GATE General Architecture for Text Engineering

Module 9: Online Abuse Detection

Diana Maynard (d.maynard@sheffield.ac.uk)

GATE General Architecture for Text Engineering

This session will be recorded

Recorded video will be available after this session

Warning:

these slides and hands-on material contain swear words and abusive terms

GATE General Architecture for Text Engineering

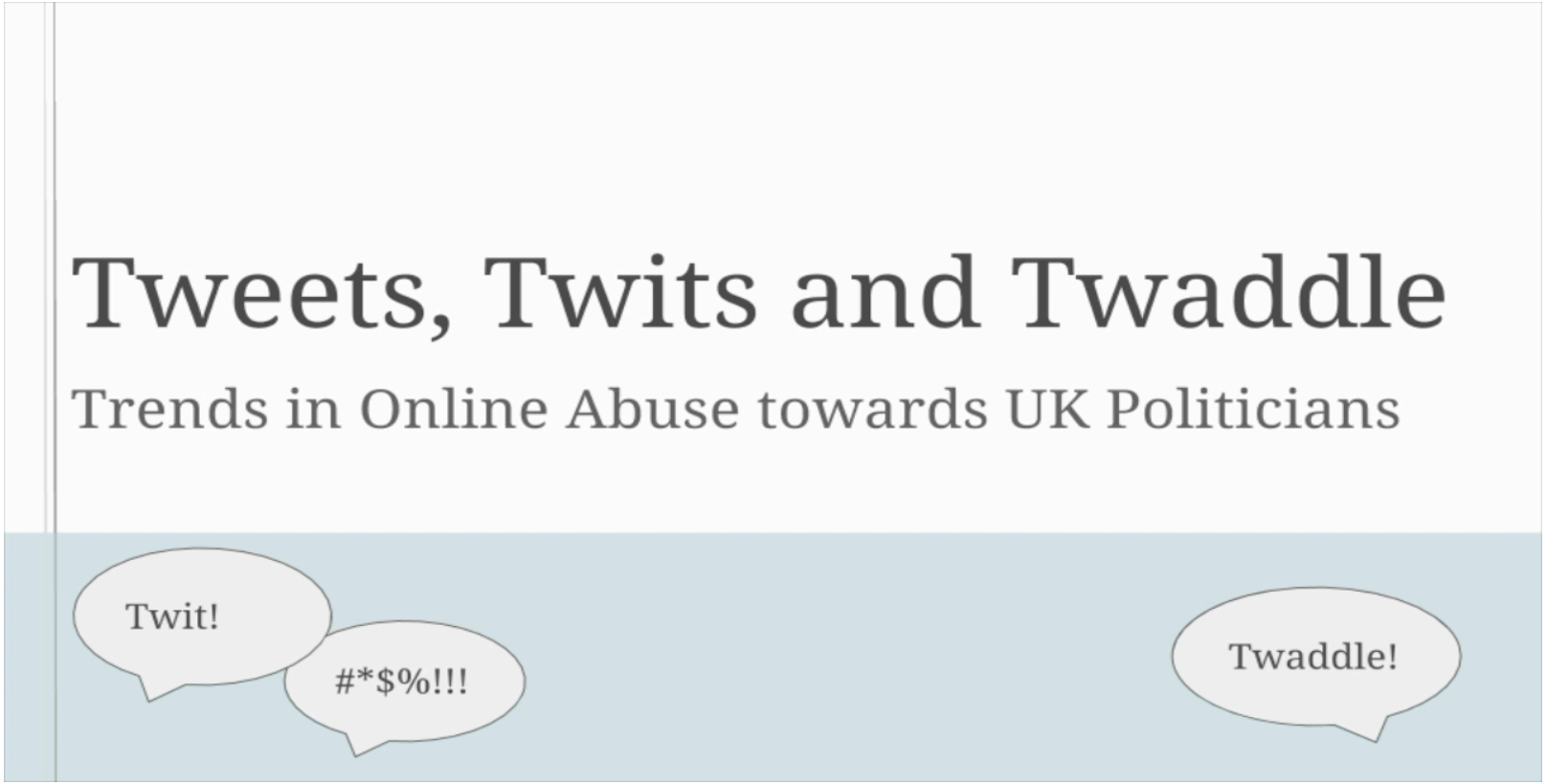
Aims of the session

- Background and overview of the approach
- The application
- Details of the gazetteer
- Details of the JAPE rules
- Extending the application
- Hands-on practice

GATE General Architecture for Text Engineering

Why?

Diane Abbott is going to tell us.



GATE General Architecture for Text Engineering

Background

- Detection of online abuse is quite complicated
- What is considered abusive?
- Abusive typologies

GATE General Architecture for Text Engineering

Some common types of abuse

- Racist, sexist, homophobic…..
  - whore, faggot, etc.
- Sexual
  - terms relating to body parts and sex acts
- Reputational (e.g. undermining journalist credibility)
  - fake news queen, liar
- General
  - f*** off, you piece of filth, etc.
- Threats
  - You deserve to die, I’m going to **** you
- It can be hard to distinguish between types
- How to represent multiple types, e.g. ”lying bitch”

GATE General Architecture for Text Engineering

An abuse typology

- There are many possible typologies
- This one distinguishes between 4 ways of expressing offensive statements:
  - **Directed:** Aimed at a specific person
  - **Generalized**: Aimed at a general demography
  - **Explicit**: Often uses common keywords/slurs
  - **Implicit**: Uses coded language, e.g. “skypes” for Jews, “Google” for n-word.

GATE General Architecture for Text Engineering

A typology

	Explicit	Implicit
Directed	• Unambiguous in its potential to be abusive • Use of slurs directed at an individual/entity	• Not immediately clearly abusive. • Often obscured by ambiguous terms, sarcasm, lack of profanity, etc. • Directed at an individual/entity
Generalised	• Unambiguous in its potential to be abusive • Use of slurs directed at a generalised *other.*	• Not immediately clearly abusive. • Often obscured by ambiguous terms, sarcasm, lack of profanity, etc. • Directed at a generalised *other*.

GATE General Architecture for Text Engineering

Examples

	Explicit	Implicit
Directed	@User shut yo beaner ass up sp*c and hop your f*ggot ass back across the border little n*gga”	“((((@User))) and what is your job? Writing cuck articles and slurping Google balls? #Dumbgoogles”
Generalized	So an 11 year old n*gger girl killed herself over my tweets? ^ ^ thats another n*gger off the streets!!”	“Totally fed up with the way this country has turned into a haven for terrorists. Send them all back home.”

