cuneiform open science

data science with text and annotations

"Securing data in Mesopotamia"

Lorentz Center, Leiden, 2022-03-14/18

Dirk Roorda

KNAW Humanities Cluster

overview

tablets and text processing:

human→digital→computational→linguistic

towards cuneiform open science

concept→practice→enabling tech

tools of the trade

repository→notebooks→data science libraries→data cycling

outlook

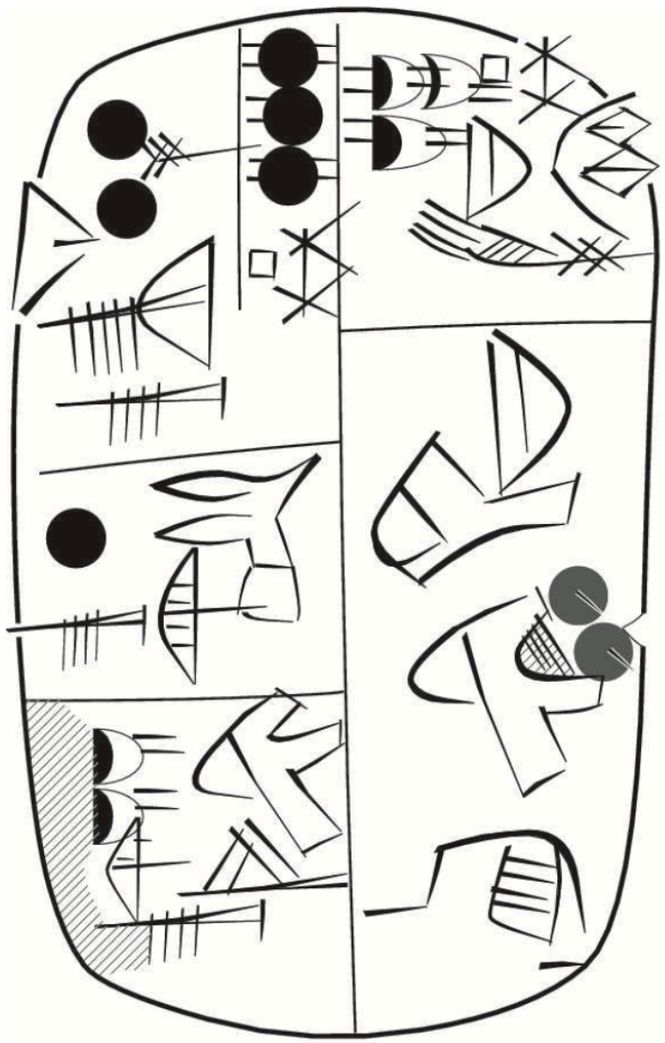
tablets-as-data

KNAW
Humanities
Cluster

tablets and text processing

photo

line art



ATF

```
&P005381 = MSVO 3, 70
#atf: lang qpc
@obverse
@column 1
1.a. 2(N14) , SZE~a SAL TUR3~a
NUN~a
1.b. 3(N19) , |GISZ.TE|
2. 1(N14) , NAR NUN~a SIG7
3. 2(N04)# , PIRIG~b1 SIG7 URI3~a
NUN~a
@column 2
1. 3(N04) , |GISZ.TE| GAR |SZU2.
((HI+1(N57))+(HI+1(N57)))| GI4~a
2. , GU7 AZ SI4~f
@reverse
@column 1
1. 3(N14) , SZE~a
2. 3(N19) 5(N04) ,
3. , GU7
@column 2
1. , AZ SI4~f
```

source: Bob Englund et al,.cdli

KNAW Humanities Cluster

text processing in the digital world

- observe with your own eyes
- read the ATF
- grep ATF files that you have locally
- count patterns, gather examples, compare contexts
- jumble it into XML or JSON
- call for assistance

KNAW Humanities Cluster

computational text processing

easy

task + search = results

Numerals

We want to find all the ShinPP numerals.

```
shinPP = dict(
    N41=0.2,
    N04=1,
    N19=6,
    N46=60,
    N36=180,
    N49=1800,
)

shinPPPat = "|".join(shinPP)
```

```
query = f"""
tablet
  sign grapheme={shinPPPat}
"""
results = A.search(query)
A.table(results, end=20, showGraphics=True)
```

0.11s 1018 results

n	p	tablet	sign
1	P448701 obverse:1:1	P448701	1(N46)
2	P448701 obverse:1:1	P448701	2(N19)
3	P448701 obverse:1:1	P448701	4(N41)
4	P006005 obverse:2:1	P006005	1(N04)
5	P002329 obverse:1:5	P002329	2(N19)
6	P002342 obverse:3:2	P002342	1(N36)
7	P002342 obverse:3:2	P002342	2(N19)
8	P002344 obverse:3:3	P002344	1(N04)
9	P002398 obverse:2:3	P002398	1(N04)
10	P002622 obverse:1:1	P002622	5(N19)
11	P002622 obverse:2:1	P002622	1(N46)
12	P002622 obverse:2:1	P002622	4(N19)
13	P002626 obverse:1:2	P002626	1(N41)
14	P003330 reverse:1:1	P003330	3(N46)
15	P003330 reverse:1:1	P003330	2(N49)
16	P003330 reverse:1:1	P003330	5(N19)
17	P003330 reverse:1:1	P003330	2(N04)
18	P003330 reverse:1:1	P003330	1(N41)
19	P003357 obverse:1:2	P003357	1(N04)
20	P003542 obverse:2:5	P003542	1(N04)

