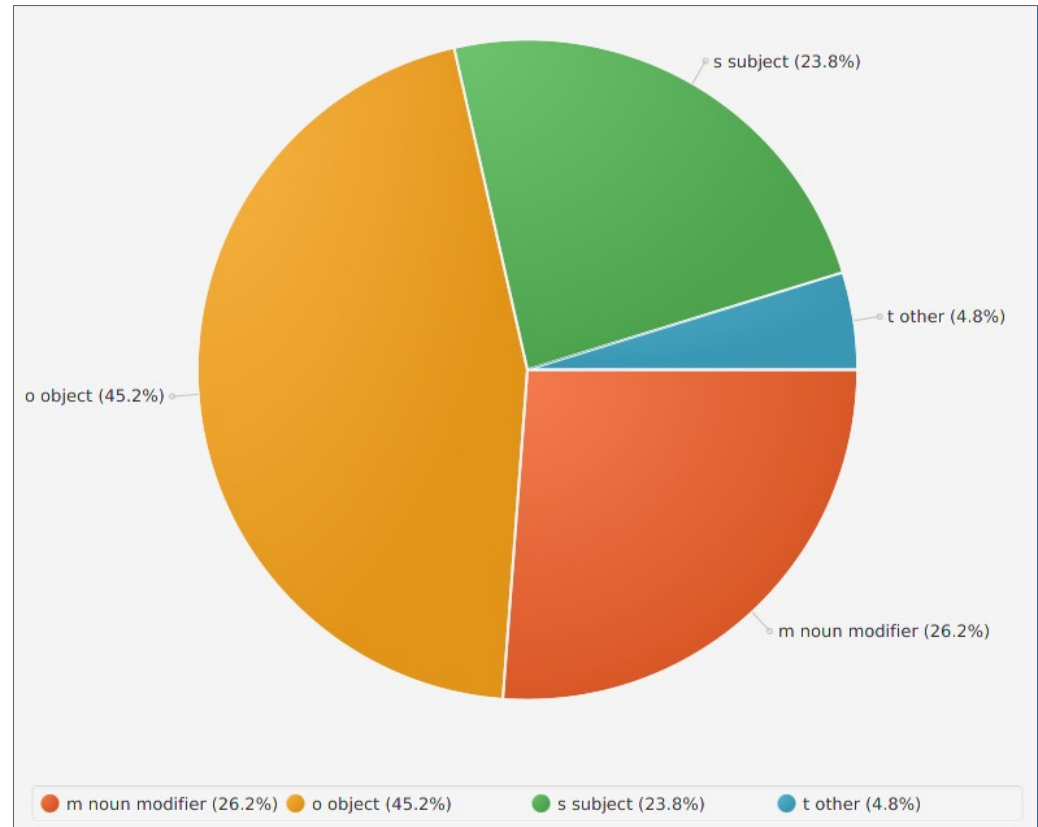
# CRViewer: Statistics

- **For Caesar, Cicero and Pliny:**

- set the filter:

Refname:

Refname:



# CRViewer: How to get the output

- **To get the output (table, graph):**
  - in the directory where the jar file is
  - a new file is created each time “Update!” is pressed
- **Format:**
  - table are exported in CSV (Comma/Tab Separated Values) files  
example: `output_frequencies.tsv`
    - you can import them in a spreadsheet like Excel / Calc
  - graph are exported in PNG format (picture)  
example: `output_pie_partofspeech.png`

# Exporting tables

# Why exporting tables

- **conversion of all the annotated data to CSV (Comma Separated Values) files**
- **used in specific tools:**
  - spreadsheet like Excel or Calc (pivot table)
  - specialized statistic software like R or Python's Pandas

# Table format

- **layout:**

- *in rows:* each element (referring expressions, chains, texts...)
- *in columns:* annotations

- **example for texts:**

	A	B	C	D	E
1	id	work	token_count	mention_count	chain_count
2	aesop	literature	113	31	9
3	caesar	politics	335	101	61
4	cicero	politics	145	47	30
5	pliny	science	183	51	23

# Table format

- example for expressions (mentions):

	A	B	C	D	E	F	G	H	
1	id	start	stop	chain_name	chain_size	token_count	partofspeech	function	string
2	0	0	2	Peasant	11	2	i noun with indefinite article	s subject	A Peasant
3	2	3	9	Eagle	8	6	i noun with indefinite article	o object	an Eagle captured in a trap
4	1	7	9	M3	1	2	i noun with indefinite article	a adverbial	a trap
5	3	13	15	Eagle	8	2	d noun with definite article	o object	the bird
6	4	17	18	Peasant	11	1	s personal pronoun	o object	him
7	5	0	2	Eagle	8	2	d noun with definite article	s subject	The Eagle
8	7	7	9	Peasant	11	2	n noun with determiner	o object	his deliverer
9	6	7	8	Eagle	8	1	e possessive adjective	t other	his
10	10	12	22	Peasant	11	10	d noun with definite article	o object	the Peasant sitting under a wall w