GATE General Architecture for Text Engineering

Identifying abuse types

- **Directed**: Mentions, proper nouns, POS, and named entities can all be used in different contexts to identify targets.
- **Generalised**: Lexical features, named demographies
- **Explicit**: Bad-words dictionaries, polarity, sentiment
- **Implicit**: Euphemisms, word embeddings, named demographies

GATE General Architecture for Text Engineering

Example Annotation Guidelines

- Uses a sexist or racist slur
- Attacks a minority
- Seeks to silence a minority
- Criticises a minority (without a well-founded argument)
- Promotes, but does not directly use, hate speech or violent crime
- Blatantly misrepresents truth or seeks to distort views on a minority with unfounded claims
- Shows support of problematic hashtags, e.g. #BanIslam, #whoriental, #whitegenocide
- Negatively stereotypes a minority, defends xenophobia or sexism
- Contains a screen name that is offensive, as per the previous criteria; the tweet is ambiguous (at best); and the tweet is on a topic that satisfies any of the above criteria

GATE General Architecture for Text Engineering

GATE-Hate: Overview of Approach

- First we identify relevant vocabulary, such as slurs, profane language and words that identify people
- Then we use linguistic rules to combine the terms to decide if abusive/hateful language was indeed used, what we can say about the type of the abuse (racist etc.) and who it was aimed at

GATE General Architecture for Text Engineering

Advantages of the approach

- Control and reproducibility
- Ability to manually correct new word usage issues
- Resistance against biased datasets, which can tend to learn stereotypes (naturalistic data tends to be biased)
- The rule-based approach is fast and stable to update and run
- On independent evaluation data, the accuracy is 80% and the precision is 72%
- Evaluation datasets
  - Kaggle https://www.kaggle.com/c/detecting-insults-in-social-commentary/data
  - OLID https://competitions.codalab.org/competitions/20011

GATE General Architecture for Text Engineering

Problems and disadvantages

- Abuse is hard to define, and different datasets label data very differently
- We use Kaggle's 2012 "Detecting Insults in Social Commentary", which is moderate, common sense. Other datasets seem to be oversensitive.
- Long tail of more complex language which is harder to detect
- We can see the examples we find, therefore, as indicative of a larger problem.
- Recall on the Kaggle dataset is 0.47: while it’s successful for general abuse, more specific abuse types (e.g. religious) can be more problematic, as language is more complex.

GATE General Architecture for Text Engineering

Accessing the application

- The application is in development and is not a “finished” product - this version was developed for a specific task
- A version of the application is available on GATE Cloud, via a REST API or for batch processing https://cloud.gate.ac.uk/shopfront/displayItem/gate-hate
- This version is also available as a Google Sheets application (not currently publicly available)
- Today we will look at the application using the GATE GUI

The Application

GATE General Architecture for Text Engineering

Background to the Application

- This version of the application was designed to recognise abuse towards UK politicians in tweets
- See e.g. this paper for more info Gorrell, Genevieve, et al. "Which politicians receive abuse? Four factors illuminated in the UK general election 2019." *EPJ Data Science* 9.1 (2020): 18.
- It's designed to analyse tweets by MPs as well as replies to them or message mentioning them written by anyone.
- It's designed to maximise precision by looking for some very specific contextual patterns containing abuse, rather than any mention of an abusive word
- We want to be sure the abuse is actually directed at the MP rather than at someone or something else

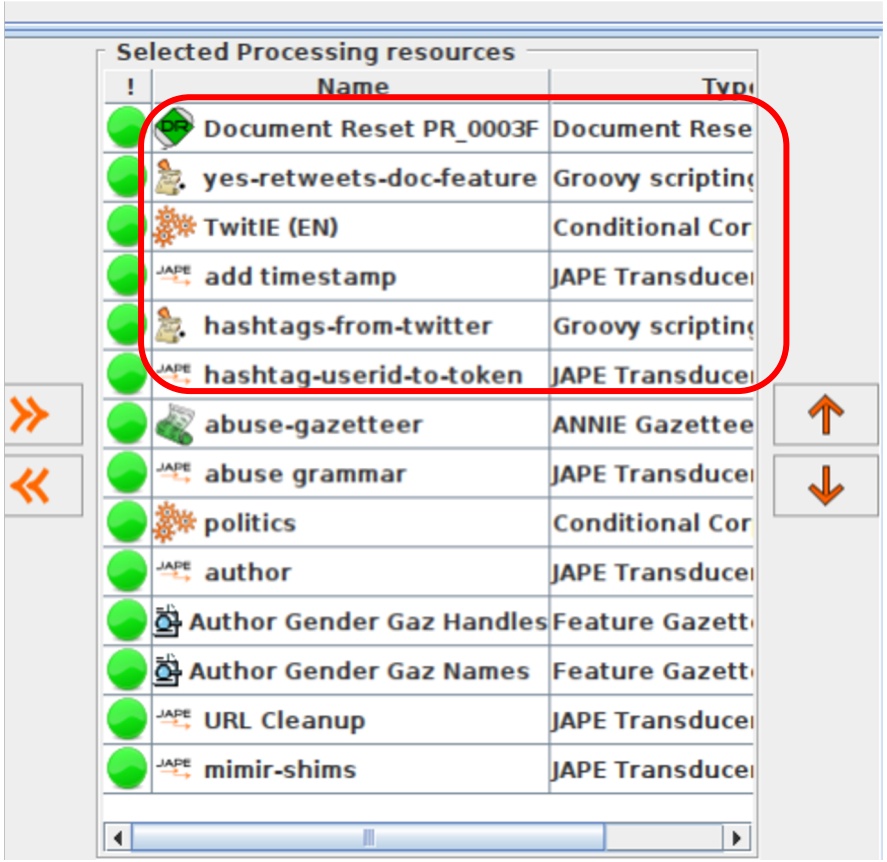
GATE General Architecture for Text Engineering

Identifying the right idiot

- Just mentioning the word “idiot” isn’t precise enough
  - “I’m an idiot” – self-abusive
  - “You idiot!” – abusive towards addressee
  - “What kind of idiot would do that?” – ambiguous (could be subtly abusive towards addressee or other person, or more general)
  - “They’re idiots” – abusive towards others (not addressee)
  - “@butterfly is an idiot” – directed towards a specific person
- We try to identify who the target of the abuse is (just like with the opinion mining application)