source: uruk/tutorial/calc

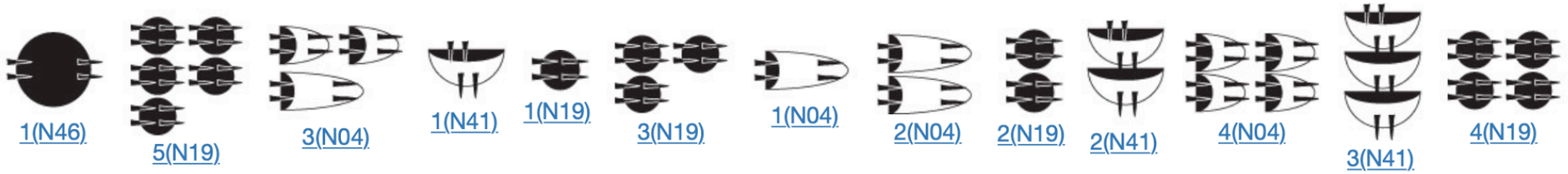
Calculate once

source: uruk/tutorial/calc

```
calcTablet("P006377")
```

P006377
no lineart for tablet `P006377`

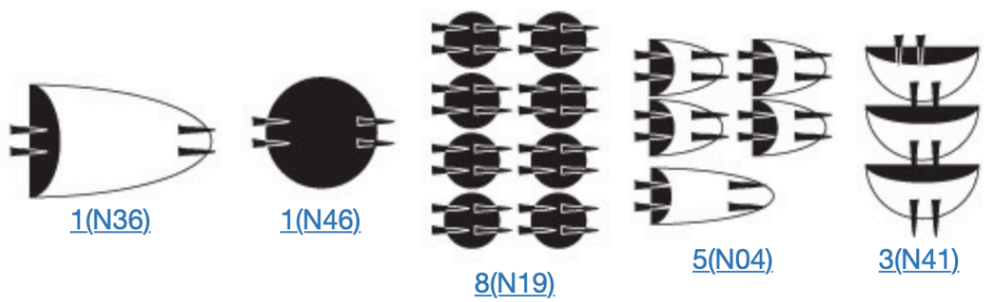
obverse




```
1 x 1(N46) = 1 x 1 x 60 = 60
1 x 5(N19) = 1 x 5 x 6 = 30
4 x 3(N04) = 4 x 3 x 1 = 12
2 x 1(N41) = 2 x 1 x 0.2 = 0.4
8 x 1(N19) = 8 x 1 x 6 = 48
2 x 3(N19) = 2 x 3 x 6 = 36
5 x 1(N04) = 5 x 1 x 1 = 5
3 x 2(N04) = 3 x 2 x 1 = 6
3 x 2(N19) = 3 x 2 x 6 = 36
1 x 2(N41) = 1 x 2 x 0.2 = 0.4
2 x 4(N04) = 2 x 4 x 1 = 8
1 x 3(N41) = 1 x 3 x 0.2 = 0.6000000000000001
1 x 4(N19) = 1 x 4 x 6 = 24
```

total = 266.4

reverse




```
1 x 1(N36) = 1 x 1 x 180 = 180
1 x 1(N46) = 1 x 1 x 60 = 60
1 x 8(N19) = 1 x 8 x 6 = 48
1 x 5(N04) = 1 x 5 x 1 = 5
1 x 3(N41) = 1 x 3 x 0.2 = 0.6000000000000001
```

total = 293.6

intermediate

hypothesis + calculation = results

Calculate ad lib

Now the first 5 tablets.

```
for tablet in sorted(pNums)[0:5]:
    calcTablet(tablet)
```

source: oldbabylonian/tutorial/cookbook/words

linguistic text processing

difficult

ideas + data = new data

i-na + ...-?im

We look for word pairs, of which the first is `i-na` and the second ends in a sign whose reading ends in `im`.

```
query = """
line
  word
    =: sign reading=i
    <: sign reading=na
    :=
  <: word
    := sign reading~im$
"""
```

```
results = A.search(query)
```

```
  0.72s 307 results
```

```
A.table(results, end=10)
```