# Table format

- example for chains:

	A	B	C	D	E	F	G	H
1	id	text_id	name	first_mention_id	last_mention_id	text_chain_index	mention_count	unique_chain_name
2	7	aesop	M24	23	23	7	1	aesop:M24
3	4	aesop	M14	14	14	4	1	aesop:M14
4	3	aesop	Wall	9	24	3	4	aesop:Wall
5	2	aesop	M3	1	1	2	1	aesop:M3
6	1	aesop	Eagle	2	30	1	8	aesop:Eagle
7	6	aesop	M17	17	17	6	1	aesop:M17
8	8	aesop	M29	28	28	8	1	aesop:M29
9	0	aesop	Peasant	0	29	0	11	aesop:Peasant

# List of tables

- ***tokens***
- ***sentences*** with annotations like the number of tokens, mentions, chains, etc.
- ***paragraphs*** with annotations like the number of tokens, mentions, chains, etc.
- ***texts*** with annotations like the number of tokens, mentions, chains, etc.
- ***chains*** with annotations like the number of mentions, etc.
- ***mentions*** with annotations like the name of the chain, the size of the chain, and all the features annotated with SACR, etc.
- ***relations*** with annotations like the distance between two mentions, etc. There are several types of relations:
  - first: A-B, A-C, A-D...
  - consecutive: A-B, B-C, C-D...
  - all: both first and consecutive relations



# Example of columns

- **Examples of columns in “mentions.csv”:**

id	text_sent_index	is_first_in_chain
start	par_sent_index	is_last_in_chain
stop	sent_mention_index	first_in_text
chain_name	par_mention_index	last_in_text
chain_size	text_mention_index	first_in_par
token_count	par_start	last_in_par
partofspeech	par_stop	first_in_sent
function	text_start	last_in_sent
string	text_stop	in_first_par
head	chain_id	in_last_par
text_id	text_mention_rank	in_first_sent
par_id	par_mention_rank	in_last_sent
sent_id	sent_mention_rank	is_singleton
text_par_index	chain_mention_index	unique_chain_name

# Databases

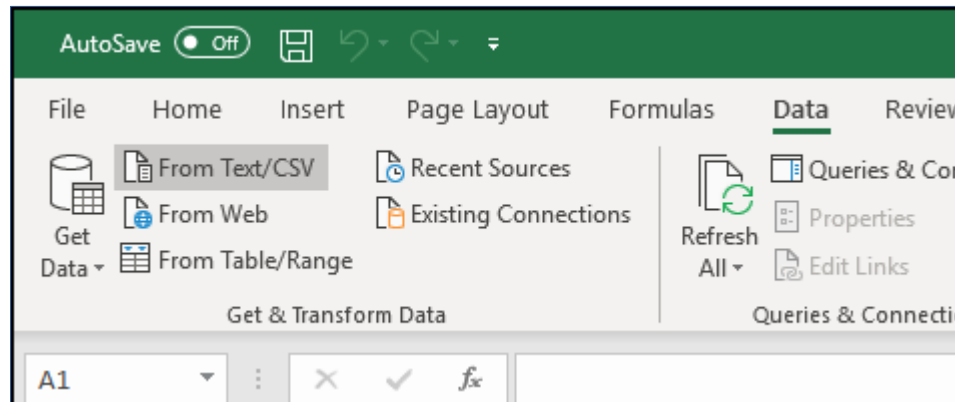
- **Each element of a table is identified by a unique key in the corpus, so the whole set of tables make a relational database**
- **You can use database operations to:**
  - extend annotations further
  - link annotations and textual structures in the corpus (sentences, paragraphs, texts)

# Getting the tables

- **Go to [b-oberle.com/tools/coref/sacr2df/](http://b-oberle.com/tools/coref/sacr2df/) and fill out the form will all the pre-annotated files**
- **Or download the zip at [boberle.com/projects/sacr/preannotated\\_tables.zip](http://boberle.com/projects/sacr/preannotated_tables.zip)**
- **Unzip the file**

# Opening a CSV file with Excel

- **Depending on the version of Excel, you need to:**
  - open a new spreadsheet,
  - select “From text/csv”



# Opening a CSV file with Excel

- ...
  - and follow the instructions

chains.csv

File Origin: 1252: Western European (Windows) | Delimiter: Comma | Data Type Detection: Based on first 200 rows

id	text_id	name	first_mention_id	last_mention_id	text_chain_index	mention_count	unique_chain_name
4	aesop	M14	14	14	4	1	aesop:M14
6	aesop	M17	17	17	6	1	aesop:M17
3	aesop	Wall	9	24	3	4	aesop:Wall
2	aesop	M3	1	1	2	1	aesop:M3
8	aesop	M29	28	28	8	1	aesop:M29
1	aesop	Eagle	2	30	1	8	aesop:Eagle
0	aesop	Peasant	0	29	0	11	aesop:Peasant