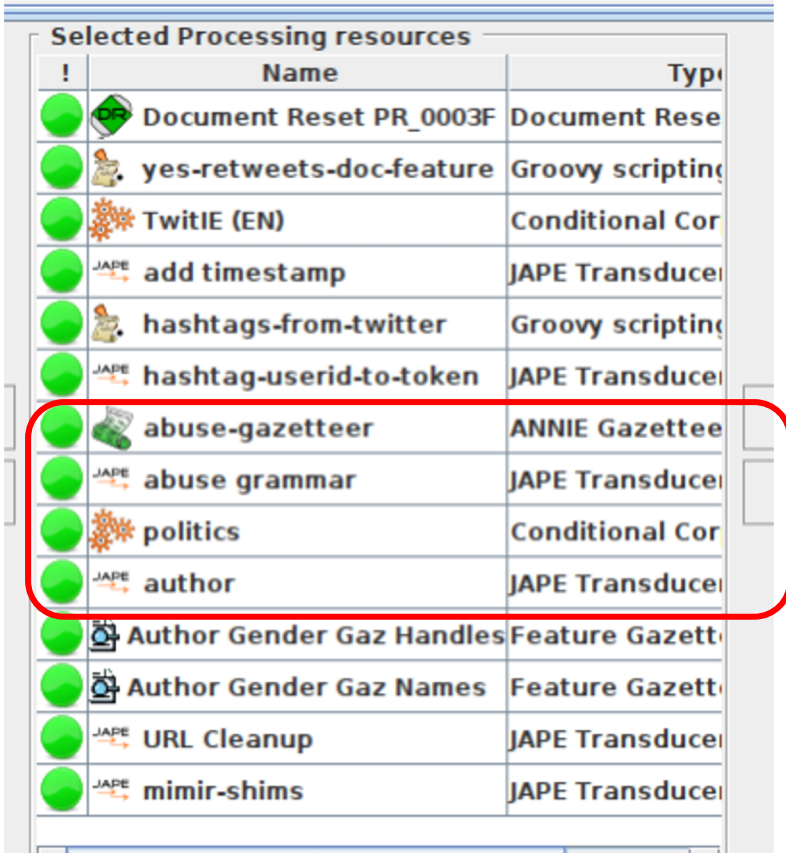
The Application - Pre-Processing

- Reset: clear existing annotations from the document
- A groovy script sets a flag "yes-retweets" - don't worry about this
- Use TwitIE to tokenise and do some basic NER
- The timestamp is taken from the tweet and added as a document feature
- Hashtags are also taken from the tweet and used to replace the ones spotted in the tweet text by TwitIE, as they may be truncated (only applies to old-style tweets)



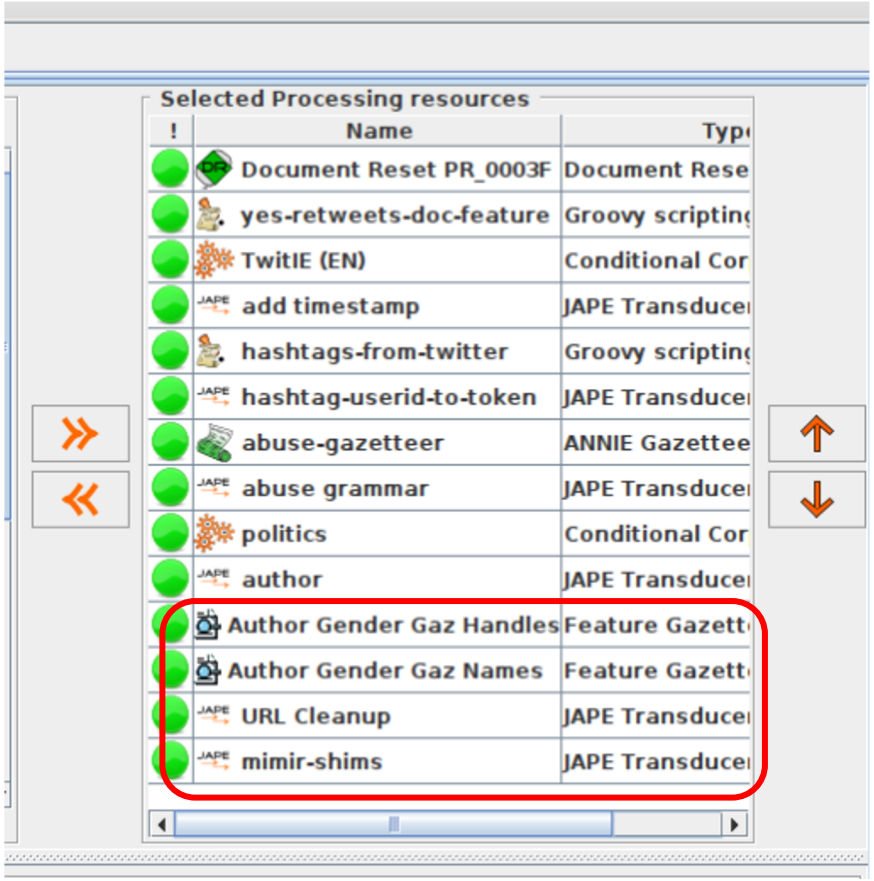
The Application - Finding Abuse

- Main components are a gazetteer (word lists) and a grammar (JAPE rules to combine the terms found by the gazetteer).
- "Politics" is a sub-application that finds mentions of politicians and adds information we know about them
- It adds this information also when a politician authored the tweet or when it was a reply to or a retweet of them



The Application - Post-Processing

- Tweet author names are annotated for gender based on a name gender gazetteer
- Abuse terms within URLs are removed
- Mimir-shims can be ignored
- Much of the app is relevant to Twitter research, especially politics, but the abuse component will work on any text (with some caveats)



The Gazetteer

GATE General Architecture for Text Engineering

Gazetteers

- The basis of the approach to finding abuse is three word lists:
  - 1081 **slurs**: words that are intrinsically offensive, e.g. "twat", "raghead".
  - 131 **offensive** words, such as "fuck" and "bloody". These words aren't abuse themselves, but intensify existing abuse or turn sensitive words into abuse.
  - 451 **sensitive** words, e.g. "gay", "black". These are also not intrinsically offensive, but when used with an offensive word or slur, become part of the abuse.
- The word lists have been built up over several years, from many sources, and manually tuned based on evaluation of terms and usage change
- Slurs can be racist, sexist etc, and sensitive words also may indicate if a word is racist, sexist, homophobic etc.