n	p	line	word	sign	sign	word	sign
1	P509375 reverse:9	i-na la-hi-a-nim	i-na	i-	na	la-hi-a-nim	nim
2	P510527 obverse:6	{disz}ip-qu2-i3-li2-szu _di-ku5_ i-na pu-uh2-ri-im	i-na	i-	na	pu-uh2-ri-im	im
3	P510527 obverse:15	i-na pu-uh2-ri-im i-na da-ba-bi-im	i-na	i-	na	pu-uh2-ri-im	im
4	P510527 obverse:15	i-na pu-uh2-ri-im i-na da-ba-bi-im	i-na	i-	na	da-ba-bi-im	im
5	P510538 obverse:10	i-na tam-li-tim	i-na	i-	na	tam-li-tim	tim
6	P510562 obverse:7	i-na# pa-ni-tim a-na a-<ma?>-az{ki} ta-al-li-ik-ma#	i-na#	i-	na#	pa-ni-tim	tim
7	P510567 reverse:7	[i-na] e-bu-ri-im	[i-na]	[i-	na]	e-bu-ri-im	im
8	P510571 reverse:13	i-na an-ni-tim at-hu-<ut>-ka#	i-na	i-	na	an-ni-tim	tim
9	P510574 obverse:8	tup-pi2 i-na a-ma-ri-im	i-na	i-	na	a-ma-ri-im	im
10	P510575 obverse:11	[i]-na# qa-tim ta-ki-il-tim	[i]-na#	[i]-	na#	qa-tim	tim

linguistic text processing

difficult

ideas + data = new data

Part of Speech tagging

Status

- 2019-06-06 Personal pronouns added
- 2019-06-05 Dirk has reorganised the messy code after the sprint into a repeatable and documented workflow. The workflow covers special cases, prepositions, and nouns, not yet the extra insights of the sprint.
- 2019-06-03/04 Martijn, Alba and Dirk do a two-day sprint to follow-up on heuristics supplied by Cale Johnson. Martijn and Alba provide extra insights.

Features

6 TF features saved: cs, gn, nu, pos, ps, subpos.

feature	%	number of nodes
cs	2	1440
gn	2	1440
nu	2	1440
pos	54	41624
ps	2	1440
subpos	14	10971

10 categories.

category	%	number of nodes
none	46	34881
all	54	41622
noun-	32	24322
prep-	8	5943
pcl-conj	3	2570
pcl-rel	3	2363
noun-numeral	3	2238
pcl-neg	2	1909
prn-prs	2	1440
adv-tmp	1	399
pcl-	1	388
prn-dem	0	52

source: oldbabylonian/tutorial/cookbook/posTag

towards cuneiform open science

- concept: ***datamodel*** for text plus annotations
    - clean separation of text and annotations
- practice: annotating is an activity that results in a ***work***:
    - it has an author
    - it can be found online
    - it can be combined
        - with the original text
        - and with ***other peoples'*** annotations
- enabling tech: ***text-fabric*** is a generic library that provides this
    - it has been applied to various ancient texts, e.g. the Hebrew Bible

KNAW
Humanities
Cluster

concept: data model

Data structure of TF - the IKEA spirit

order!		order!
stacks	of	components
uniquely		identified

concept: one feature per file

atf.tf

```
@node
@converters=Cale Johnson, Dirk Roorda
@description=full atf of a sign
@editor=Cale Johnson et al.
@name=NinMed Medical Texts from Nineveh
@project=BabMed
@valueType=str
@writtenBy=Text-Fabric
@dateWritten=2022-02-10T09:59:19Z

DU₃
DU₃
BI
...
5
KA#
INIM
MA
x
x
x
x
EN₂
su
ub
hur
ri
im#
su#
ub
...
ša₂
sa
ku
tu₂
hi
```

grapheme.tf

```
@node
@converters=Cale Johnson, Dirk Roorda
@description=grapheme of a sign
@editor=Cale Johnson et al.
@name=NinMed Medical Texts from Nineveh
@project=BabMed
@valueType=str
@writtenBy=Text-Fabric
@dateWritten=2022-02-10T09:59:19Z

DU₃
DU₃
BI
6   KA
INIM
MA
13  EN₂
38  KA
INIM
MA
GIG
GIR
ZI
45  DU₃
DU₃
BI
SIG₂
SA₅
51  EN₂
54  GIM
59  KA₂
78  DU₃
DU₃
BI
82  ŠIM
IGI
```

reading.tf

```
@node
@converters=Cale Johnson, Dirk Roorda
@description=reading of a sign
@editor=Cale Johnson et al.
@name=NinMed Medical Texts from Nineveh
@project=BabMed
@valueType=str
@writtenBy=Text-Fabric
@dateWritten=2022-02-10T09:59:19Z

14  su
ub
hur
ri
im
su
ub
22  ša
sa
ku
tu
hi
si
a
pi
il
lat
aš
kur
ba
an
ni
44  hi
52  ma
mit
55  šar
ra
```

concept: corpora

master ninmed / tf / 0.3 /

dirkroorda improved dataversion 0.3

..

after.tf
atf.tf
atfpost.tf
atfpre.tf
col.tf
collated.tf
collection.tf
colofon.tf
comment.tf
damage.tf
description.tf
det.tf
docnumber.tf
erasure.tf
excised.tf
face.tf
flags.tf

master oldbabylonian / tf / 1.0.6 /

Dirk Roorda tf data release 1.0.6

..

ARK.tf
after.tf
afterr.tf
afteru.tf
atf.tf
atfpost.tf
atfpre.tf
author.tf
col.tf
collated.tf
collection.tf
comment.tf
damage.tf
det.tf
docnote.tf
docnumber.tf
excavation.tf

master oldassyrian / tf / 0.1 /

Switch branches or tags

Dirk Roorda first data version

..