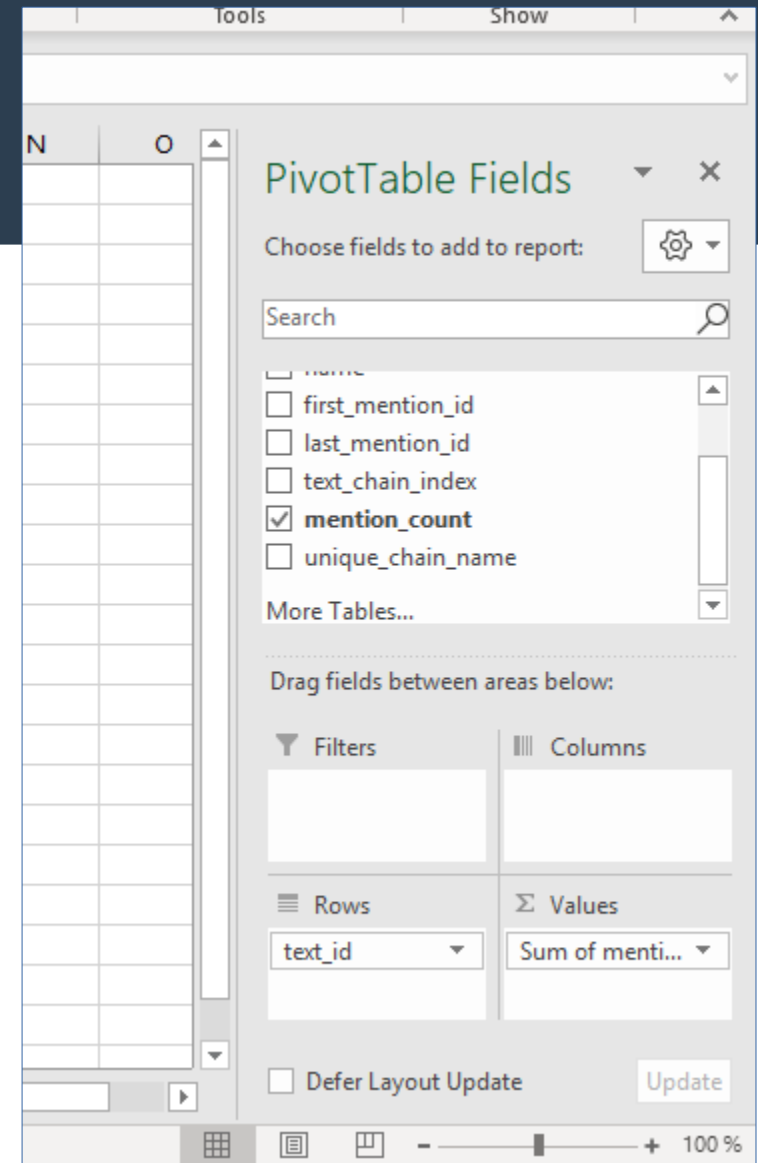
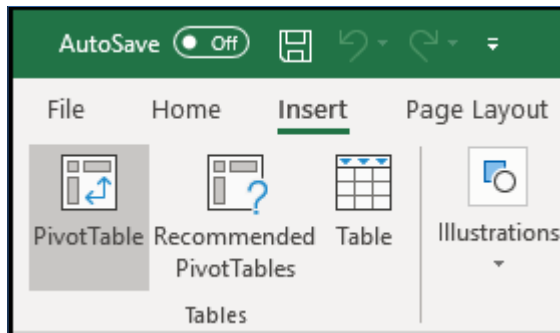
# Opening a CSV file with Excel

- Open the “chains” table in Excel

	A	B	C	D	E	F	G	H
1	id	text_id	name	first_mention_id	last_mention_id	text_chain_index	mention_count	unique_chain_name
2	4	aesop	M14	14	14	4	1	aesop:M14
3	6	aesop	M17	17	17	6	1	aesop:M17
4	3	aesop	Wall	9	24	3	4	aesop:Wall
5	2	aesop	M3	1	1	2	1	aesop:M3
6	8	aesop	M29	28	28	8	1	aesop:M29
7	1	aesop	Eagle	2	30	1	8	aesop:Eagle
8	0	aesop	Peasant	0	29	0	11	aesop:Peasant
9	7	aesop	M24	23	23	7	1	aesop:M24
10	5	aesop	Bundle	15	21	5	3	aesop:Bundle

# Creating a Pivot Table

- **Select the whole table**
- **Insert > Pivot Table**
- **A left-hand pane shows up**



# Anatomy of the Pivot Table form

- **We want the average size of a chain (number of mention) for each text**
  - Drag and drop fields to the boxes
  - Change the aggregate function to “average”

The image illustrates the configuration of a PivotTable. It features three main components:

- Value Field Settings Dialog:** A central dialog box titled "Value Field Settings" with the following details:
  - Source Name: mention\_count
  - Custom Name: Average of mention\_count
  - Summarize Values By: Show Values As
  - Summarize value field by: Average (highlighted in the list)
  - Buttons: Number Format, OK, Cancel
- PivotTable Layout:** A grid with four areas: Filters, Columns, Rows, and Values.
  - Rows: text\_id
  - Values: Average of m...
- Field List:** A panel on the left titled "Drag fields between areas below:" with a list of fields including mention\_count.

