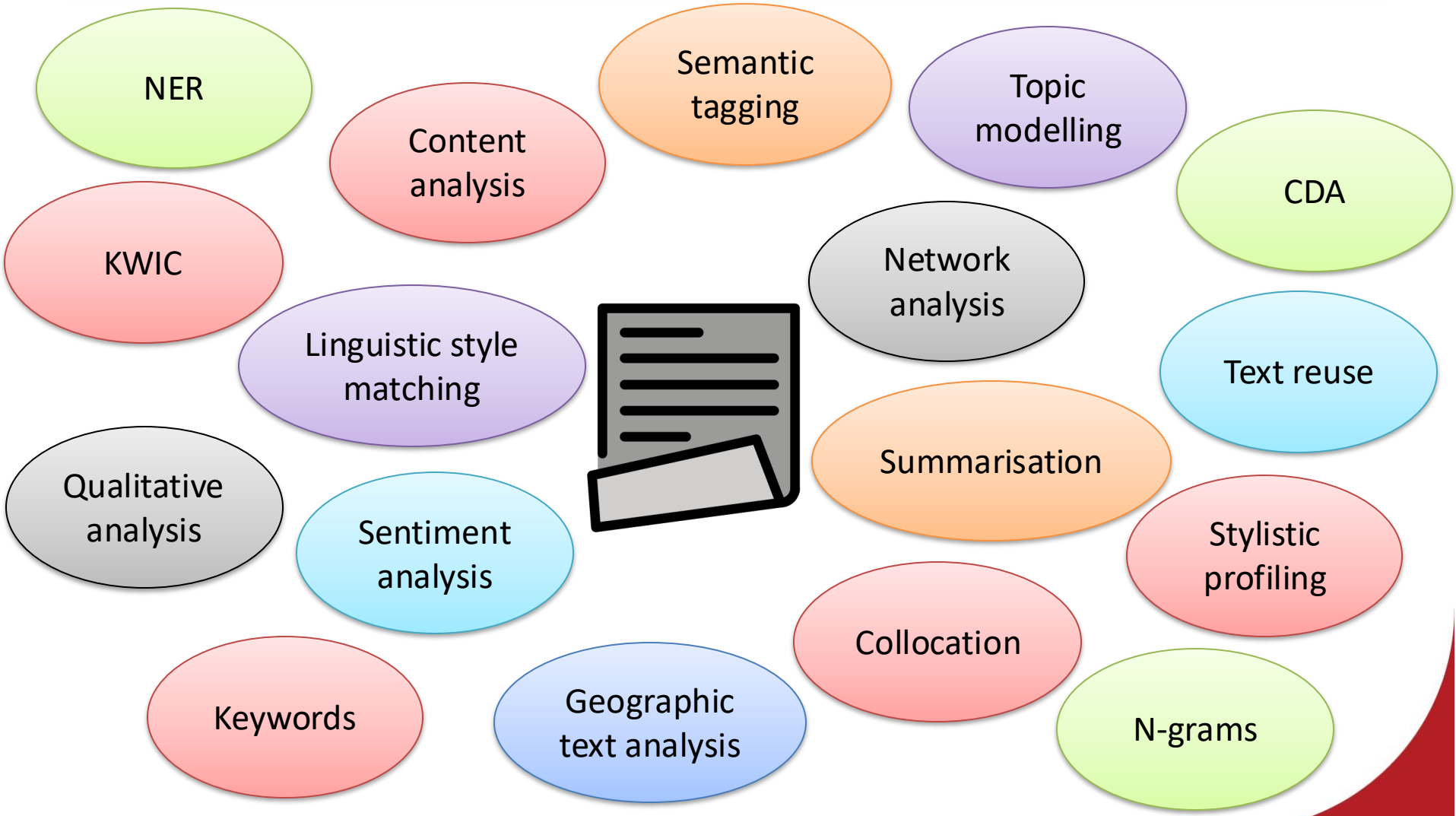
Open access and open source tools for corpus linguistics: Wmatrix version 7 and PyMUSAS

CL2025 Pre-conference workshop: 29th June 2025

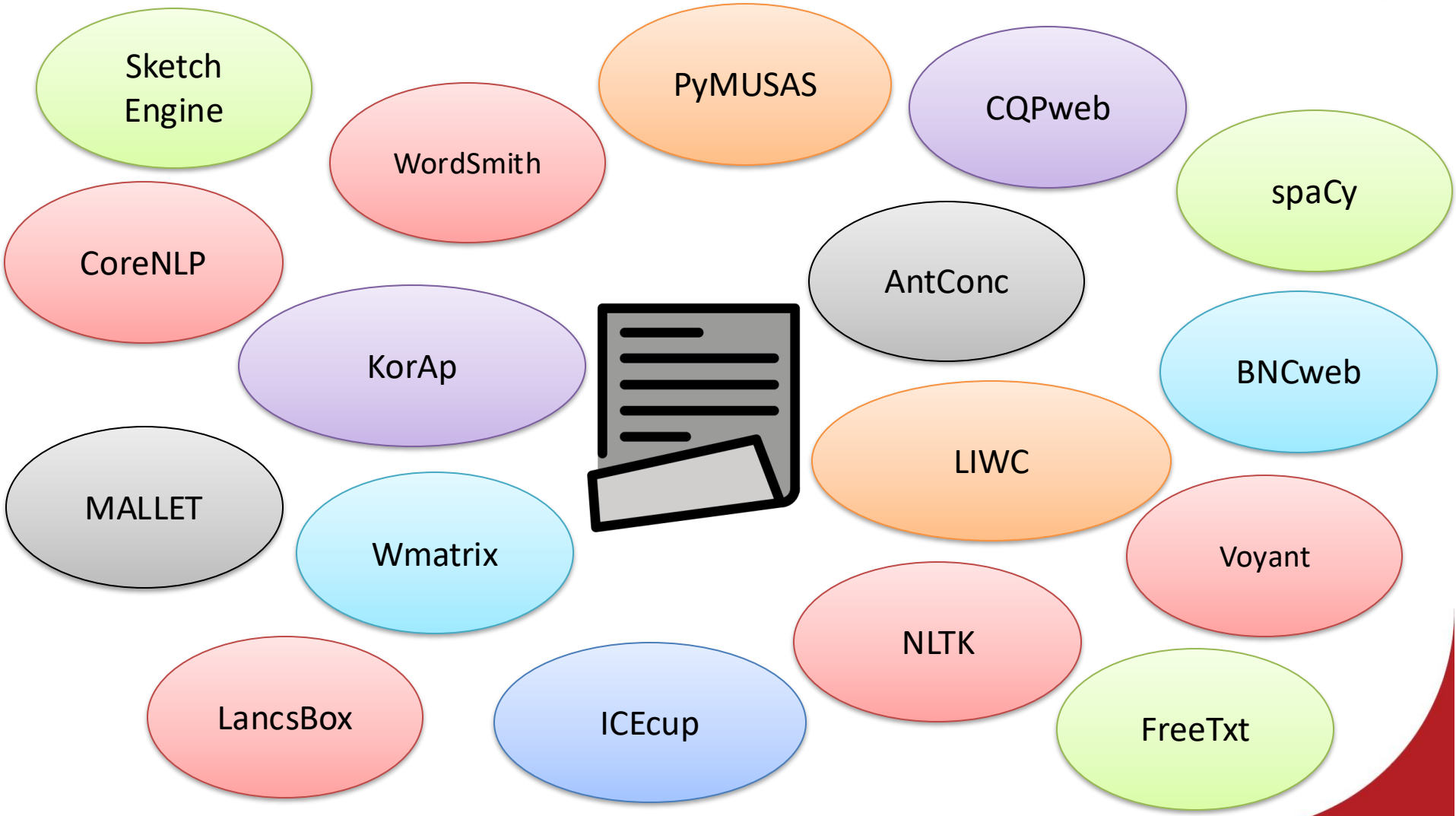
Slides at https://ucrel.lancs.ac.uk/paul/

Prof Paul Rayson, Dr Daisy Lal, Dr John Vidler, Dr Andrew Moore
UCREL research centre
School of Computing and Communications
Lancaster University, UK
perayson.bsky.social, johnvidler.co.uk, ucrelnlp.bsky.social

A myriad of NLP and CL methods ...



A myriad of NLP and CL tools …



Importance of open source and open access tools & resources

Lancaster University

- Vital for reproducibility and replicability of corpus linguistics studies
- Explainability of annotation c.f. generative AI / LLM methods, many of which do not declare their training materials or methods
- Extensibility and unrestricted free access to tools and resources in the global south and low resource contexts

Openness of LLMs.....