ARK.tf
after.tf
afterr.tf
afteru.tf
atf.tf
atfpost.tf
atfpre.tf
author.tf
col.tf
collection.tf
comment.tf
damage.tf
det.tf
docnote.tf
docnumber.tf
excavation.tf

master uruk / tf / uruk / 1.0 /

Dirk Roorda depth

..

catalogId.tf
comments.tf
crossref.tf
damage.tf
depth.tf
excavation.tf
fragment.tf
fullNumber.tf
grapheme.tf
identifier.tf
modifier.tf
modifierFirst.tf
modifierInner.tf
name.tf

KNAW Humanities Cluster

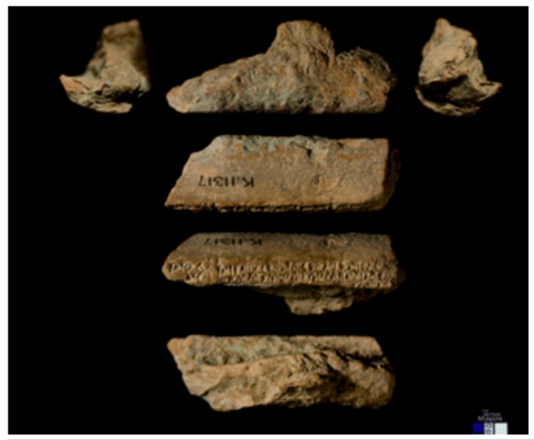
practice: annotations as work

-

with author and provenance

Nino-cunei / ninmed Public

README.md



Nino-cunei

archived repository | DOI 10.5281/zenodo.6034063 | repo status WIP

Cuneiform corpora in Text-Fabric

This repo is a research environment for the study of cuneiform tablets. You can run your own programs off-line, and publish your work in online notebooks.

Corpus

This repo contains transliterations of texts of the Nineveh Medical Encyclopedia (ca. 800 BCE).

See also about and transcription.

Software

The main processing tool is Text-Fabric. It is instrumental to turn the analysis of ancient data into computing narratives.

The ecosystem is Python and Jupyter notebooks.

Getting started

Start with the tutorial.

Authors

- Dirk Roorda at DANS
- J. Cale Johnson at the Institut für Wissensgeschichte des Altertums, Freie Universität Berlin

source: Nino-cunei/ninmed

enabler: text-fabric

source: Nino-ninmed/tutorial/start

```
A = use("Nino-cunei/ninmed", hoist=globals())
```

TF-app: ~/text-fabric-data/Nino-cunei/ninmed/app

data: ~/text-fabric-data/Nino-cunei/ninmed/tf/0.3

```
This is Text-Fabric 9.2.5
Api reference : https://annotation.github.io/text-fabric/tf/cheatsheet.html

48 features found and 0 ignored
```

Text-Fabric: Text-Fabric API 9.2.5, Nino-cunei/ninmed/app v3, Search Reference

Data: NINMED, Character table, Feature docs

Features:

▼ **Nineveh Medical Encyclopedia 800 BCE: Cuneiform tablets**

after	str	what comes after a sign or word (- or space)
atf	str	full atf of a sign
atfpost	str	cluster characters that follow a sign or word
atfpre	str	cluster characters that precede a sign or word
col	int	ATF column number
collated	int	whether a sign is collated (*)
collection	str	collection name from metadata field "collection"
colofon	str	colofon comment to a line
comment	str	comment to a line
damage	int	whether a sign is damaged