GATE General Architecture for Text Engineering

Veto - not abuse

- "not-abuse" is a veto list of terms that are not abuse (but would otherwise be annotated as abuse)
- We often find new names that are confused with abuse. The veto list makes it easy to rule these out and keep precision up.

cressida dick	type	veto
ed balls	type	veto
got your ass	type	veto
jonathan spink	type	veto
knee jerk	type	veto
lucky bitch	type	veto
my nigga	type	veto
polish a turd	type	veto
poor bastards	type	veto
pussy cat	type	veto
pussy cats	type	veto
pussy footing	type	veto
pussy grabbing	type	veto
pussy whipped	type	veto
slag off	type	veto
witch hunt	type	veto
witch-hunt	type	veto
witch hunted	type	veto
witch-hunted	type	veto
witch hunting	type	veto

GATE General Architecture for Text Engineering

Offensive words

- "offensive words" are not abuse in themselves, but can tip things over into being abusive.
- e.g. "bloody Jews" is abusive, though "bloody" in itself isn't.
- Sequences of offensive words tend to be abusive.

bloody	strength	mild		
blooming	category	adjective	strength	mild
bollocks	strength	medium		
bollox	strength	medium		
boner				
boob	type	sexist		
boobies	type	sexist	origin	farrell
boobs	type	sexist	origin	farrell
booner	type	racist	origin	farrell
boong	type	racist	origin	farrell
boonga	type	racist	origin	farrell
boonie	type	racist	origin	farrell
booobs	type	sexist	origin	farrell
boooobs	type	sexist	origin	farrell
booooobs	type	sexist	origin	farrell
boooooooobs	type	sexist	origin	farrell
braindead	category	adjective		
brain dead	category	adjective		
bugger	strength	mild		
bullshit	strength	medium		

Identity - sensitive words

- "sensitive words" are identity terms, which have to be used carefully.
- In conjunction with offensive words, they become abusive.
- The list contains national/racial terms, sexuality and gender terms, religious and political terms.

anarchist	type	political		
anarchistic	type	political		
anarchists	type	political		
Angolan	type	racist	subtype	nationality
Angolans	type	racist	subtype	nationality
anti europe	type	political	subtype	leave
anti european	type	political	subtype	leave
Argentine	type	racist	subtype	nationality
Argentines	type	racist	subtype	nationality
asexual	community	gendernonconforming	type	homophobic
asian	type	racist	subtype	continent
asians	type	racist	subtype	continent
Australian	type	racist	subtype	nationality
Australians	type	racist	subtype	nationality
Austrian	type	racist	subtype	nationality
Austrians	type	racist	subtype	nationality
authoritarian	type	political		
authoritarians	type	political		
Bangladeshi	type	racist	subtype	nationality
Bangladeshis	type	racist	subtype	nationality
Belarusian	type	racist	subtype	nationality
Belarusians	type	racist	subtype	nationality
Belgian	type	racist	subtype	nationality

Slurs - expanded

- Slurs are words that intrinsically constitute abuse
- A core list, "slurs-core", is expanded to include plurals and bleeped out versions, to create this list of ~70k terms (slurs-expanded)

a5s bag	strength	medium		
a5s-bag	strength	medium		
a5sbags	plural	true	strength	medium
a5s bags	plural	true	strength	medium
a5s-bags	plural	true	strength	medium
a5sbandit	type	homophobic	community	men
a5s bandit	type	homophobic	community	men
a5s-bandit	type	homophobic	community	men
a5sbandits	plural	true	type	homophobic
a5s bandits	plural	true	type	homophobic
a5s-bandits	plural	true	type	homophobic
a5sbanger	type	homophobic	origin	farrell
a5s banger	type	homophobic	origin	farrell
a5s-banger	type	homophobic	origin	farrell
a5sbangers	plural	true	type	homophobic
a5s bangers	plural	true	type	homophobic
a5s-bangers	plural	true	type	homophobic
a5sbite	strength	medium		
a5s bite	strength	medium		
a5s-bite	strength	medium		

GATE General Architecture for Text Engineering

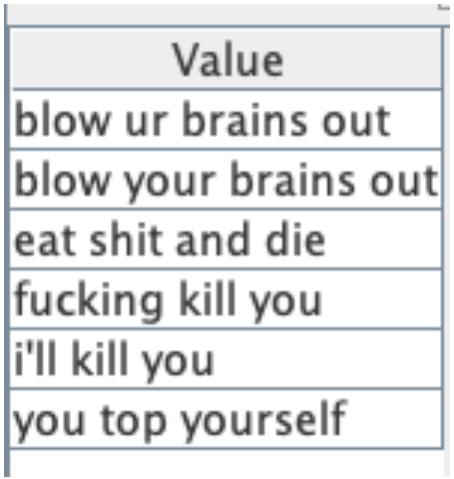
Slurs - no expansion

- This expansion doesn’t work for all slurs
- "slurs-no-permute" is for manually expanded terms such as phrases that are abusive but aren't epithets, and slurs with irregular pluralisation

get fucked	pos	youphrase
GFY	pos	youphrase
go back to where u came from	type	racist
go back to wherever u came from	type	racist
go back to wherever you came from	type	racist
go back to where you came from	type	racist
go back where u came from	type	racist
go back wherever u came from	type	racist
go back wherever you came from	type	racist
go back where you came from	type	racist
go fuck yourself	pos	youphrase
how dumb u are	pos	youphrase
how dumb you are	pos	youphrase
how stupid u are	pos	youphrase
how stupid you are	pos	youphrase
piece of excrement		

Threats - not really covered

- Threats aren't really covered as they are linguistically too irregular to work well with this approach
- The final gazetteer includes a few threat terms of sufficiently high precision, but the recall is very low



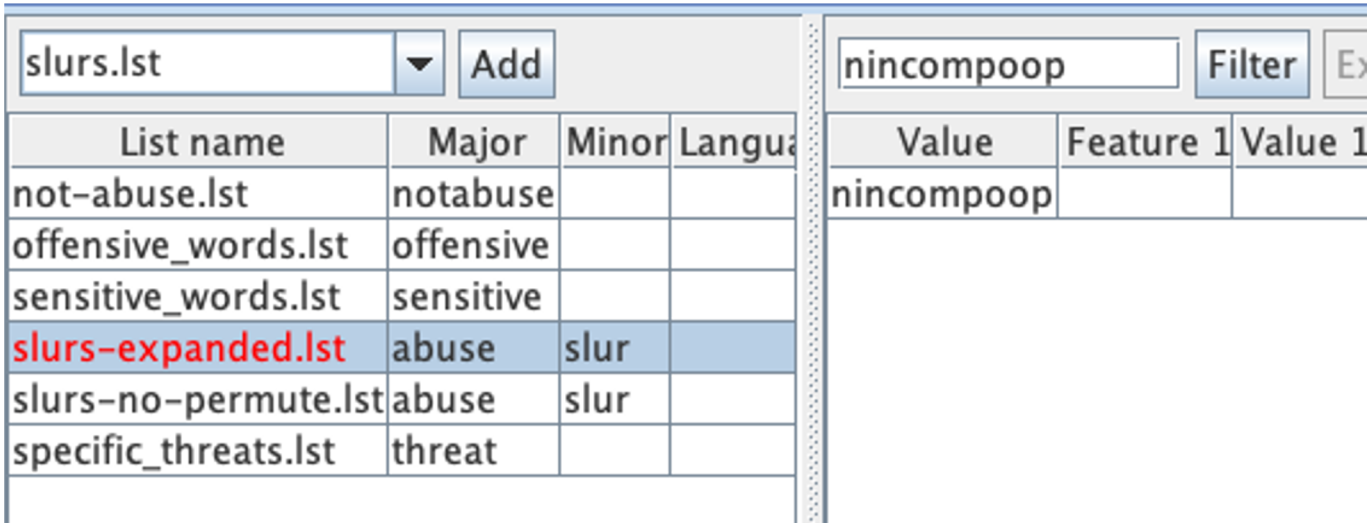
Value
blow ur brains out
blow your brains out
eat shit and die
fucking kill you
i'll kill you
you top yourself