Arrows indicate the workflow: one arrow points from the "Average" option in the dialog to the "Average of m..." field in the Values area, and another arrow points from the "Average of m..." field in the Values area to the "Average of mention\_count" custom name in the dialog.

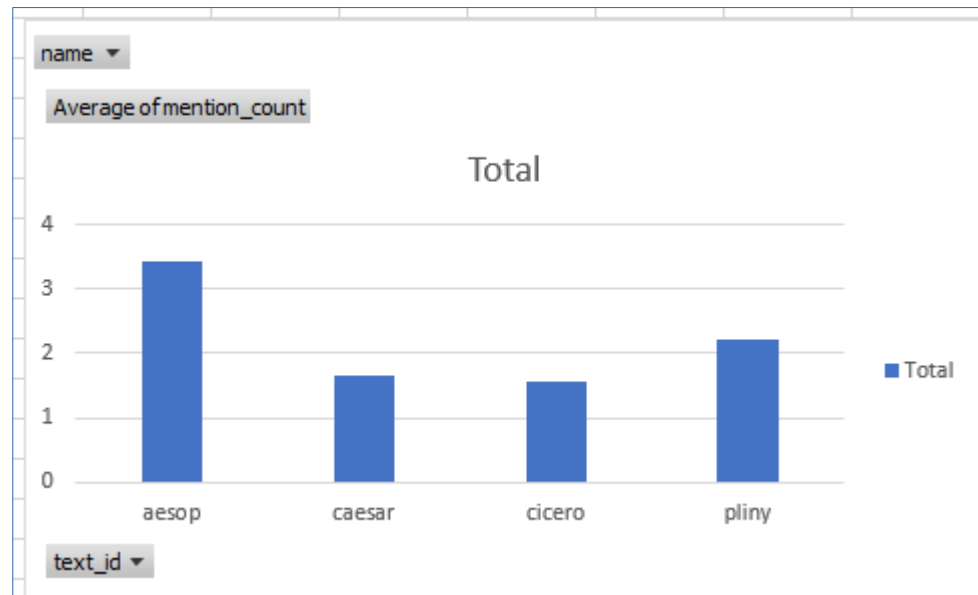


# The table

- **The pivot table:**

Row Labels	Average of mention_count
aesop	3,444444444
caesar	1,655737705
cicero	1,566666667
pliny	2,217391304
<b>Grand Total</b>	<b>1,869918699</b>

- **You can create a graph:**



# Adding a filter

- Add “name” (of the chain) as filter
- Select “Caesar” value: this take the chain name “Caesar” into account

Drag fields between areas below:

<b>Filters</b>	<b>Columns</b>
name	
<b>Rows</b>	<b>Values</b>
text_id	Average of m...

Search

- (All)
- Britain
- Bundle
- Caesar
- Cicero
- Crassus
- Eagle
- Latin
- M1

Select Multiple Items

OK Cancel

	A	B
1	name	Caesar
2		
3	Row Labels	Average of mention_count
4	caesar	23
5	Grand Total	23

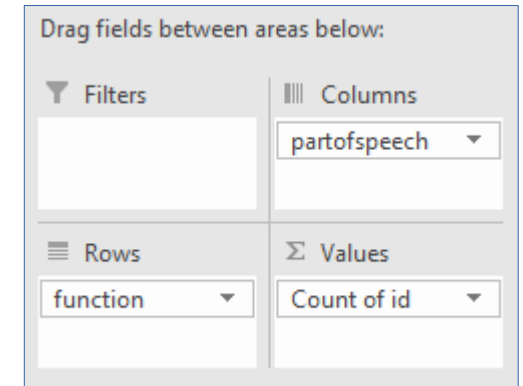
# Adding a filter

- **Now select the chains Caesar, Pompey and Crassus (the triumvirat) in the filters**
  - the number is the average size of the chains Caesar, Pompey and Crassus in the text “Caesar”

	A	B
1	unique_chain_name (Multiple Items)	
2		
3	Row Labels	Average of mention_count
4	caesar	9,666666667
5	Grand Total	9,666666667

# Getting results on mentions

- Now we want to use the “mentions” table to see the relations between part of speech and grammatical function