KNAW Humanities Cluster

source: Nino-ninmed/tutorial/start

enabler: text-fabric

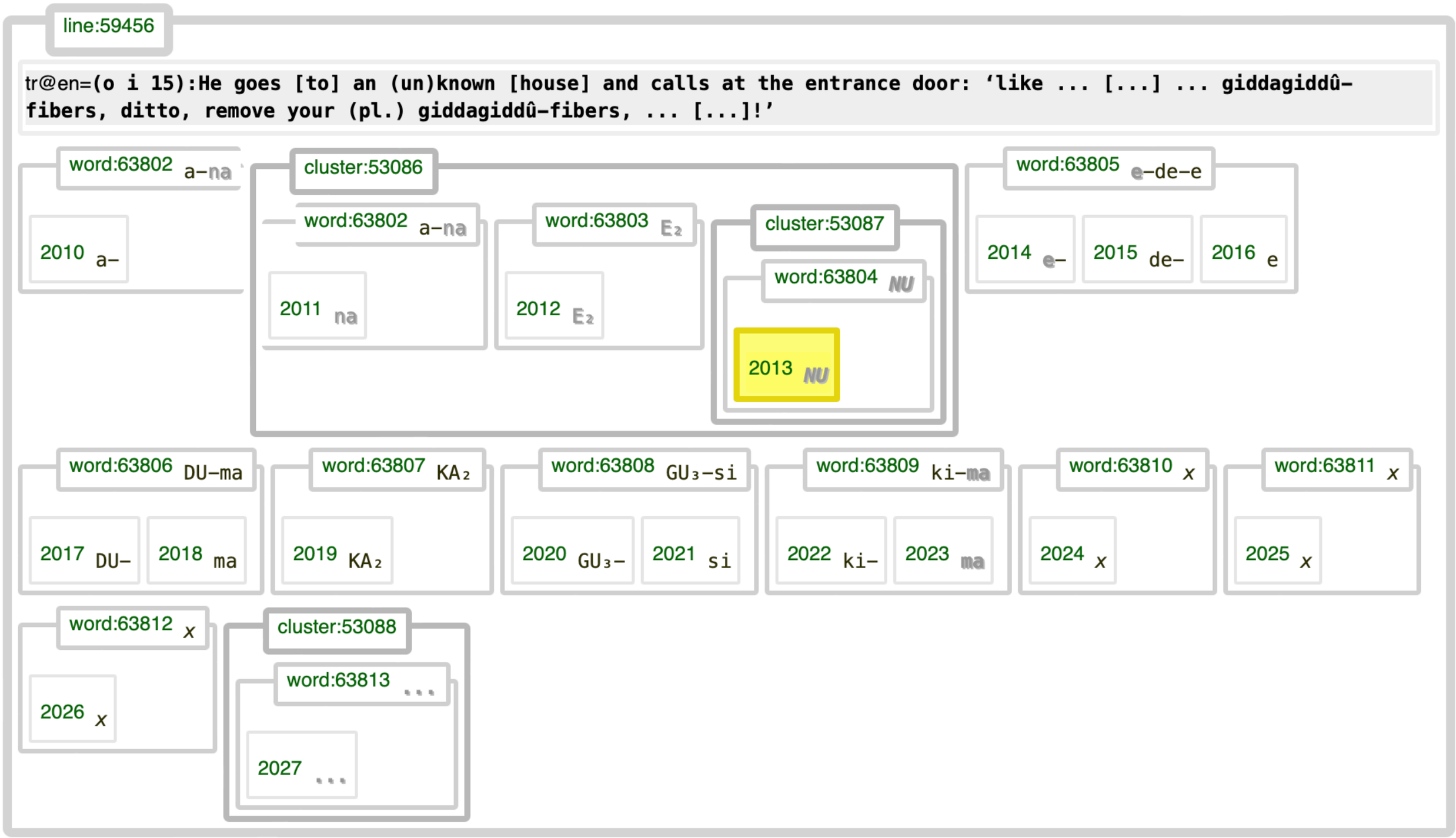
```
F.lemma.freqList()[0:20]
```

```
(('', 8538),
 ('ina I', 1420),
 ('ana I', 647),
 ('šumma I', 520),
 ('sâku I', 500),
 ('awīlu I', 445),
 ('šikaru I', 326),
 ('libbu I', 319),
 ('u I', 287),
 ('šamnu I', 287),
 ('mû I', 279),
 ('ša I', 262),
 ('šiptu I', 242),
 ('īnu I, -šu I', 239),
 ('lā I', 222),
 ('šatû II', 201),
 ('qû II', 195),
 ('balāṭu II', 194),
 ('kasû II', 180),
 ('šanû I', 178))
```

KNAW Humanities Cluster

enabler: text-fabric

P394104 obverse:1:14



source: Nino-ninmed/tutorial/start

KNAW Humanities Cluster

built to be findable, accessible, interoperable, reliable (FAIR)

tools of the trade: repository

Nino-cunei / ninmed Public

https://github.com/Nino-cunei/ninmed

authorship

master · 1 branch · 2 tags | Go to file | Add file | Code

dirkroorda about — 30f3c09 7 days ago — 21 commits

app	improving the signs	27 days ago
docs		7 days ago
parallels/tf	improved dataversion 0.3	26 days ago
programs		23 days ago
report	ion and valid tf generated	onth
source/json/0.1	conversion well underway	last month
tf	improved dataversion 0.3	26 days ago
tutorial		26 days ago
yaml	ns	27 days ago
.gitignore	first content	last month
LICENSE	Initial commit	2 months ago
README.md	improved dataversion 0.3	26 days ago

documentation

reproducible

versions

hands-on

About

Medical texts from Nineveh in Text-Fabric

Readme
MIT License
0 stars
0 watching
0 forks

Releases 2

Improved data version 0.3 Latest
26 days ago

+ 1 release

Packages

No packages published
Publish your first package

archived repository | DOI 10.5281/zenodo.6034063 | repo status WIP

archived & published! while still work in progress, ...

tools of the trade: notebooks

load your corpus with 1 command

show the provenance

write a query

execute the query

present query results

show it online

jupyter nbviewer

oldbabylonian / tutorial / cookbook

nbviewer/oldbabylonian/cookbook/ummama

Between `um-ma` and `-ma`

tell a computational story ... live!

What happens between `um-ma` and `ma` can help to identify proper nouns.

More precisely: we are looking for single words, immediately following the sign sequence `um-ma`, and where the word itself ends in `-ma`.