Model Name	Availability					Documentation						Access		
	Base Data	Fine Tuning Data	Base Weights	Fine Tuning Weights	Code	Code Docs	Model Architecture	Pre-print	Paper	Model Card	Datasheet	Package	API	License
OLMo	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	~	Y
Deepseek V3	N	N	N	Y	~	~	~	Y	N	~	N	Y	Y	~
Phi	N	N	Y	Y	N	N	Y	Y	N	Y	N	Y	N	Y
Mistral	N	N	Y	Y	~	~	~	~	N	N	N	Y	Y	Y
Gemma	N	N	~	~	~	N	Y	Y	N	Y	N	Y	N	N
Chat-GPT	N	N	N	N	N	N	N	~	N	N	N	N	~	N
	N = No				~ = Partial					Y = Yes				

Based off the European Open-Source AI Index:
https://osai-index.eu/the-index

Recent releases

- All semantic lexicons are now open access (CC-BY-NC-SA-4.0)
  - https://github.com/UCREL/Multilingual-USAS
- Python reimplementation of semantic tagger, PyMUSAS is now open source (Apache 2.0)
  - https://pypi.org/project/pymusas/
- Wmatrix7, with PyMUSAS tagging for 8 languages, is now free & open access for researchers worldwide albeit with filestore limits, and a new indexing system built on SQLite

A workshop of five parts

1. Semantic annotation (tagging)
   - a little bit computational
2. Key semantic tags (key domains)
   - a little bit of statistics
3. Wmatrix and PyMUSAS software
   - Hands on practical
4. Refreshments (15:00–15:30)
5. Current and future developments
   - UCREL-Hex ‘medium’ scale compute cluster
   - 4D Picture application and PyMUSAS
6. Yet more hands on practical …
   - And your chance to provide feedback and influence future plans!

Lexical ambiguity

- Question:
  - How many senses does *spring* have?
- Answer:
  - A: 3
  - B: 4
  - C: 5
  - D: 31

to arrange the escape of (someone) from prison. **17** (*intr*) *Archaic or poetic.* (of daylight or dawn) to begin to appear. ◆ *n* **18** the act or an instance of springing. **19** a leap, jump, or bound. **20a** the quality of resilience; elasticity. **20b** (*as modifier*): *spring steel*. **21** the act or an instance of moving rapidly back from a position of tension. **22a** a natural outflow of ground water, as forming the source of a stream. **22b** (*as modifier*): *spring water*. **23a** a device, such as a coil or strip of steel, that stores potential energy when it is compressed, stretched, or bent and releases it when the restraining force is removed. **23b** (*as modifier*): *a spring mattress*. **24** a structural defect such as a warp or bend. **25a** (*sometimes cap.*) the season of the year between winter and summer, astronomically from the March equinox to the June solstice in the N hemisphere and from the September equinox to the December solstice in the S hemisphere. **25b** (*as modifier*): *spring showers*. Related adj: **vernal**. **26** the earliest or freshest time of something. **27** a source or origin. **28** one of a set of strips of rubber, steel, etc., running down the inside of the handle of a cricket bat, hockey stick, etc. **29** Also called: **spring line**. *Nautical*. a mooring line, usually one of a pair that cross amidships. **30** a flock of teal. **31** *Architect.* another name for **springing**. [Old English *springan;* related to Old Norse *springa,* Old High German *springan,* Sanskrit *sprhayati* he desires, Old Slavonic *pragu* grasshopper] ▸ **'springless** *adj* ▸ **'spring,like** *adj*

spring balance *or esp. U.S.* **spring scale** *n* a device in which an object to be weighed is attached to the end of a helical spring, the extension of which indicates the weight of the object on a calibrated scale.

Definition of *spring* from Collins English Dictionary (21st Century Edition)

Spring (https://dictionary.cambridge.org)

- *spring* was found in the Cambridge Advanced Learner's Dictionary at the entries listed below.
  - spring (MOVE QUICKLY)
  - spring (APPEAR SUDDENLY)
  - spring (SEASON)
  - spring (CURVED METAL)
  - spring (WATER)
  - box spring
  - spring chicken
  - spring-clean
  - spring greens
  - spring onion
  - spring roll
  - spring from sth
  - spring sth on sb
  - be full of the joys of spring
  - spring to life
  - spring to mind
  - a spring in your step

What is Semantic Tagging?

Lancaster University

- Semantic field annotation has applications for conceptual or topic tagging:
  - *Last_T1.1.1 year_T1.1.1* was_A3+ the_Z5 UK_Z2 's_Z5 second_N4 warmest_O4.6+++ *on_A11.2+ record_A11.2+* ,_PUNC *according_Z5 to_Z5* provisional_T1.3- data_X2.2 from_Z5 the_Z5 Met_S3.1 Office_I2.1/H1c ._PUNC This_Z8 *puts_X2.2-* it_Z8 just_A14 *behind_X2.2-* 2022_N1 ,_PUNC which_Z8 recorded_Q1.2 an_Z5 average_A6.2+ temperature_O4.6 of_Z5 only_A14 0.06C_Z99 higher_N3.7++ ._PUNC

- A3+ = being; A6.2 = comparing; A11.2 = importance; A14 = exclusivisers; H1 = architecture, buildings; I2.1 = business; N1 = numbers; N3.7 = measurement; N4 = linear order; O4.6 = temperature; Q1.2 = documents, writing; S3.1 = relationship; T1.1.1 = Time past; T1.3 = time period; X2.2 = knowledge; Z2 = geographical names; Z5 = grammatical bin; Z8 = pronouns etc; Z99 = unmatched

Multiword expressions: plain sailing?

- Phrasal verbs
  - *Stubbed out*
- Noun phrases
  - *Riding boots*
  - *Pony nuts*
- Proper names
  - *United States of America*
- *Named entities*
  - *23rd November 1963*
  - *British Broadcasting Corporation*
- Multiword prepositions
  - *In terms of*
  - *As soon as*
- Idiomatic expressions
  - *Spill the beans*
  - *A pain in the neck*

UCREL Semantic Analysis System (USAS)

- Full text tagging, not just selected words (c.f. Diction, LIWC, RID)
- Tagging the coarse-grained sense in context, not just the word
- Not task specific categories
- Flexible category set with hierarchical structure
- Words and multi-word expressions (MWE) e.g. phrasal verbs (stubbed out), noun phrases (riding boots), proper names (United States of America), true idioms (living the life of Riley)

- https://ucrel.lancs.ac.uk/usas/
- Lexicons available free for academic use:
  - https://github.com/UCREL/Multilingual-USAS

The work of many hands …

- Joint research with
  - Geoffrey Leech
  - Roger Garside
  - Jenny Thomas
  - Andrew Wilson
  - Dawn Archer
  - Scott Piao
  - Sheryl Prentice
  - Andrew Moore
  - Daisy Lal
  - Ignatius Ezeani










Semantic fields

- AKA concepts, semantic domains
- ‘groups together word senses that are related by virtue of their being connected at some level of generality with the same mental concept’
- Not only synonymy and antonymy but also hypernymy and hyponymy
- E.g. EDUCATION: academic, coaching, coursework, deputy head, exams, PhD, playschool, revision notes, studious, swot, viva

A General and abstract terms	**B** The body and the individual	**C** Arts and crafts	**E** Emotion
F Food and farming	**G** Government and public	**H** Architecture, housing and the home	**I** Money and commerce in industry
K Entertainment, sports and games	**L** Life and living things	**M** Movement, location, travel and transport	**N** Numbers and measurement
O Substances, materials, objects and equipment	**P** Education	**Q** Language and communication	**S** Social actions, states and processes
T Time	**W** World and environment	**X** Psychological actions, states and processes	**Y** Science and technology
Z Names and grammar			

Lexical resources for English

- Lexicon of 56,316 items
  - presentation NN1 Q2.2 A8 S1.1.1 K4
- MWE list of 18,971 items
  - travel_NN1 card*_NN* M3/Q1.2
- A small wildcard lexicon
  - *kg NNU N3.5
- Unknown words using WordNet synonym lookup

English Disambiguation methods (1)

- 1. POS tag
  - *spring* noun [season sense] [coil sense]
  - *spring* verb [jump sense]
- 2. General likelihood ranking for single-word and MWE tags
  - *green* referring to [colour] is generally more frequent than *green* meaning [inexperienced]
- 3. Overlapping MWE resolution
  - Heuristics applied: semantic MWEs override single word tagging, length and span of MWE also significant

English Disambiguation methods (2)

- 4. Domain of discourse
  - adjective *battered*
    - [Violence] (e.g. battered person)
    - [Judgement of Appearance] (e.g. battered car)
    - [Food] (e.g. battered cod)
- 5. Text-based disambiguation
  - one sense per text
- 6. Template rules
  - *Auxiliary verbs (be/do/have)*
  - *account* of NP [narrative]
  - balance of xxx *account* [financial]

Evaluation (English data)

- Hand tagged test corpus of 124,839 words
- Error rate of 8.95%
- Ambiguity ratio 47.73%
- Reduced to 17.06% by disambiguation
- Not all ambiguity is resolved, but 1st choice tag selection gives 91% accuracy.