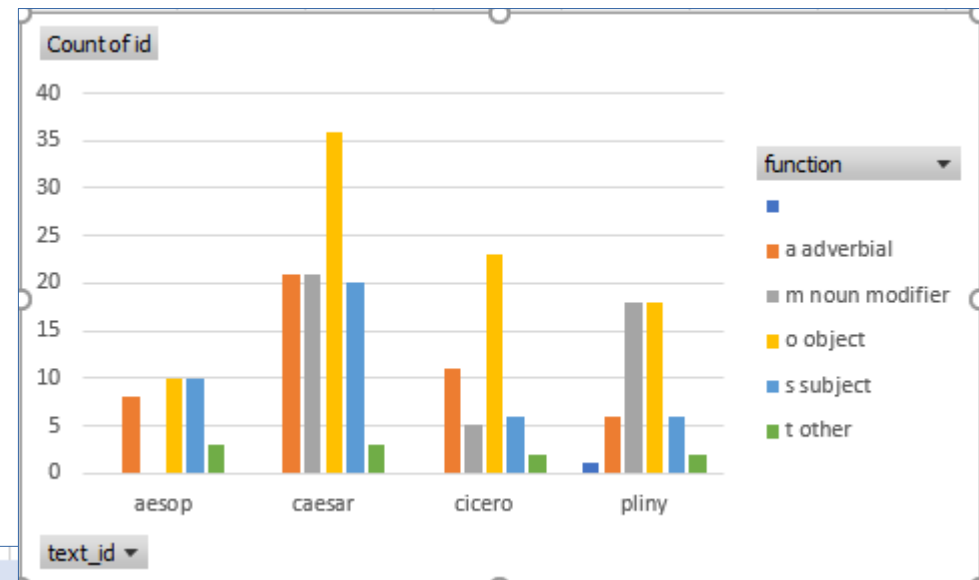
Count of id	Column Labels	a name	d noun	e posse	i noun	w in noun	r relatives	person	t noun	w	Grand Total
	1										1
a adverbial		5	15			6	4	1	2	13	46
m noun modifier		10	7	14			5		2	6	44
o object		6	19			22	17	6	8	9	87
s subject		13	7			1	4	7	10		42
t other		1		3						6	10
Grand Total	1	35	48	17		29	30	14	22	34	230

# Grammatical functions by text

- See the grammatical functions by text

Drag fields between areas below:

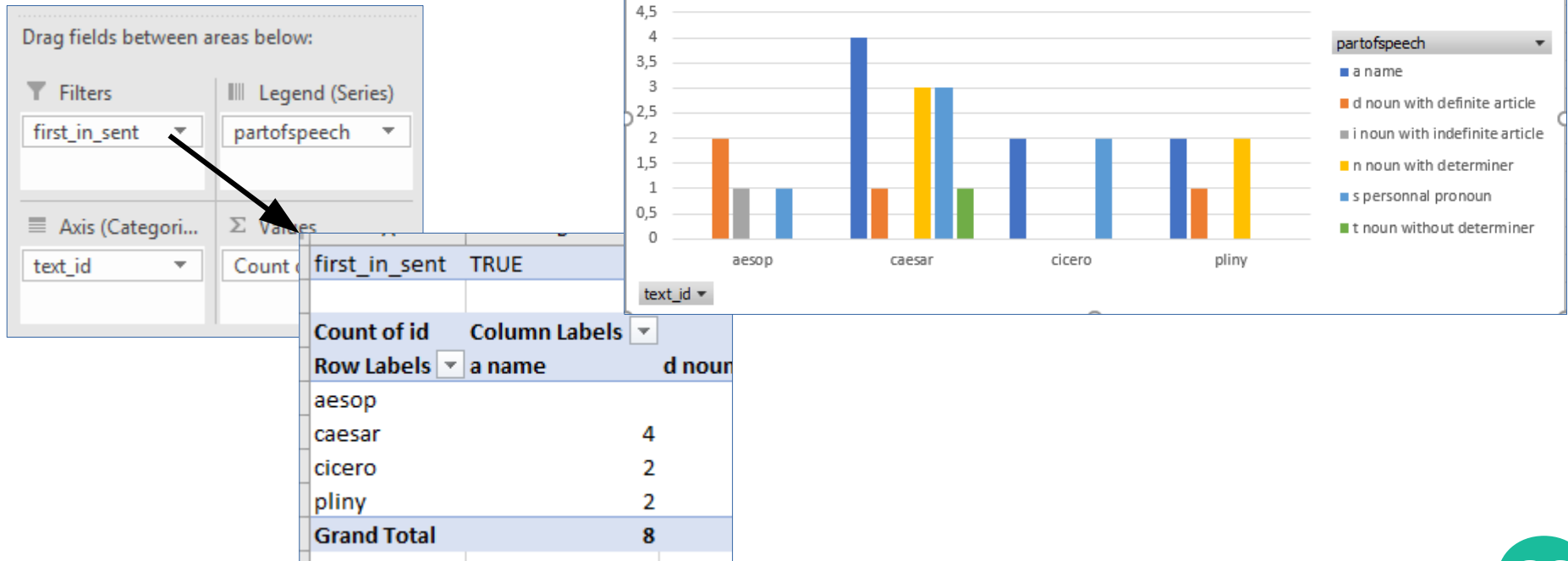
<b>Filters</b>	<b>Columns</b>
	function
<b>Rows</b>	<b>Values</b>
text_id	Count of id