[1]:
```
import collections

from tf.app import use
```

[2]:
```
A = use("Nino-cunei/oldbabylonian", hoist=globals())
```

TF-app: ~/text-fabric-data/Nino-cunei/oldbabylonian/app
data: ~/text-fabric-data/Nino-cunei/oldbabylonian/tf/1.0.6
This is Text-Fabric 9.3.1
Api reference : https://annotation.github.io/text-fabric/tf/cheatsheet.html

67 features found and 0 ignored
Text-Fabric: Text-Fabric API 9.3.1, Nino-cunei/oldbabylonian/app v3, Search Reference
Data: OLDBABYLONIAN, Character table, Feature docs
Features:
▶ **Old Babylonian Letters 1900-1600: Cuneiform tablets**

Text-Fabric API: names N F E L T S C TF directly usable

The following query captures the intention of finding words after `um-ma` ending in `-ma`.

See basic relations for the meaning of `<:` and `:=`. You find them under **slot comparison**.

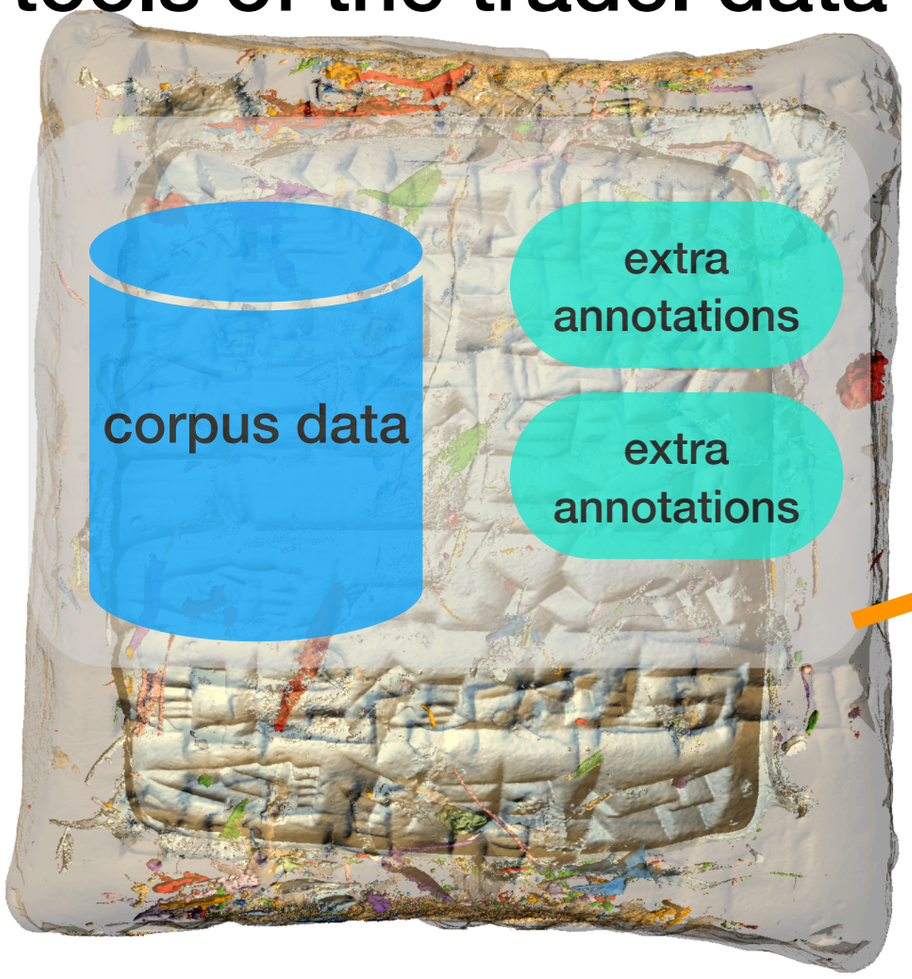
[3]:
```
query = """
line
   sign reading=um
   <: sign reading=ma
   <: word
     := sign reading=ma
"""
results = sorted(S.search(query))
print(f"{len(results)} results")
```

1472 results

[4]:
```
A.table(results, start=1000, end=1010, fmt="layout-orig-rich")
```

n	p	line		sign	sign		word	sign
1000	P386007 obverse:6	um-ma	šu-u₂-ma	um-	ma	šu-u₂-ma		ma
1001	P386008 obverse:3	um-ma	ha-am-mu-ra-pi₂-ma	um-	ma	ha-am-mu-ra-pi₂-ma		ma

tools of the trade: data science



process

crunch your data!

NumPy

pandas

nd easy to use open sour
of the Python programmi

Install pandas now!