GATE General Architecture for Text Engineering

Hands-on 1 – improving the gazetteers

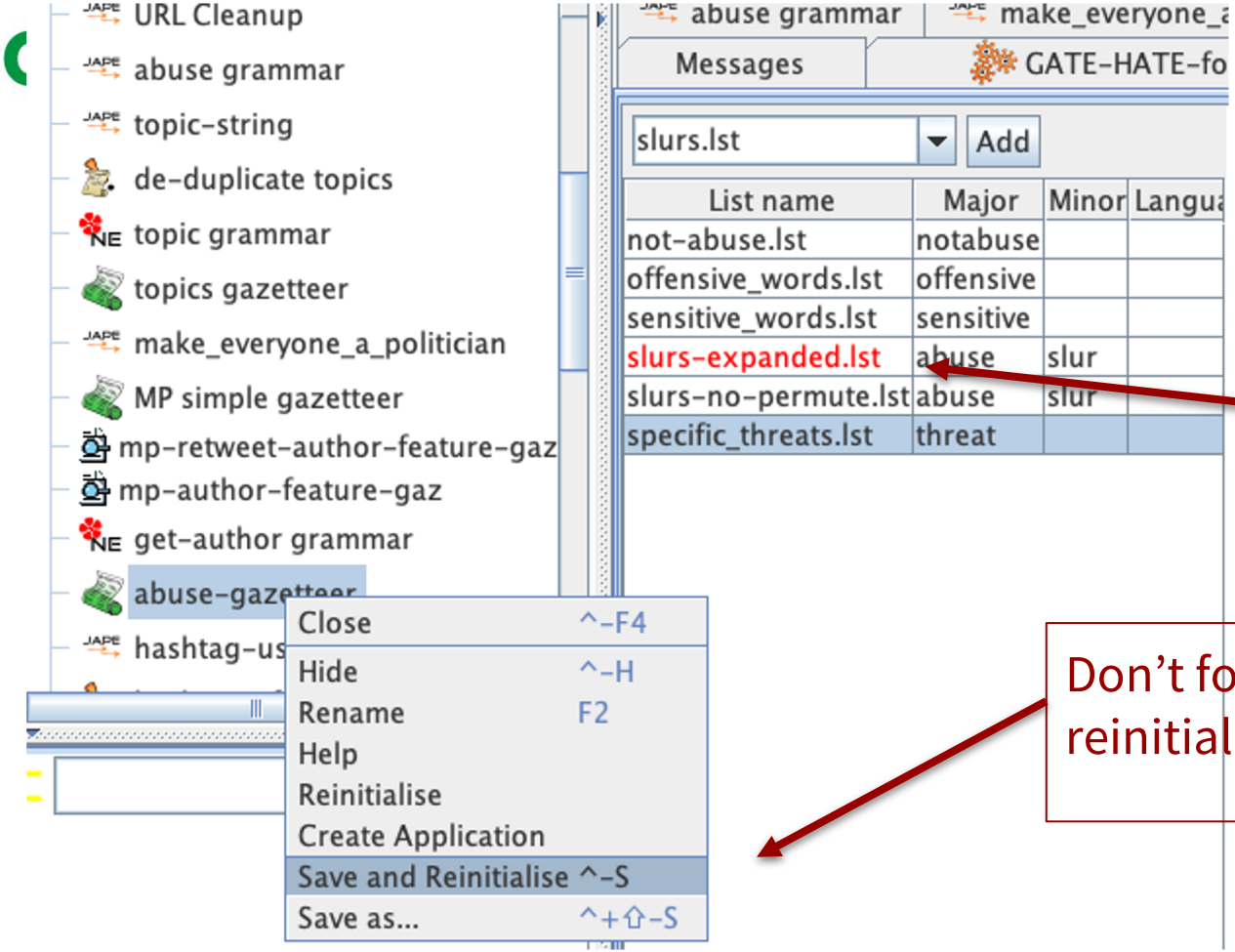
- Unzip the application from your hands-on and load the application.xgapp
- Try adding some new swear words to a gazetteer list - you can make up your own.
- Create a test document in a text editor which contains some sentences containing your new abusive words. Add it to a corpus.
- For example, add the term “nincompoop” to the slurs-expanded gazetteer list in the GUI. Don’t forget to reinitialise the list.
- Add the sentence “You are a complete and utter nincompoop.” to your test document.
- Run the application on your new corpus and check the results.