KEY SEMANTIC DOMAINS AND FURTHER APPLICATIONS

Wmatrix

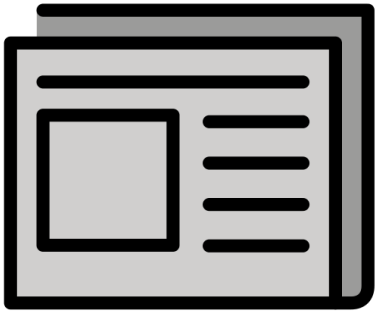
Lancaster University

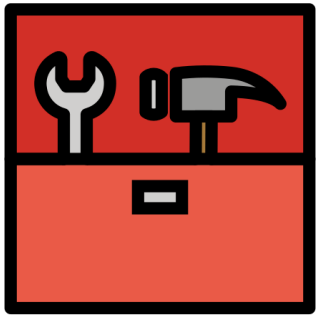
Keywords

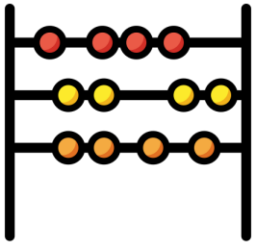
		LibDem manifesto		Labour manifesto			
	Word	Frequency	Rel. freq.	Frequency	Rel. freq.	O/U-use	LL
1	liberal	47	0.23	0	0.00	+	81.41
2	would	70	0.34	10	0.04	+	71.89
3	democrats	40	0.20	0	0.00	+	69.29
4	our	76	0.37	272	0.97	-	63.22
5	labour	33	0.16	152	0.54	-	49.56
6	is	119	0.58	330	1.17	-	47.04
7	which	92	0.45	37	0.13	+	45.13
8	now	8	0.04	76	0.27	-	43.97
9	1997	4	0.02	54	0.19	-	36.76
10	green	26	0.13	2	0.01	+	32.81
11	environmental	47	0.23	14	0.05	+	30.98
12	establish	34	0.17	7	0.02	+	29.06
13	since	2	0.01	38	0.14	-	29.06
14	ten-year	0	0.00	25	0.09	-	27.29
15	also	88	0.43	50	0.18	+	26.30
16	Governments	15	0.07	0	0.00	+	25.98
17	britains	15	0.07	0	0.00	+	25.98
18	long_term	15	0.07	0	0.00	+	25.98
19	new	57	0.28	165	0.59	-	25.91
20	's	29	0.14	106	0.38	-	25.46

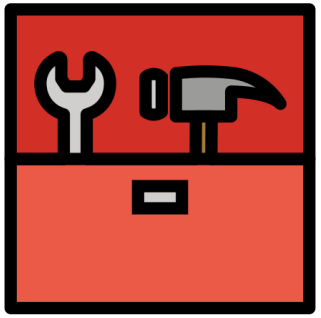
Text

Text or reference corpus









the	351
of	243
a	221
and	153
to	139
in	134
is	123
be	83
for	81
phrase	69
that	67
which	66
are	64
by	60
words	57
x	53
as	50
not	48
or	46
phrases	44

Word frequency list

the	351
of	243
a	221
and	153
to	139
in	134
is	123
be	83
for	81
phrase	69
that	67
which	66
are	64
by	60
words	57
x	53
as	50
not	48
or	46
phrases	44

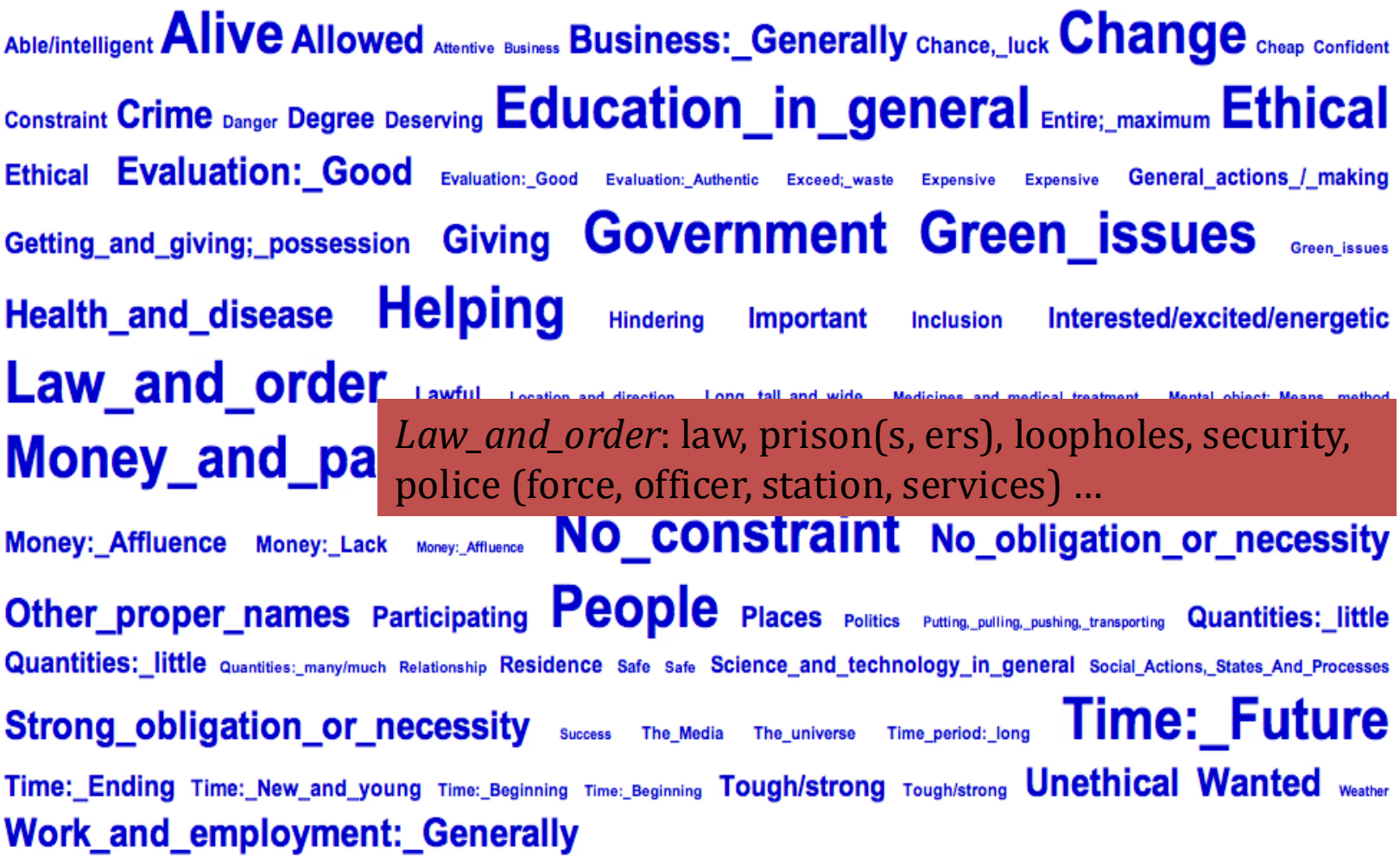
Word frequency list

Significance and effect size

- Log-likelihood (LL) Wizard online at:
  - https://ucrel.lancs.ac.uk/llwizard.html
- Spreadsheet and code also available for download
  - https://github.com/UCREL/SigEff
- Very important to consider dispersion and effect size measures (depending on your corpus) – included in Wmatrix frequency lists and keyness measures
  - See the work of Hardie, Gabrielatos, Brezina and others
  - Rayson and Potts (2021)

Figure 1: keywords in LibDem 2010 manifesto

Figure 2: key domains (semantic fields) in LibDem 2010 manifesto



Law_and_order: law, prison(s, ers), loopholes, security, police (force, officer, station, services) …

Applications of semantic analysis

100+ papers listed at https://ucrel.lancs.ac.uk/wmatrix/

- Analysis of market research interview transcripts
- Intelligent dictionaries
- Assistance for human translators
- Software Engineering domain understanding
- Language profiling for online child protection
- Actionability
- Corpus stylistics
- Prediction of real-world events from social media
- Metaphor and end-of-life care
- Pattern analysis of the language of psychopaths
- Political discourse analysis
- Describing the language of extremism and counter-extremism
- UK General Election Manifestos (Rayson 2008)

Metaphor, cancer and end of life care (MELC)

Lancaster University

- Analysis of metaphorical language used to talk about cancer, dying and death: people 'fight' their cancer, 'win' or 'lose' their 'battle' against it, hope for a positive end to their cancer 'journey', and so on.
- 1.5M word corpus of interviews and online forum posts from patients, carers and healthcare professionals
- Methods: Manual analysis (MIP) and Wmatrix (Semantic analysis & concordancing)
- http://wp.lancs.ac.uk/melc/

G3 Warfare (e.g. *fight* as a verb, *battle*)
A1.1.1 General actions, making (e.g. *blast*, *confront*)
A1.1.2 Damaging and destroying (e.g. *destroy*, *shatter*)
E3– Violent/angry (e.g. *hit*, *attack*)
S8+ Helping (e.g. *defend*, *protect*)
S8– Hindering (e.g. *fight* as a noun)
X8+ Trying hard (e.g. *struggle*)

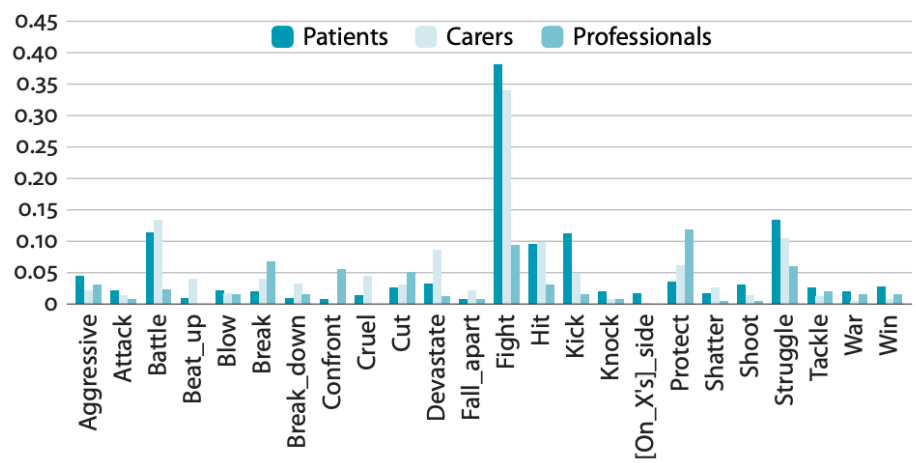


Figure 3. Relative use of most frequent Violence metaphors by each stakeholder group (per 1,000 tokens): Online forum posts