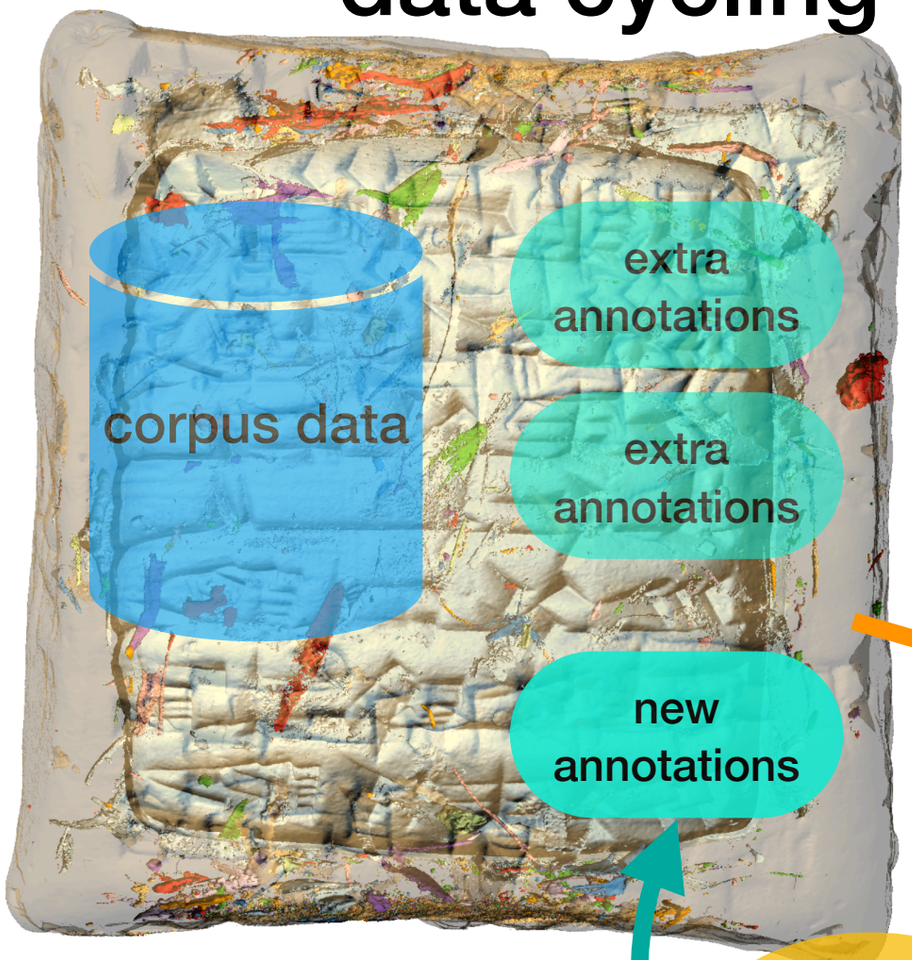
Natural Language Toolkit

export

R	S1	S2	S3	NODE1	TYPE1	TEXT1	lnno1	NODE2	TYPE2	TEXT2	type2	NODE3	TYPE3	TE
1	P510754	obverse	11	235994	line	5(diš) lu₂-meš an-nu-ti-in lu₂ la-ga-aš-ki-meš						11	207965	cl
2	P313375	obverse	10	245175	line	1(u) 8(diš) gin₂ ku₃-babbar wa-at-ra-am						10	215367	cluster 1(
3	P373035	reverse	3	255006	line	ša 5(diš) gin₂ ku₃-babbar gu₄ hi-a						3	223992	cluster ša
4	P373044	left	1:2	255237	line	1(u) 5(diš) ma-na siki hi-a ši-na						1:2	224274	cluster 1(
5	P509373	obverse	6	230793	line	diš-še-ep-d-suen a₂-gal₂ dumu um-mi-a-meš						6	203234	cl
6	P510550	obverse	4	231520	line	aš-šum ha-za-nu-um-sar	4	203984	cluster	ha-za-nu-um-sar				la
7	P510550	obverse	7	231523	line	šum-ma ha-za-nu-um-sar la ba-aš-lu-ma					7	203987	cluster	ha
8	P510550	obverse	9	231525	line	ha-za-nu-um-sar i-ka-am x x-al-la-x-ma					9	203989	cluster	ha
9	P510550	reverse	1	231528	line	a-na mi-nim ha-za-nu-um-sar					1	203992	cluster	ha-za-nu-u

KNAW Humanities Cluster

data cycling



load

process

crunch your data!

NumPy

pandas

nd easy to use open sourc
of the Python programmi

Install pandas now!

R

Natural Language Toolkit

display

export

share

research data cycle

R	S1	S2	S3	NODE1	TYPE1	TEXT1	lnno1	NODE2	TYPE2	TEXT2	type2	NODE3	TYPE3	TE
1	P510754	obverse	11	235994	line	5(diš) lu₂-meš an-nu-ti-in lu₂ la-ga-aš-ki-meš						11	207965	cl
2	P313375	obverse	10	245175	line	1(u) 8(diš) gin₂ ku₃-babbar wa-at-ra-am						10	215367	cluster 1(
3	P373035	reverse	3	255006	line	ša 5(diš) gin₂ ku₃-babbar gu₄ hi-a						3	223992	cluster ša
4	P373044	left	1:2	255237	line	1(u) 5(diš) ma-na siki hi-a ši-na						1:2	224274	cluster 1(
5	P509373	obverse	6	230793	line	diš-še-ep-d-suen a₂-gal₂ dumu um-mi-a-meš						6	203234	cl
6	P510550	obverse	4	231520	line	aš-šum ha-za-nu-um-sar	4	203984	cluster	ha-za-nu-um-sar	la			
7	P510550	obverse	7	231523	line	šum-ma ha-za-nu-um-sar la ba-aš-lu-ma					7	203987	cluster	ha
8	P510550	obverse	9	231525	line	ha-za-nu-um-sar i-ka-am x x-al-la-x-ma					9	203989	cluster	ha
9	P510550	reverse	1	231528	line	a-na mi-nim ha-za-nu-um-sar					1	203992	cluster	ha-za-nu-u