Adding to the gazetteer



- You can add features and values if you want, or leave them empty

Tips

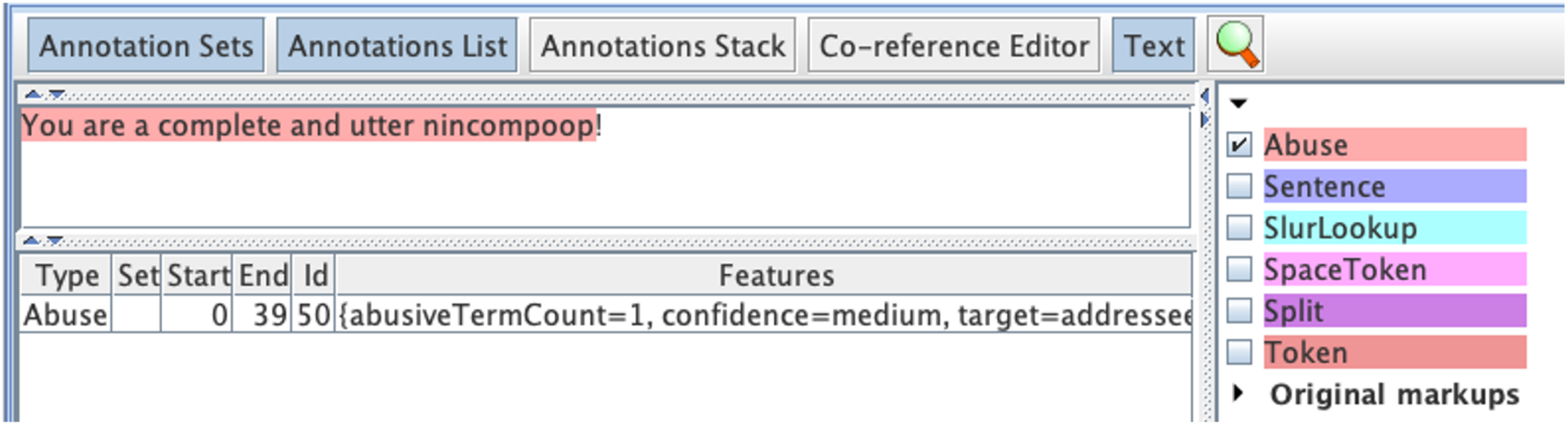


A red name means it isn't saved!

Don't forget to save and reinitialise your gazetteer list!

GATE General Architecture for Text Engineering

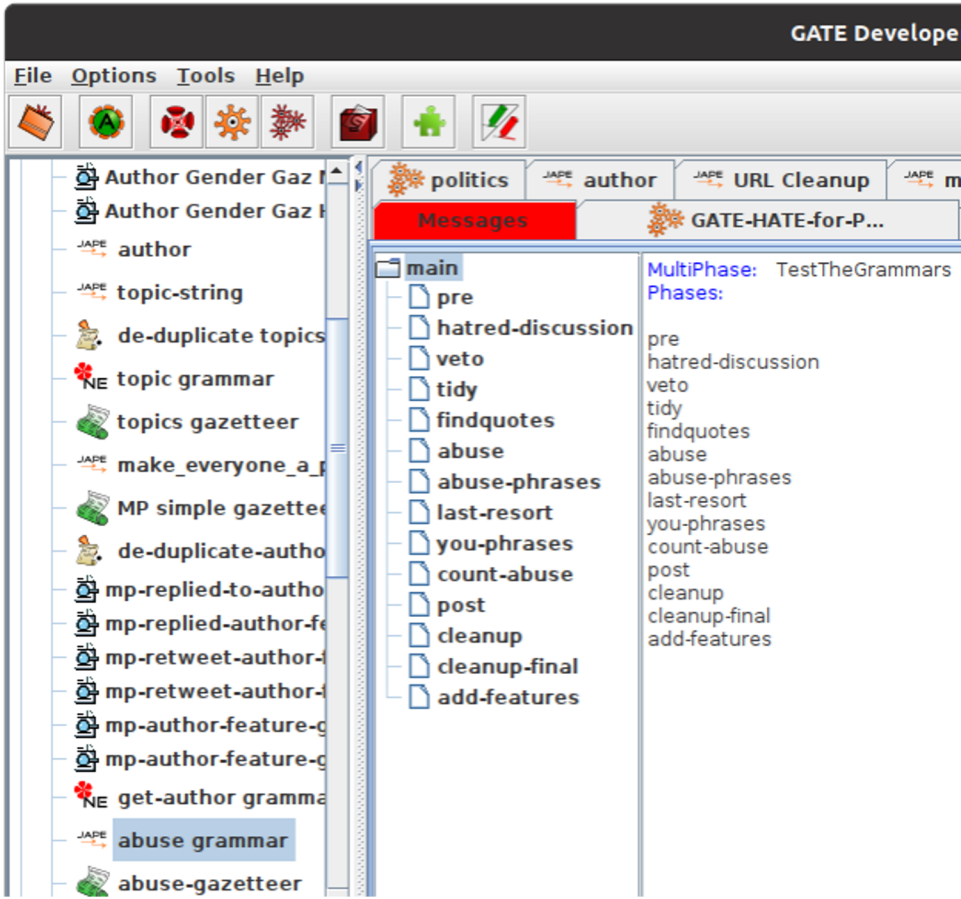
My result



The JAPE grammars

Overview

- We use a set of 31 rules to match sequences of the above, and assign features to the overall abuse term, such as "racist" or "homophobic".
- The rules are generally successful in identifying abuse accurately.
- Further rules then attempt to match the abuse term to a pronoun, to decide if the abuse is aimed at the tweet recipient or someone else.
- This is less successful as pronouns are often not used, leaving target to be inferred from context.



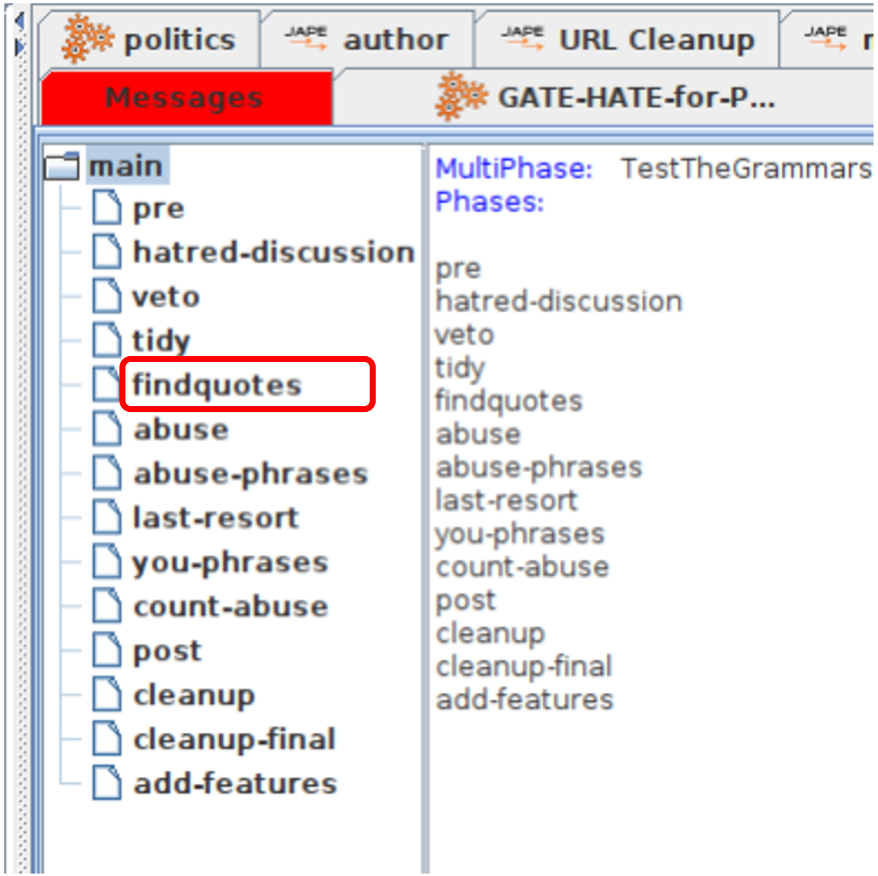
GATE General Architecture for Text Engineering