Count of id	a adverbial	m noun modifier	o object	s subject	t other	Grand Total
aesop	8		10	10	3	31
caesar	21	21	36	20	3	101
cicero	11	5	23	6	2	47
pliny	1	6	18	18	6	51
Grand Total	1	46	44	87	10	230

# Adding a filter

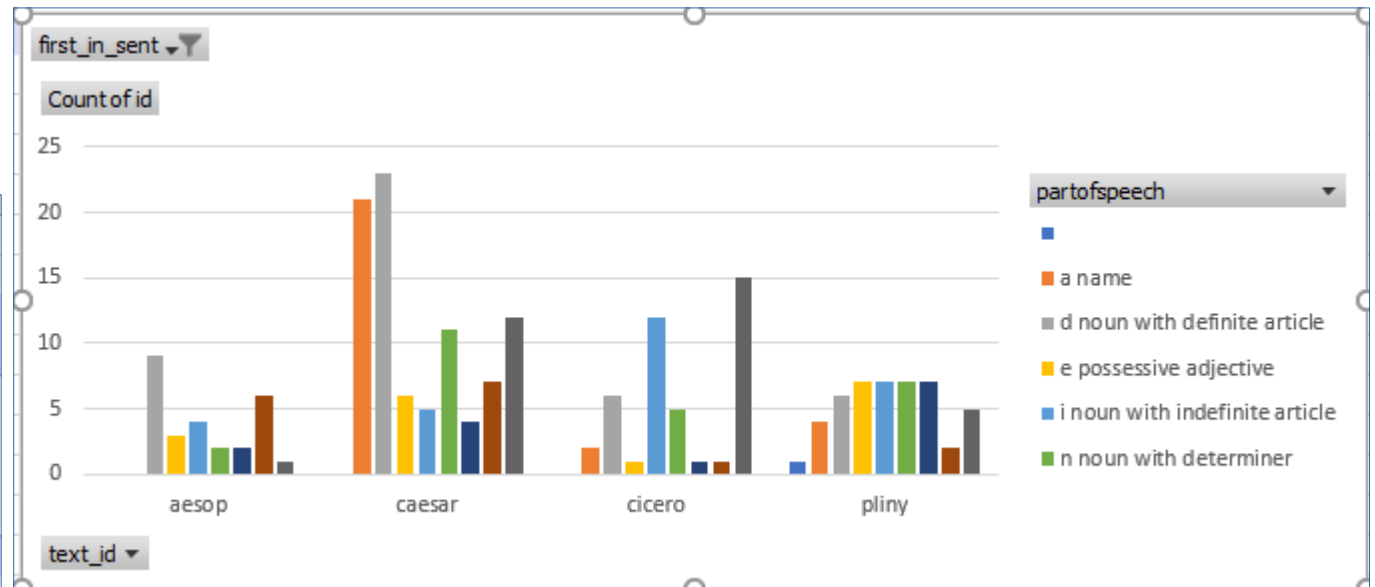
- **Functions by text, but with a filter to limit the results to only the first mentions of every sentence**



# Adding a filter

- Select “false” to get mentions NOT in the first place of each sentence

first_in_sent	FALSE	
Count of id		Column Labels
Row Labels		a name d
aesop		
caesar		21
cicero		2
pliny	1	4
Grand Total	1	27



# Other filters

- **You can add other filters, for example:**
  - filtering referring expressions by the size of the chain: expressions in short/long chains
  - filtering referring expressions by their length (token count): short/long expressions
  - etc.



# With LibreOffice Calc

- You can do much the same with Calc

Count - id	Data										Total Result
function	a name	d noun with d	e possessive	i noun with in	n noun with d	r relative pr	s personal p	t noun with	(empty)		
a adverbial	5	15		6	4	1	2	13			46
m noun modi	10	7	14								
o object	6	19		22							
s subject	13	7		1							
t other	1		3								
(empty)											
<b>Total Result</b>	<b>35</b>	<b>48</b>	<b>17</b>	<b>29</b>							

Pivot Table Layout

Page Fields:

Column Fields:

- Data
- partofspeech

Row Fields:

- function

Data Fields:

- Count - id

Available Fields:

- id
- start
- stop
- function
- head
- partofspeech
- chain\_name
- string
- text\_id
- par\_id
- sent\_id
- text\_par\_index
- text\_sent\_index
- par\_sent\_index
- sent\_mention\_index
- par\_mention\_index
- text\_mention\_index
- par\_start
- par\_stop
- text\_start
- text\_stop
- chain\_id
- text\_mention\_rank
- par\_mention\_rank
- sent\_mention\_rank