Qualitative survey analysis: FreeTxt/TestunRhydd project (2022-3)

- Surveys are widely used in many areas of professional practice, e.g. staff development, professional training, product design, testing as well as for many types of hotel, movie and product reviews
- Very little support for bilingual free-text survey and questionnaire data analysis in English and Welsh
- Follow on funding impact project building on CorCenCC project (National Corpus of Contemporary Welsh), we will develop an open access user friendly online interface
- Partners: National Trust Wales, Cadw and National Museum Wales
- https://ucrel.lancs.ac.uk/freetxt/

Lancaster University

1990-6

English semantic tagger
(C version)

2002-5

Finnish semantic tagger
(Java version)

2005-7

Russian semantic tagger

2013-

onwards: Extension to 10+ languages

2014-5

Historical Thesaurus Semantic Tagger (utilising HTE after 40 years of development at Glasgow University and 150+ years for the OED

2016–2021

Specialisation for genetics discourse (extension with a pre-existing Gene Ontology)

2021-2

Python code released open source (pymusas)

- https://ucrel.github.io/pymusas/
- https://pypi.org/project/pymusas/

2025

PyMUSAS integrated into Wmatrix7 (Chinese, Dutch, Finnish, French, Italian, Portuguese, Spanish, Welsh)

Recipe for creating a tagger in a new language

1. re-evaluate USAS semantic tagset for new language context
2. find freely available (open source if possible) POS tagger & lemmatiser
3. integrate these into USAS Multilingual software framework (PyMUSAS)
   a. consider whether other new components are needed e.g. tokeniser or compound tool
4. develop single-word semantic lexicon and MWE dictionary
   a. bilingual dictionary
   b. parallel aligned corpus (Moses / Giza)
   c. machine translation / translation memory
   d. crowdsourcing by non-experts
   e. named entity recognition and gazetteers
   f. vector-based approaches
   g. multi-task & deep learning
   h. manual checking and editing by experts
5. extend disambiguation routines
6. release lexicons with CC-BY-NC-SA licence
7. release software as REST API and/or open-source licence

PyMUSAS

https://pypi.org/project/pymusas/

- Open source – Apache License Version 2.0
- Open resources – Creative Commons licence version 4
- Rule based tagger
- Identify and tag Multi Word Expressions (MWE)
- Supports multiple languages through downloadable spaCy pipelines
- Supports Indonesian and Welsh via other POS taggers (TreeTagger for Indonesian and CyTag for Welsh)

Language (BCP 47 language code)	MWE Support	Size
Mandarin Chinese (cmn)	✔	1.28MB
Welsh (cy)	✔	1.09MB
Spanish, Castilian (es)	✔	0.20MB
French (fr)	×	0.08MB
Indonesian (id)	×	0.24MB
Italian (it)	✔	0.50MB
Dutch, Flemish (nl)	×	0.15MB
Portuguese (pt)	✔	0.27MB

PyMUSAS – Language Support

Each language that we support has a guide on how to semantically tag text for that language:

https://ucrel.github.io/pymusas/usage/how_to/tag_text

Tag Text

Chinese
Dutch
French
Italian
Portuguese
Spanish
Welsh
Indonesian

In this guide we are going to show you how to tag text using the PyMUSAS RuleBasedTagger so that you can extract token level USAS semantic tags from the tagged text. The guide is broken down into different languages, for each guide we are going to:

1. Download the relevant pre-configured PyMUSAS `RuleBasedTagger` spaCy component for the language.
2. Download and use a Natural Language Processing (NLP) pipeline that will tokenise, lemmatise, and Part Of Speech (POS) tag. In most cases this will be a spaCy pipeline. **Note** that the PyMUSAS `RuleBasedTagger` only requires at minimum the data to be tokenised but having the lemma and POS tag will improve the accuracy of the tagging of the text.
3. Run the PyMUSAS `RuleBasedTagger`.
4. Extract token level linguistic information from the tagged text, which will include USAS semantic tags.
5. For Chinese, Italian, Portuguese, Spanish, and Welsh taggers which support Multi Word Expression (MWE) identification and tagging we will show how to extract this information from the tagged text as well.

Chinese

Expand

Dutch

Expand

French

Expand

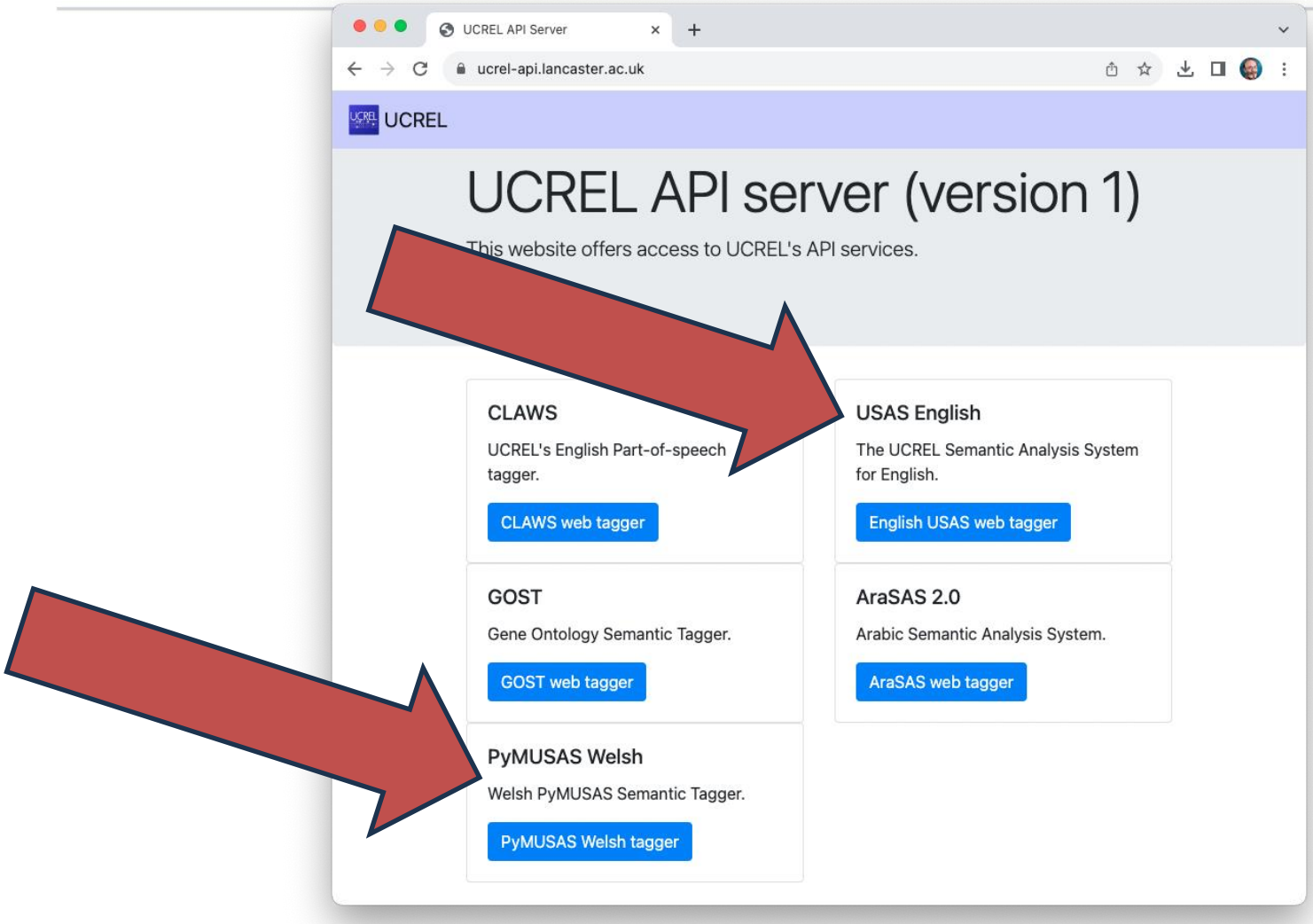
Try this Python Notebook during the hands on session:
https://github.com/UCREL/pymusas_notebook

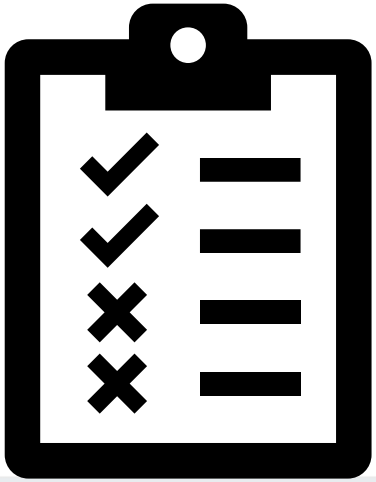
Recent developments in new languages

- Used the IgboAPI dataset (33 distinct Igbo dialects, 5,095 Igbo words with 17,979 unique dialectal word variations, complemented by 27,816 example parallel sentences) to bootstrap a lexicon for the Igbo semantic tagger
  - https://aclanthology.org/2024.lrec-main.1384/
- Creation of high quality linguistic resources (MWE lexicon) via LLMs to retrieve MWE definitions facilitating accurate translation from English to Danish lexicons, coverage evaluation and manual annotation for metaphor analysis in 4D Picture project
  - Puts et al. (2025) Pushing the boundaries: creating a Danish semantic tagger for metaphor analysis of cancer narratives. Corpus Linguistics 2025, Birmingham, UK.