Identifying what is not abuse

- The first three phases centre on what isn't abuse.
- "**pre**" sets up some pronoun macros, and prepares hyphenated terms to be spotted in the next phase
- "**hatred-discussion**" - some of the hardest "not abuse" to filter out are cases where hatred is discussed. This phase spots some common examples
- The **veto** phase (as opposed to the veto gazetteer shown earlier) allows more flexible exclusion to be done than having to list every phrase to veto
- "**tidy**" then actions the above by removing gazetteer terms that would cause confusion later, as well as e.g. terms in URLs

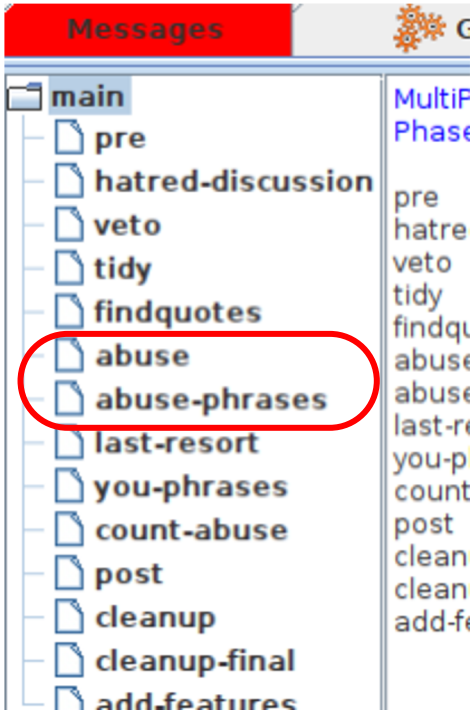
Find quotes

- "**findquotes**" puts a quote annotation on quoted items, for later use
- People very often repeat the abusive or offensive words of others, possibly critically, so we don't want to count this as their being abusive
- E.g. most usage of the N word in a politician study we did last year related to others' use of the word and how acceptable it was



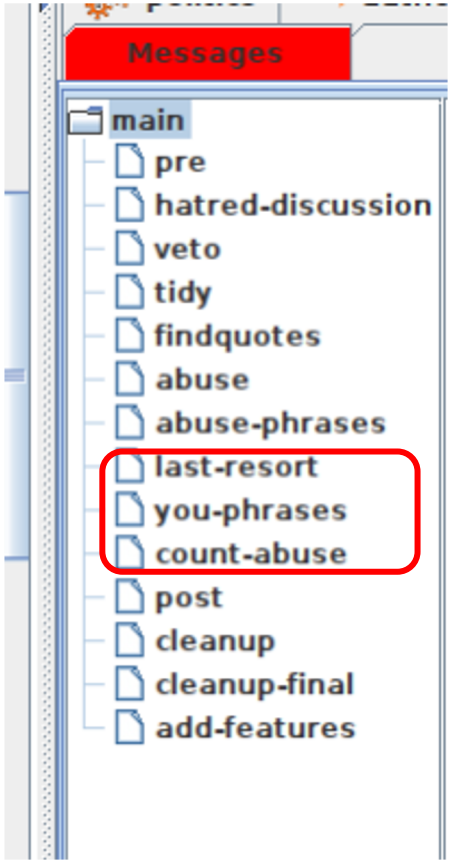
Abuse and its referents

- "**abuse**" spots 3 distinct types of abuse term;
  - Slur with optional sensitive markers
  - Combinations of at least one offensive word and at least one sensitive word
  - Sequences of offensive words
- "**abuse-phrases**" then seeks to match the abuse term to some kind of referent, e.g. a pronoun or a person mention
- This allows a "target" to be assigned (author, addressee or other) and a confidence depending on how tightly specified the phrase was



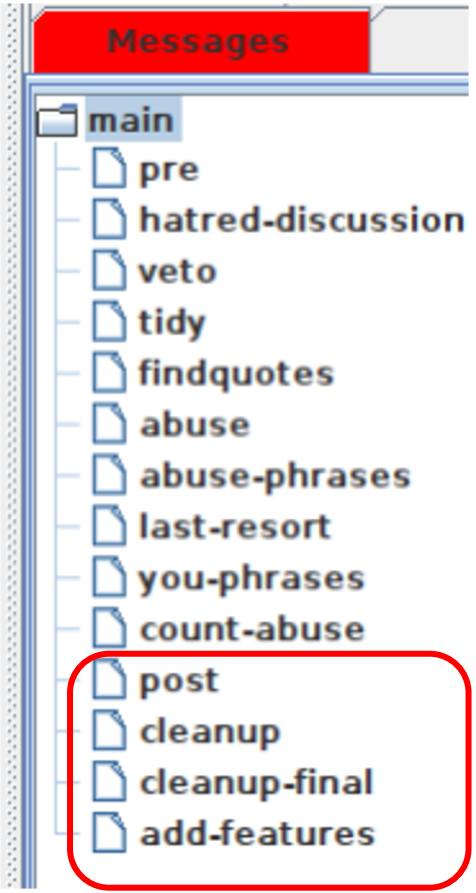
More referents

- "**last-resort**", if no target person is found, assigns a target based on plurality and gives it a low confidence
- In "**you-phrases**", the list of slurs already marked as "youphrase" in the gazetteer, e.g. "f**k you" or "go back where you came from" are marked as addressed to the addressee with high confidence
- In "**count-abuse**" we count the number of terms that have gone into this abuse, which might give some indicator of severity



Final stages

- In "**post**" we modify the target feature if the abuse was found between quote marks or in a quoted tweet or retweet. So if someone retweets someone that said "I'm an idiot", the target feature, marked "author", now becomes "author-retweet".
- "**cleanup**" and "**cleanup-final**" aim to remove intermediate annotations, but keep a record as it helps to trace why something was not annotated as abuse
- "**add-features**" adds the string of the abuse in original and lower case, and a "type", or "general" where there is no type (racism etc.) These features are helpful when we come to use the data



GATE General Architecture for Text Engineering

Final comments

- It's not perfect - this kind of approach will always have exceptions, and gets increasingly unwieldy to extend
- It's a research system so contains some bits relevant to previous work we did and other tools, such as indexing into Mimir
- This version of the application is quite basic, and not very generic
- What do you think we could do to improve it?