Drag the Items into the Desired Position

Options

Source and Destination

Help OK Cancel

# With LibreOffice Calc

Count - id	partofspe						
text_id	a name	d noun with o	i noun with in	n noun with o	s personal p	t noun without	Total Result
aesop			2	1			
caesar		4	1			3	
cicero		2					
pliny		2	1				2
<b>Total Result</b>		<b>8</b>	<b>4</b>	<b>1</b>		<b>5</b>	

**Pivot Table Layout**

**Page Fields:**  
first\_in\_sent

**Column Fields:**  
partofspeech

**Row Fields:**  
text\_id

**Data Fields:**  
Count - id

**Available Fields:**  
par\_mention\_index  
text\_mention\_index  
par\_start  
par\_stop  
text\_start  
text\_stop  
chain\_id  
text\_mention\_rank  
par\_mention\_rank  
sent\_mention\_rank  
chain\_mention\_index  
is\_first\_in\_chain  
is\_last\_in\_chain  
first\_in\_text  
last\_in\_text  
first\_in\_par  
last\_in\_par  
first\_in\_sent  
last\_in\_sent  
in\_first\_par  
in\_last\_par  
in\_first\_sent  
in\_last\_sent  
is\_singleton  
chain\_size

Drag the Items into the Desired Position

Options  
Source and Destination

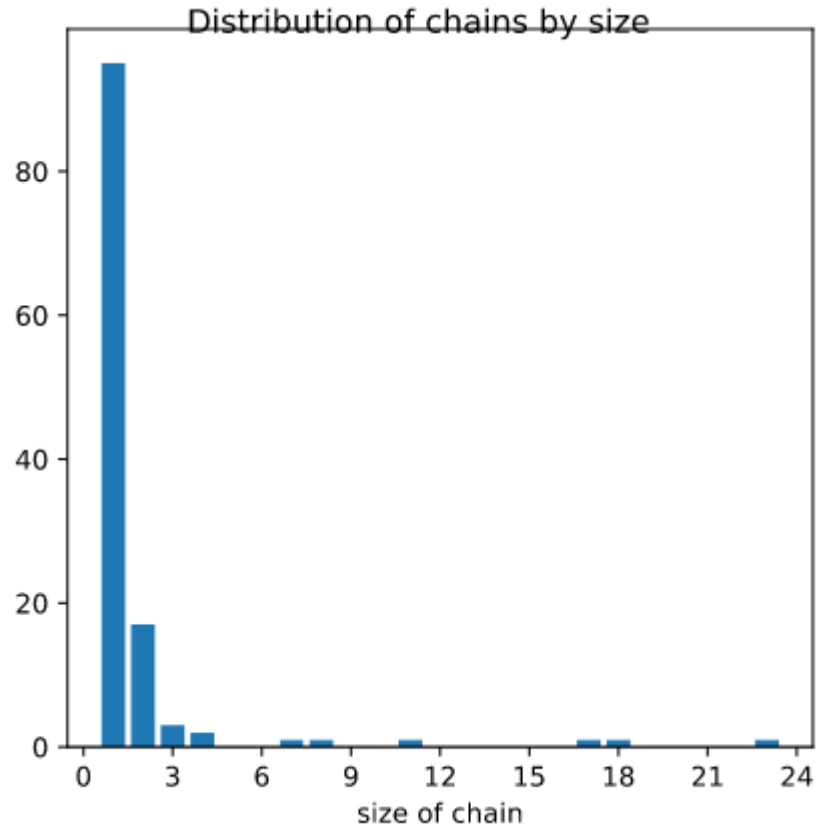
Help OK Cancel

# Using an online tool

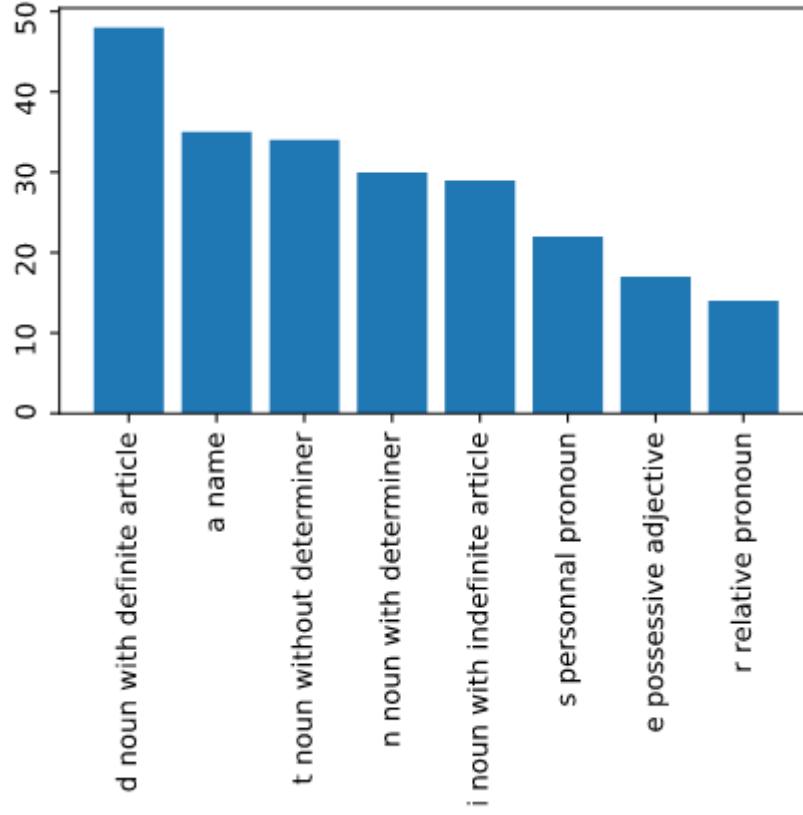
# Online version of CRViewer

- **Go to [b-oberle.com/tools/coref/manage\\_files/](https://b-oberle.com/tools/coref/manage_files/)**
- **Upload the zip file with all the tables (or create a new one with the SACR files)**
- **Choose the graph parameters and click “draw”**

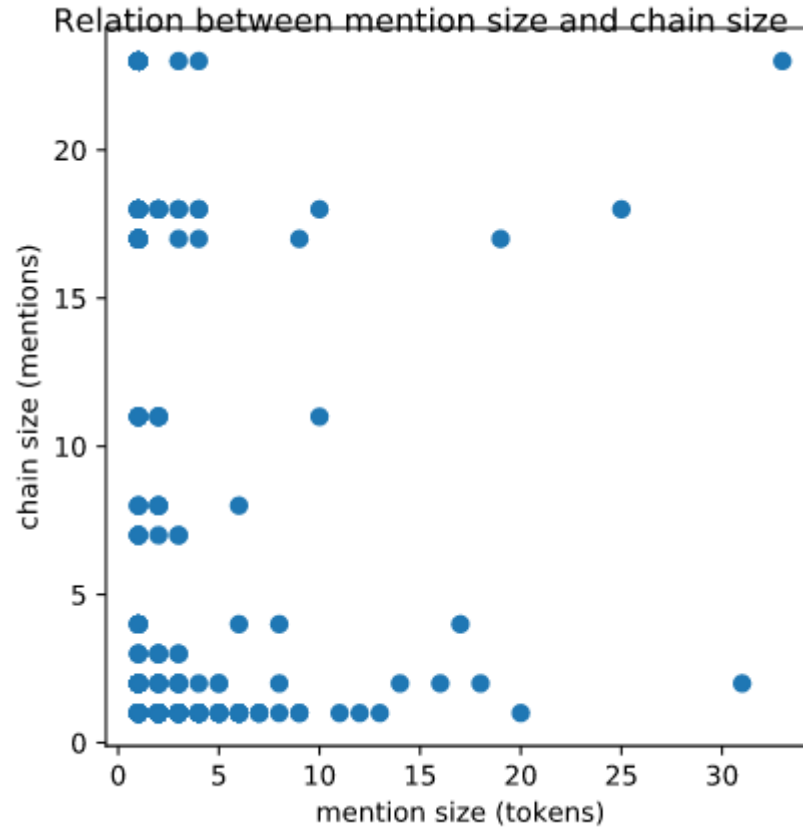
# Distribution



# Distribution

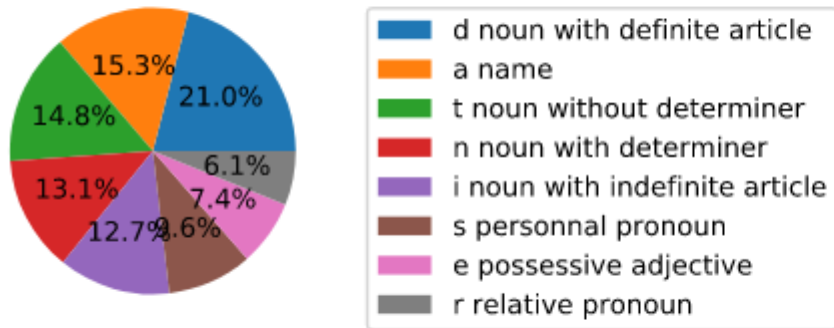


# Scatter plot

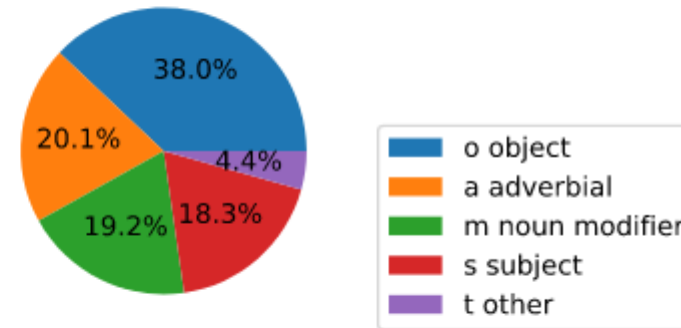


# Pie Charts

part of speech



function





Thank you for your attention!