https://ucrel-api.lancaster.ac.uk/

You can also test USAS without a login for Wmatrix





WMATRIX VERSION 7

Key points

- Web-based (c.f. BNCweb, CQPweb, SketchEngine)
- Dedicated server, Secure HTTPS access
- You can load your own data (Multilingual in v7)
- Incorporates main methods in corpus linguistics toolbox
  - frequency lists, concordances, key words, collocations, n-grams
- Adds two levels of linguistic annotation (NLP methods)
  - POS tagging, Semantic field tagging
- Novelty
  - key domain analysis, semantic collocations

Hands on practical

- 2005 UK general election
  - Liberal Democrat party manifesto
  - Labour party manifesto
- 2010 UK general election
  - manifestos for all three main parties
- 2015, 2017, 2019 and 2024 UK general elections
  - manifestos for seven parties
- Aims:
  - To help you understand the basic Wmatrix features and key domains method
  - To give you some awareness of the semantic tagset

Version 7 compared to version 5

	Wmatrix5	Wmatrix7
Indexing system	Bespoke from 1990s	SQLite
Folders / Corpus	Single file, up to 1M words	Multiple files (zip), tested up to 30M words
Concordances	Corpus order	Various sort options
N-grams and collocations	NSP and Java code	SQLite
Language	USAS English, Spanish beta	PyMUSAS for Chinese, Dutch, Finnish, French, Italian, Portuguese, Spanish, and Welsh
MWEs	Tagged, displayed in frequency lists	Tagged but not yet displayed in frequency lists
Optional features	Domain and My Tag Wizard, Metaphor features, folder sharing	

Open two web-browser windows or tabs

- All URLs linked from Wmatrix home page:
  - https://ucrel.lancs.ac.uk/wmatrix/

1. Wmatrix tutorials
   - https://ucrel.lancs.ac.uk/wmatrix/tutorial7/

1. Wmatrix tool:
   - https://ucrel-wmatrix7.lancaster.ac.uk/
   - Apply for login now if you haven't already got one

Your tasks!!

- https://ucrel.lancs.ac.uk/wmatrix/tutorial7/
- On your own or in small groups ...
  - **Do** tutorials A and B (you can either upload the manifesto documents yourself into Wmatrix, or use the ones I made earlier in the corpus library)
  - **Do** tutorial C (key words, key domains and concordances)

  - For the keen ones amongst you, move on to the other tutorials
  - You can use your own data if you wish
  - Ask questions any time!

Lancaster University

Beyond Wmatrix:

Parallel and Cluster Systems

Or “This is taking too long…” and “Why is my computer really hot?”

Working with truly vast amounts of data…

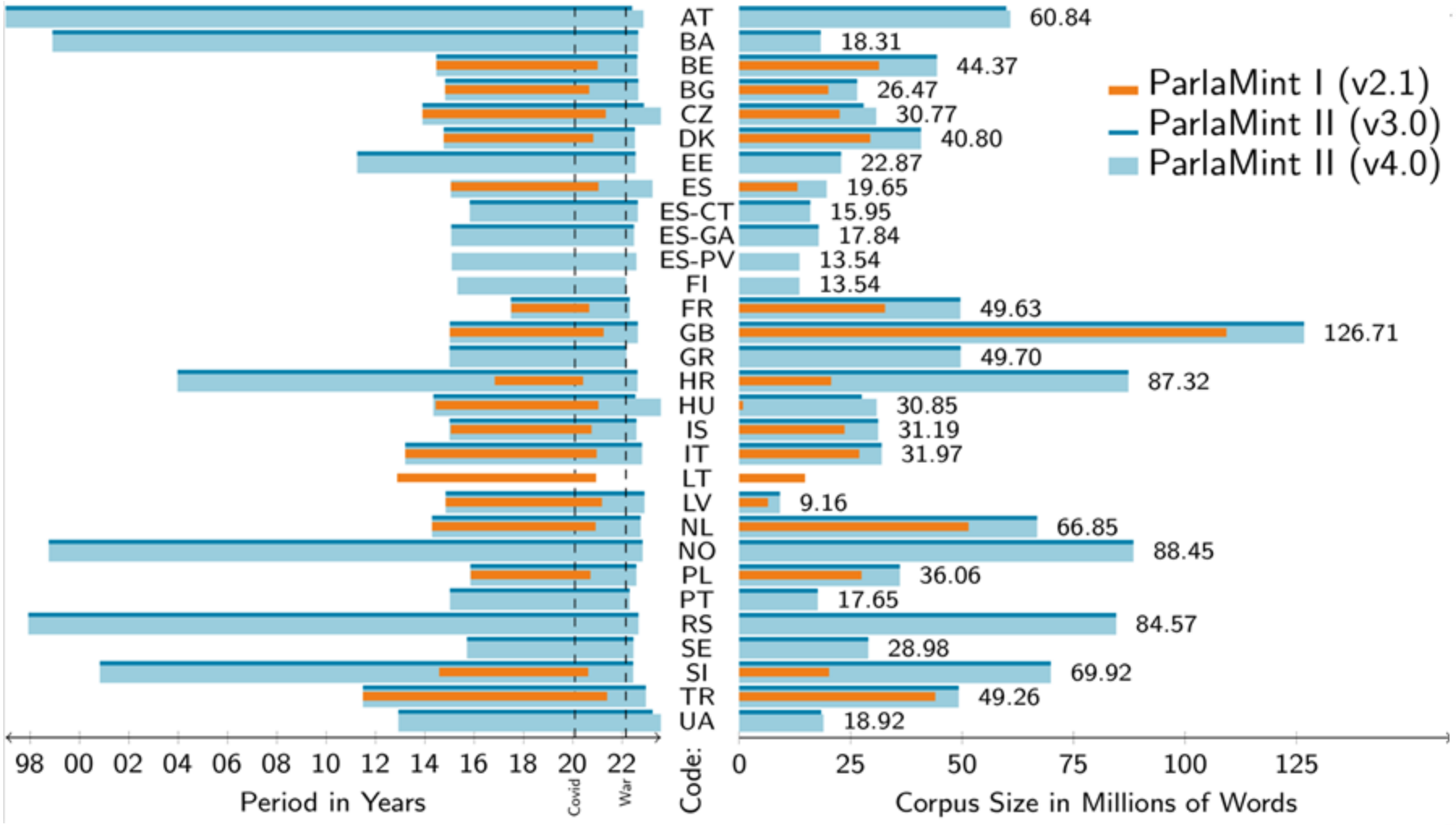
If any of these apply, then you may need to start seriously thinking about scaling your software:

- Does everything take too long?
  - Are you ‘at risk’ for process instability?
- Are you running out of RAM?
  - “Isn’t 128GiB enough?”
- Are you resource bound?
  - “But I need a GPU for every process…”

An Example: ParlaMint

- CLARIN flagship project (https://www.clarin.eu/parlamint)
- Creation of comparable, uniformly annotated, CC-BY corpora of parliamentary debates across Europe
- Two project stages (2020-21, 2022-23)
- V4.0 released October 2023: http://hdl.handle.net/11356/1859
- English MT annotated version released November 2023: http://hdl.handle.net/11356/1864
- 7.8M utterances; 1.2B words; 29 languages

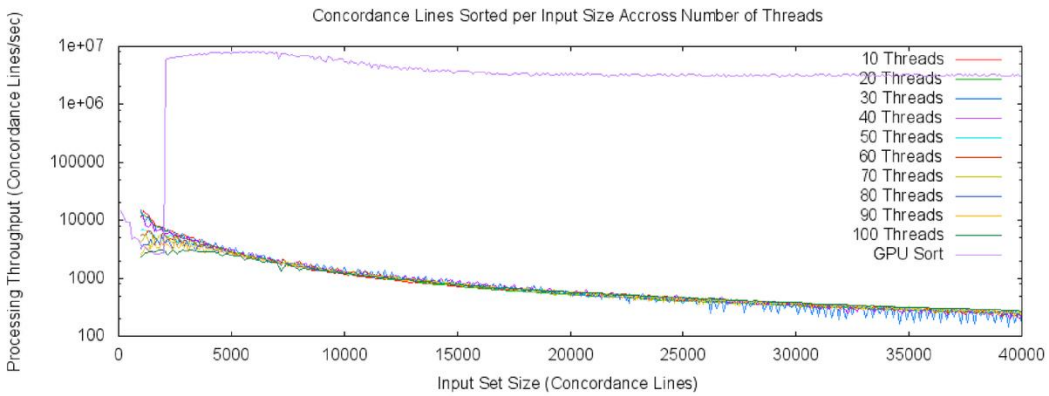
ParlaMint: Endlessly Growing Data



NB. These figures are old now!