KNAW Humanities Cluster

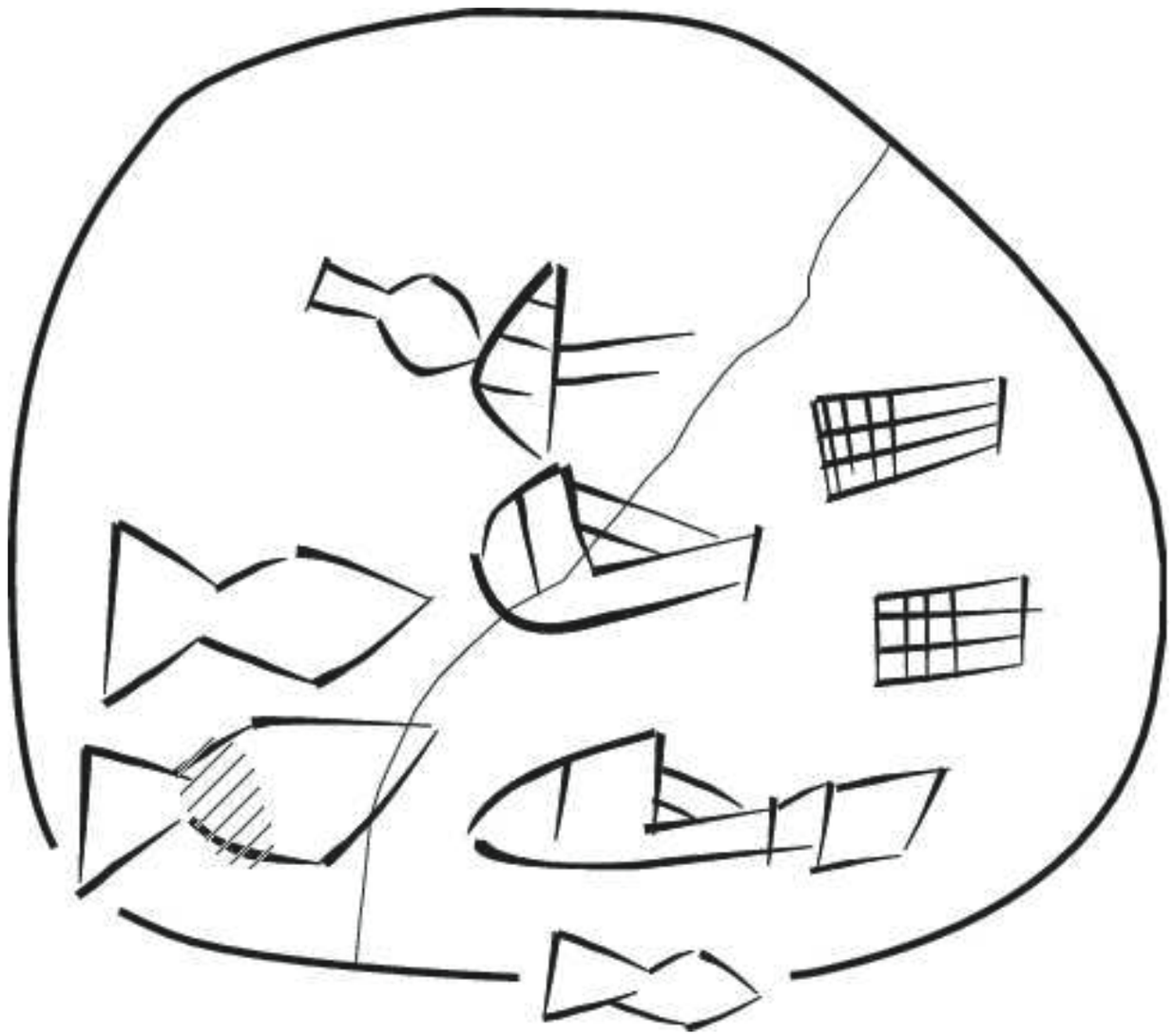
tablets-as-data

```
@node
@converters=Cale Johnson, Dirk Roorda
@description=reading of a sign
@editor=Cale Johnson et al.
@name=NinMed Medical Texts from Nineveh
@project=BabMed
@valueType=str
@writtenBy=Text-Fabric
@dateWritten=2022-02-10T09:59:19Z

14   su
ub
hur
su
ub
22   ša
sa
aš
kur
ni
44   hi
52   ma
mit
55   šar
ra
qi
ina
60   pil
ši
81   ina
85   ta
kar
bu
uš
111  šim
ar
139  ina
141  tur
ar
```

```
@node
@author=Ngan-Tillard et al.
@description=organic traces as color
@valueType=str
@project=Securing Data
@writtenBy=Text-Fabric
@dateWritten=2023-07-19T15:09:18Z

15 red
17 indigo
23-25 green
37 green
143 indigo
red
146 red
```

KNAW Humanities Cluster

P000022



thank you

dirk.roorda@di.huc.knaw.nl

KNAW Humanities Cluster