Data Dependencies

- If dataset 'A' never interacts with dataset 'B' we can process them independently
  - We can often cheat this too (kinda!)
- In general, loops over large collections are good candidates for handling in parallel
  - This is formalised as the 'map' semantics from map/reduce



Dealing With Big Data Outside Of The Cloud: GPU Accelerated Sort – Vidler et.al

Data Dependencies

The ParlaMint dataset is *highly* independent

- Three levels of iteration:
  1. By Language
  2. By Year
  3. By File

But… the corpus is also *very large™*, so rather than attempting to parallelise everywhere, instead focus on "easy wins"

- For reference, let us take a quick look at the folder structure

ParlaMint Corpus Folder Structure

One folder for every language

```
ParlaMint-AT-en.conllu
 ├── 1996
 |   ├── ParlaMint-AT-en_1996-01-15-020-XX-NRSITZ-00001.conllu
 |   ├── ParlaMint-AT-en_1996-01-15-020-XX-NRSITZ-00002.conllu
 |   ├── etc...
 ├── 1997
 |   ├── ParlaMint-AT-en_1997-01-14-020-XX-NRSITZ-00055.conllu
 |   ├── ParlaMint-AT-en_1997-01-14-020-XX-NRSITZ-00056.conllu
 |   ├── etc...
```

One folder for every year collection

One file for every data set

ParlaMint Corpus Folder Structure

Easy Win #1 - We can run more than one *language* at once

One folder for every language

```
ParlaMint-AT-en.conllu
 ├── 1996
 |   ├── ParlaMint-AT-en_1996-01-15-020-XX-NRSITZ-00001.conllu
 |   ├── ParlaMint-AT-en_1996-01-15-020-XX-NRSITZ-00002.conllu
 |   ├── etc…
 ├── 1997
 |   ├── ParlaMint-AT-en_1997-01-14-020-XX-NRSITZ-00055.conllu
 |   ├── ParlaMint-AT-en_1997-01-14-020-XX-NRSITZ-00056.conllu
 |   ├── etc…
```

One folder for every year collection

One file for every data set

Easy Win #2 – We can run more than one *year* at once

Easy Win #3 – We can run more than one *file* at once

ParlaMint Corpus Folder Structure

Easy Win #1 - We can run more than one *language* at once

One folder for every language

```
ParlaMint-AT-en.conllu
 ├── 1996
 |   ├── ParlaMint-AT-en_1996-01-15-020-XX-NRSITZ-00001.conllu
 |   ├── ParlaMint-AT-en_1996-01-15-020-XX-NRSITZ-00002.conllu
 |   ├── etc...
 ├── 1997
 |   ├── ParlaMint-AT-en_1997-01-14-020-XX-NRSITZ-00055.conllu
 |   ├── ParlaMint-AT-en_1997-01-14-020-XX-NRSITZ-00056.conllu
 |   ├── etc...
```

One folder for every year collection

Easy Win #2 – We can run more than one *year* at once

One file for every data set

~~Easy Win #3 – We can run more than one *file* at once~~

ParlaMint Corpus Folder Structure

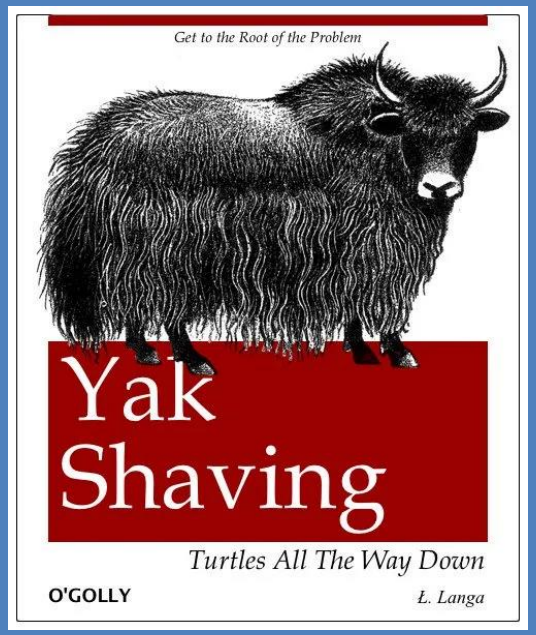
Easy Win #1 - We can
one

One folder for
every year collecti

Easy Win #2 – W
can run more t
one *year* at onc

Why not #3?

We may run the risk of over producing very short-lived jobs; which *might* not be a problem but could easily result in the majority of our time spent running being taken up by starting/stopping the jobs rather than actually running them.



Tagging Parlamint: How it began

Lancaster University

~ — ubuntu@instance-20230908-2241: ~ — ssh oci-arm

```
nmon-16n                    Hostname=instance-2023Refresh= 2secs ---01:03.30
 CPU Utilisation
-------------------------+-------------------------------------------------+
CPU User%  Sys% Wait%  Idle|0          |25         |50         |75       100|
  1   0.5   0.0   0.0  99.5|                                                 >
  2   0.0   0.0   0.0 100.0|                                                 >
  3   0.5   0.0   0.0  99.5|                                                 >
  4   0.5   0.0   0.0  99.5|                                                 >
  5   0.5   0.0   0.0  99.5|                                                 >
  6   0.5   0.0   0.0  99.5|                                                 >
  7   0.0   0.0   0.0 100.0|                                                 >
  8   0.0   0.0   0.0 100.0|                                                 >
  9   0.5   0.0   0.0  99.5|                                                 >
 10   0.0   0.0   0.0 100.0|                                                 >
 11   0.5   0.0   0.0  99.5|                                                 >
 12   0.5   0.0   0.0  99.5|                                                 >
 13   0.5   0.0   0.0  99.5|                                                 >
 14   0.0   0.0   0.0 100.0|                                                 >
 15   0.5   0.0   0.0  99.5|                                                 >
 16   0.0   0.0   0.0 100.0|                                                 >
 17   0.5   0.0   0.0  99.5|                                                 >
 18   0.0   0.0   0.0 100.0|                                                 >
 19   0.0   0.0   0.0 100.0|                                                 >
 20   0.0   0.0   0.0 100.0|                                                 >
 21   0.5   0.0   0.0  99.5|                                                 >
 22   0.5   0.0   0.0  99.5|                                                 >
 23   0.5   0.0   0.0  99.5|                                                 >
 24   0.5   0.0   0.0  99.5|                                                 >
 25   0.5   0.0   0.0  99.5|                                                 >
 26   0.5   0.0   0.0  99.5|                                                 >
 27   0.0   0.0   0.0 100.0|                                                 >
 28   0.0   0.0   0.0 100.0|                                                 >
 29   0.5   0.0   0.0  99.5|                                                 >
 30   0.5   0.0   0.0  99.5|                                                 >
 31   0.5   0.0   0.0  99.5|                                                 >
 32   0.0   0.0   0.0 100.0|                                                 >
 33   0.0   0.0   0.0 100.0|                                                 >
 34   0.5   0.0   0.0  99.5|                                                 >
 35   0.0   0.0   0.0 100.0|                                                 >
 36   0.5   0.0   0.0  99.5|                                                 >
 37   0.0   0.0   0.0 100.0|                                                 >
 38   0.0   0.0   0.0 100.0|                                                 >
 39   0.0   0.0   0.0 100.0|                                                 >
 40   0.5   0.0   0.0  99.5|                                                 >
 41   0.5   0.0   0.0  99.5|                                                 >
 42   0.0   0.0   0.0 100.0|                                                 >
 43   0.5   0.0   0.0  99.5|                                                 >
 44   0.5   0.0   0.0  99.5|                                                 >
 45   0.0   0.0   0.0 100.0|                                                 >
 46   0.0   0.0   0.0 100.0|                                                 >
 47   0.5   0.0   0.0  99.5|                                                 >
 48   0.5   0.0   0.0  99.5|                                                 >
 49   0.5   0.0   0.0  99.5|                                                 >
 50   0.0   0.0   0.0 100.0|                                                 >
 51   0.0   0.0   0.0 100.0|                                                 >
 52   0.0   0.0   0.0 100.0|                                                 >
 53   0.0   0.0   0.0 100.0|                                                 >
 54   0.0   0.0   0.0 100.0|                                                 >
 55   0.5   0.0   0.0  99.5|                                                 >
 56   0.0   0.0   0.0 100.0|                                                 >
 57   0.5   0.0   0.0  99.5|                                                 >
 58   0.0   0.0   0.0 100.0|                                                 >
 59   0.5   0.0   0.0  99.5|                                                 >
 60   0.5   0.0   0.0  99.5|                                                 >
 61   0.0   0.0   0.0 100.0|                                                 >
 62   0.5   0.0   0.0  99.5|                                                 >
 63   0.5   0.5   0.0  99.0|                                                 >
          Warning: Some Statistics may not shown
```

```
ubuntu@instance-20230908-2241:~$ ls ParlaMint-AT-en.conllu
1996  1998  2000  2002  2004  2006  2008  2010  2012  2014  2016  2018  2020  2022
1997  1999  2001  2003  2005  2007  2009  2011  2013  2015  2017  2019  2021
ubuntu@instance-20230908-2241:~$ ls ParlaMint-AT-en.conllu | wc -l
27
ubuntu@instance-20230908-2241:~$ /opt/pymusas-conllu-parlamint-main/pymusas_parlamint_wrapper.sh /home/ubuntu/ParlaMint-AT-en.conllu
```

```
  2 bash
CONTAINER ID   NAME      CPU %     MEM USAGE / LIMIT   MEM %     NET I/O   BLOCK I/O   PIDS
```

Tagging Parlamint: Afterwards

Lancaster University

~ — ubuntu@instance-20230908-2241: ~ — ssh oci-arm

```
nmon-16n                    Hostname=instance-2023Refresh= 2secs ---01:14.09
 CPU Utilisation
-------------------------+-------------------------------------------------+
CPU User%  Sys% Wait%  Idle|0          |25         |50         |75       100|
  1   0.0   0.0   0.0 100.0|                                                 >
  2   0.0   0.0   0.0 100.0|       >                                         |
  3   0.0   0.0   0.0 100.0|                                   >             |
  4   0.0   0.0   0.0 100.0|>                                                |
  5   0.0   0.0   0.0 100.0|>                                                |
  6   0.0   0.0   0.0 100.0|    >                                            |
  7   0.0   0.0   0.0 100.0|>                                                |
  8   0.0   0.0   0.0 100.0|                                                 >
  9   0.0   0.0   0.0 100.0|                     >                           |
 10   0.0   0.0   0.0 100.0|                                                 >
 11   0.0   0.0   0.0 100.0|  >                                              |
 12   0.0   0.0   0.0 100.0|  >                                              |
 13   0.0   0.0   0.0 100.0|                 >                               |
 14   0.0   0.0   0.0 100.0|>                                                |
 15   0.0   0.0   0.0 100.0| >                                              |
 16   0.0   0.0   0.0 100.0|                                                >|
 17   0.0   0.0   0.0 100.0|                                                 >
 18   0.0   0.0   0.0 100.0|>                                                |
 19   0.0   0.0   0.0 100.0| >                                              |
 20   0.0   0.0   0.0 100.0|    >                                           |
 21   0.0   0.0   0.0 100.0| >                                              |
 22   0.0   0.0   0.0 100.0|      >                                         |
 23   0.0   0.0   0.0 100.0|   >                                            |
 24   0.0   0.0   0.0 100.0|                                                 >
 25   0.0   0.0   0.5  99.5|                   >                             |
 26   0.0   0.0   0.0 100.0|>                                                |
 27   0.0   0.0   0.0 100.0|                >                                |
 28   0.0   0.0   0.0 100.0|>                                                |
 29   0.0   0.0   0.0 100.0|    >                                           |
 30   0.0   0.0   0.0 100.0|                  >                              |
 31   0.0   0.0   0.0 100.0|>                                                |
 32   0.0   0.0   0.0 100.0|    >                                           |
 33   0.0   0.0   0.0 100.0|                                 >               |
 34   0.0   0.0   0.0 100.0|      >                                         |
 35   0.0   0.0   0.0 100.0|      >                                         |
 36   0.0   0.0   0.0 100.0|                             >                   |
 37   0.0   0.0   0.0 100.0|      >                                         |
 38   0.0   0.0   0.0 100.0| >                                              |
 39   0.0   0.0   0.0 100.0|                   >                             |
 40   0.0   0.0   0.0 100.0| >                                              |
 41   0.0   0.0   0.0 100.0| >                                              |
 42   0.0   0.0   0.0 100.0|           >                                    |
 43   0.0   0.0   0.0 100.0|>                                                |
 44   0.0   0.0   0.0 100.0|  >                                             |
 45   0.0   0.5   0.0  99.5|                  >                              |
 46   0.0   0.0   0.0 100.0|                 >                               |
 47   0.0   0.0   0.0 100.0|                >                                |
 48   0.0   0.0   0.0 100.0|   >                                            |
 49   0.0   0.0   0.0 100.0| >                                              |
 50   0.0   0.0   0.0 100.0|        >                                       |
 51   0.0   0.0   0.0 100.0| >                                              |
 52   0.0   0.0   0.0 100.0|        >                                       |
 53   0.0   0.0   0.0 100.0|      >                                         |
 54   0.0   0.0   0.0 100.0|  >                                             |
 55   0.0   0.0   0.0 100.0|                                                 >
 56   0.0   0.0   0.0 100.0|                     >                           |
 57   0.0   0.0   0.0 100.0|   >                                            |
 58   0.0   0.0   0.0 100.0|>                                                |
 59   0.0   0.0   0.0 100.0|>                                                |
 60   0.0   0.0   0.0 100.0|    >                                           |
 61   0.0   0.0   0.0 100.0|>                                                |
 62   0.0   0.0   0.0 100.0|      >                                         |
 63   0.0   0.0   0.0 100.0|      >                                         |
-----------Warning: Some Statistics may not shown-----------------------------
```

```
ubuntu@instance-20230908-2241:~$ /opt/pymusas-conllu-parlamint-main/pymusas_parlamint_wrapper.sh /home/ubuntu/ParlaMint-AT-en.conllu
```

```
  1 bash
CONTAINER ID   NAME      CPU %     MEM USAGE / LIMIT   MEM %     NET I/O   BLOCK I/O   PIDS
```

ParlaMint: A Summary

Shortened a 18-day runtime to a **7-hour** one!

- For all languages!

What tools can we use to achieve this in notebooks?

- Serial Operations:
  - https://tqdm.github.io/

- Parallel Operations:
  - https://pypi.org/project/p-tqdm/

(This may not be the best solution out there, but definitely one of the more simple and effective!)

Running tasks with Slurm!

https://slurm.schedmd.com/

Slurm is a job batch and queuing system on many supercomputer clusters:

- BEDE: https://n8cir.org.uk/bede/
- HEC: https://lancaster-hec.readthedocs.io
- Hex: https://www.lancaster.ac.uk/scc/research/research-facilities/hex/

This is my one ☺

```
vidlerj@login:~$ squeue
          JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
        2126_53  a2000-6h slurmRun  hylandr  R    1:02:59      1 hex-p3-g1
        2126_52  a5000-6h slurmRun  hylandr  R    1:41:36      1 hex-vm-002
        2126_50  a5000-6h slurmRun  hylandr  R    3:32:30      1 hex-vm-001
   2126_[54-75%3] a5000-6h, slurmRun  hylandr PD       0:00      1 (JobArrayTaskLimit)
```

One of our users using 3 GPUs at once!

Parallel Processing Corpus Data: Tips!

- Think about which parts of your program(s) are accessing which data
  - Can parts be split apart to run in parallel without 'side effects?'
- Avoid 'state' and especially 'shared state'
  - If a process has state, it means it cannot run without external information
- Consider 'idempotency':
  - Multiple operations can be called, but only the initial result will be used.

```
If alreadyRun:
   return result
result = processThings()
return result
```

Roughly this, but there are nuance here… check with the libraries you're using to see how to do this safely!

Other Tools

- Docker! https://www.docker.com/
  - Isolate your strange dependencies, so you can deploy on anything ☺
  - https://docs.docker.com/guides/text-classification/

- My own tutorials (slightly old)
  - https://johnvidler.co.uk/blog/docker-101/
  - https://johnvidler.co.uk/blog/docker-102/

USAS-based Sentiment Analysis

4DPicture

The 4D PICTURE project aims to help cancer patients, their families, and healthcare providers better understand their options. It supports their treatment and care choices, at each stage of disease.

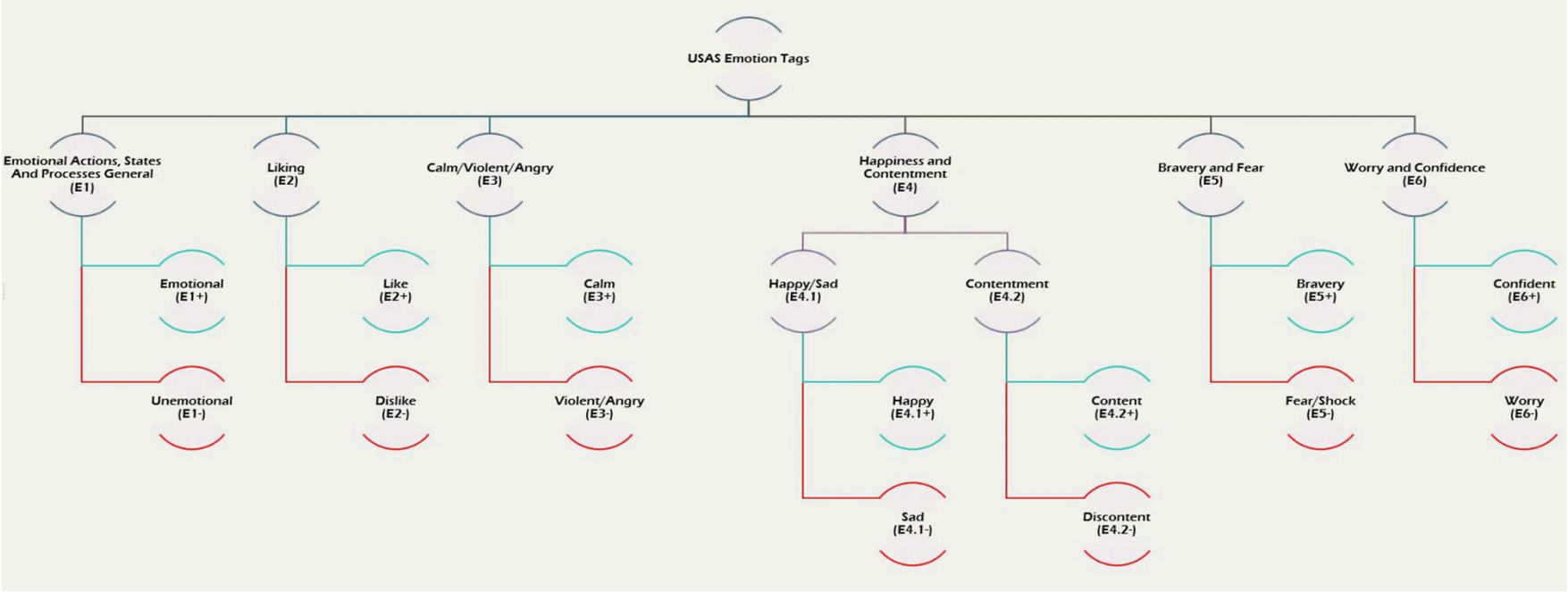
The project's primary objective is to improve decision-making about cancer treatment by better predicting treatment outcomes by developing data-driven algorithms, resulting in decision-support tools for people with breast cancer, prostate cancer, or melanoma.

Design-based Data-Driven Decision-support (4D) Tools

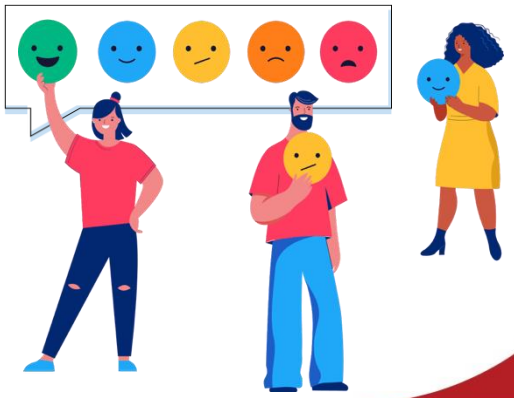
Producing Improved Cancer Outcomes Through User-centered Research

USAS-based Sentiment Analysis

Code



https://drive.google.com/drive/folders/1Yaqr-PrtaB14Oqhx8etdqoBlIeVuQ8n8?usp=sharing



Continue with your tasks!!

Lancaster University

- https://ucrel.lancs.ac.uk/wmatrix/tutorial7/
- On your own or in small groups ...
  - **Do** tutorials A and B (you can either upload the manifesto documents yourself into Wmatrix, or use the ones I made earlier in the corpus library)
  - **Do** tutorial C (key words, key domains and concordances)

  - Ask questions any time!
  - Chance to provide feedback and influence future plans!

Thanks for listening!

- Questions and comments?

- PyMUSAS collaboration for existing and new languages welcome!!

- Contact:
  - Email: p.rayson@lancaster.ac.uk
  - https://bsky.app/profile/perayson.bsky.social

- Icons from https://openmoji.org/

Key papers

- Rayson, P., Archer, D., Piao, S. L., McEnery, T. (2004). The UCREL semantic analysis system. In proceedings of the workshop on Beyond Named Entity Recognition Semantic labelling for NLP tasks in association with 4th International Conference on Language Resources and Evaluation (LREC 2004), 25th May 2004, Lisbon, Portugal, pp. 7-12.
  - http://www.lancaster.ac.uk/staff/rayson/publications/usas_lrec04ws.pdf

- Piao, S., Bianchi, F., Dayrell, C., D'Egidio, A. and Rayson, P. (2015). Development of the multilingual semantic annotation system. In proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT 2015), Denver, Colorado, United States, pp. 1268-1274.
  - http://aclweb.org/anthology/N/N15/N15-1137.pdf

Key papers

- Piao, Scott Songlin; Rayson, Paul; Archer, Dawn; Bianchi, Francesca; Dayrell Gomes Da Costa, Maria Carmen; El-Haj, Mahmoud; Jiménez, Ricardo-María; Knight, Dawn; Křen, Michal; Lofberg, Laura; Nawab, Rao Muhammad Adeel ; Shafi, Jawad; Teh, Phoey Lee; Mudraya, Olga / Lexical Coverage Evaluation of Large-scale Multilingual Semantic Lexicons for Twelve Languages. 2016. In proceedings of the 10th edition of the Language Resources and Evaluation Conference (LREC2016), Portorož, Slovenia, pp. 2614-2619.
  - http://www.lrec-conf.org/proceedings/lrec2016/pdf/257_Paper.pdf
- El-Haj, M., Rayson, P., Piao, S., & Wattam, S. (2017). Creating and validating multilingual semantic representations for six languages: expert versus non-expert crowds. In Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. (pp. 61-71). Association for Computational Linguistics.
  - http://aclweb.org/anthology/W17-1908

References …

- Wmatrix, CLAWS and USAS websites:
  - https://ucrel.lancs.ac.uk/wmatrix/
  - https://ucrel.lancs.ac.uk/claws/
  - https://ucrel.lancs.ac.uk/usas/
- Semantic lexicon expansion
  - Sheryl Prentice, Paul Rayson, Jo Knight, Mahmoud El-Haj, Solly Elstein (2021) A Domain Based Approach to Semantic Lexicon Expansion, International Journal of Lexicography. https://doi.org/10.1093/ijl/ecab028
- Useful background reading (keyness, annotation and MWE):
  - Rayson, P., Archer, D., Piao, S. L., McEnery, T. (2004). The UCREL semantic analysis system. In proceedings of the workshop on Beyond Named Entity Recognition Semantic labelling for NLP tasks in association with 4th International Conference on Language Resources and Evaluation (LREC 2004), 25th May 2004, Lisbon, Portugal, pp. 7-12. http://www.lancaster.ac.uk/staff/rayson/publications/usas_lrec04ws.pdf
  - Rayson, P. (2008). From key words to key semantic domains. International Journal of Corpus Linguistics. 13:4, pp. 519-549.
  - Piao, S., Rayson, P., Archer, D., McEnery, T. (2005) Comparing and combining a semantic tagger and a statistical tool for MWE extraction. Computer Speech and Language, 19 (4), pp. 378 – 397 http://dx.doi.org/10.1016/j.csl.2004.11.002
  - Piao, S. (2002) Word alignment in English-Chinese parallel corpora. Literary and linguistic computing, 17 (2), 207-230. doi:10.1093/llc/17.2.207

Further reading …

- Baker, P. (2004) Querying keywords: questions of difference, frequency and sense in keywords analysis. Journal of English Linguistics. 32: 4, pp. 346-359. DOI: 10.1177/0075424204269894
- Gries, S. T. (2006). Exploring variability within and between corpora: some methodological considerations. Corpora 1(2), pp. 109-151. http://www.eupjournals.com/doi/abs/10.3366/cor.2006.1.2.109
- Gabrielatos, C. and Marchi, A. (2012) Keyness: Appropriate metrics and practical issues. CADS International Conference 2012. Corpus-assisted Discourse Studies: More than the sum of Discourse Analysis and computing?, 13-14 September, University of Bologna, Italy.
- Hardie, A. (2014) Log Ratio – an informal introduction. http://cass.lancs.ac.uk/log-ratio-an-informal-introduction/
- Leech, G. and Fallon, R. (1992). Computer corpora - what do they tell us about culture? ICAME Journal, 16, pp. 29 - 50. http://icame.uib.no/archives/No_16_ICAME_Journal_index.pdf
- Mahlberg, M. (2007). Clusters, key clusters and local textual functions in Dickens. Corpora 2 (1), pp. 1-31. http://www.eupjournals.com/doi/abs/10.3366/cor.2007.2.1.1
- Rayson, P., Leech, G., and Hodges, M. (1997). Social differentiation in the use of English vocabulary: some analyses of the conversational component of the British National Corpus. International Journal of Corpus Linguistics. 2 (1), pp 133 - 152. http://ucrel.lancs.ac.uk/papers/rlh97.html
- Scott, M. (1997). PC analysis of key words - and key key words. System 25 (2), pp. 233 - 245.
- Adam Kilgarriff (2005) Language is never ever ever random. Corpus Linguistics and Linguistic Theory 1 (2): 263-276. http://www.kilgarriff.co.uk/Publications/2005-K-lineer.pdf
- Baron, A., Rayson, P., & Archer, D. (2009). Word frequency and key word statistics in corpus linguistics. Anglistik, 20(1), 41-67.
- Rayson, P. and Potts, A. (2021) Analysing keyword lists. In Gries, S. Th. And Paquot, M. (eds.) A Practical Handbook of Corpus Linguistics. Springer.

Acknowledgements

- Wmatrix was initially developed within the REVERE project (REVerse Engineering of Requirements) funded by the EPSRC, project number GR/MO4846, 1998-2001. Collocation Network Explorer (CONE), developed by David Gullick, was partly funded by an EPSRC vacation bursary at Lancaster University in 2010, and incorporates a collocation library designed by Scott Piao.
- Ongoing maintenance of taggers (e.g. Linux porting work by Stephen Wattam), development of new components (e.g. L-gram developed by Eddie Bell, C-grams developed by Andrew Stone, Java taggers developed by Scott Piao, Python 'pymusas' developed by Andrew Moore) and dictionary updates (e.g. by Sheryl Prentice) are funded by user licence fees.
- Metaphor extensions have been developed in the MELC project (Metaphor in end-of-life care) funded by the ESRC (grant reference ES/J007927/1). The Historical Thesaurus Semantic Tagger (HTST) was developed in the SAMUELS project (Semantic Annotation and Mark-Up for Enhancing Lexical Searches) funded by the AHRC in conjunction with the ESRC (grant reference AH/L010062/1). Welsh semantic tagger developed in the CorCenCC project funded by ESRC and AHRC (grant reference ES/M011348/1).