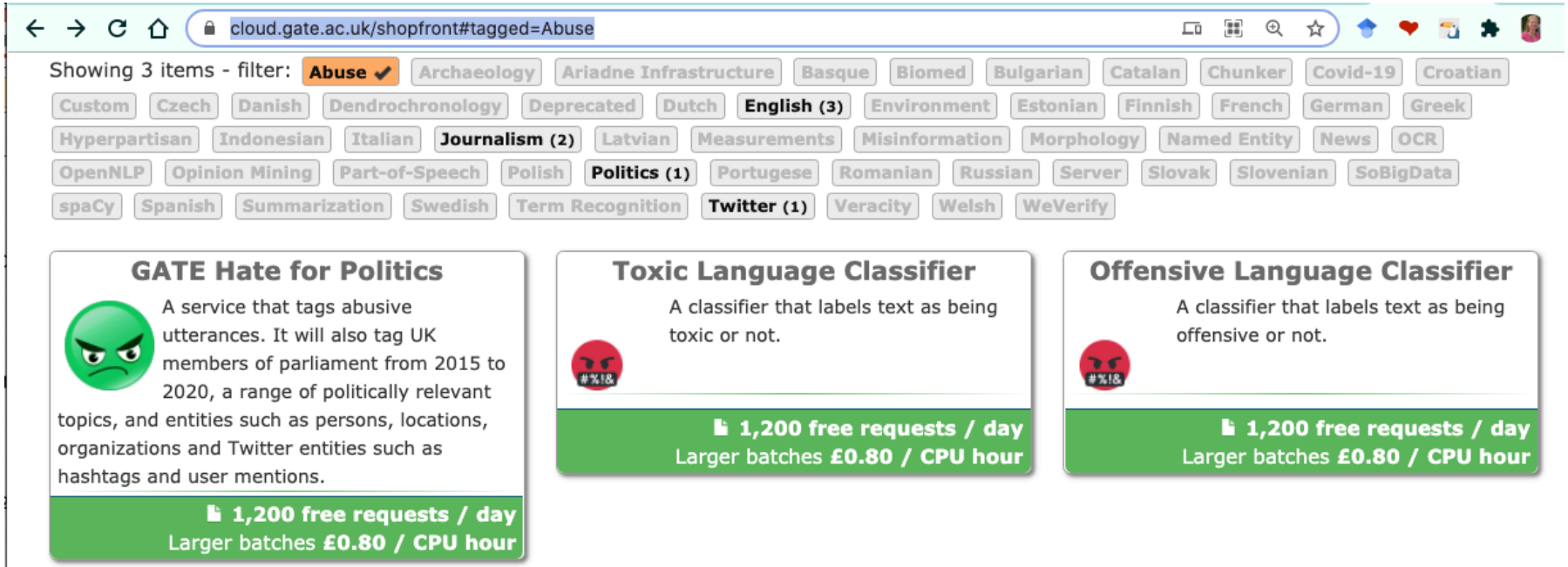
GATE General Architecture for Text Engineering

Ways to improve the application

- Better typology of abuse terms
- More abuse terms
  - Train on a big corpus (word embeddings, but typically needs manual verification)
  - Find other classifications (e.g. HateBase)
- More variants of abuse terms (linguistic variants)
- Better constraint of targets
- Better recognition of implicit abuse
- Targeted apps for specific cases (e.g. journalists, women, etc.)
- Combine with ML approaches
- We are working on all these things ☺

More ways to find abuse

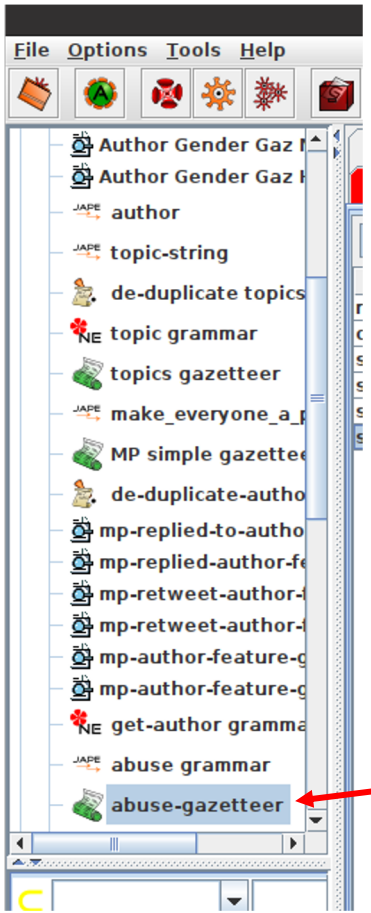
- 3 apps on the GATE Cloud
- The other two are based on ML and are much more generic
- They also capture a lot more abuse, but over-generate a lot



More hands-on

For the interested, adventurous, or bored

More Hands-on: gazetteers



- There are lots of things missing in the gazetteers – try adding your favourite insult and see if it gets recognised
- If not, don't worry, it might be connected with a JAPE rule (coming next)
- The gazetteers can be edited via the GATE GUI or in the text files, which are located under hate-resources/gazetteer/abuse
- If you want to add a whole new gazetteer file, you need to include it in the lists.def
- Tip: if you are working in the GUI but you updated your gazetteers outside GATE, you need to right click on the gazetteer in the left pane ("resources pane") and select "reinitialise" to pull the changes in

GATE General Architecture for Text Engineering

Try adding to different gazetteers

- Add a specific phrase to the slurs-no-permute list
- Add a threat to the threats list
- Add something that sounds like an abusive comment but isn't.
  - Which list would you put it in?
  - Try it and see if it works!
- Don't forget to reinitialise your gazetteer each time
- Don't forget to add relevant sentences to your document (or new documents)

GATE General Architecture for Text Engineering

Improve the abuse gazetteer (2)

- Try replacing the abuse gazetteer with an extended gazetteer
- Why might we want to do that?
  What might be a disadvantage?

GATE General Architecture for Text Engineering

Adventurous hands-on: Play with the application

- Make sure you keep a copy of the original application in case you mess it up!
- The best way to understand the app is to play with it and try things out
- Maybe you could add a new grammar rule?
- Adding a feature called “rule” on an annotation helps you figure out which JAPE rule was fired

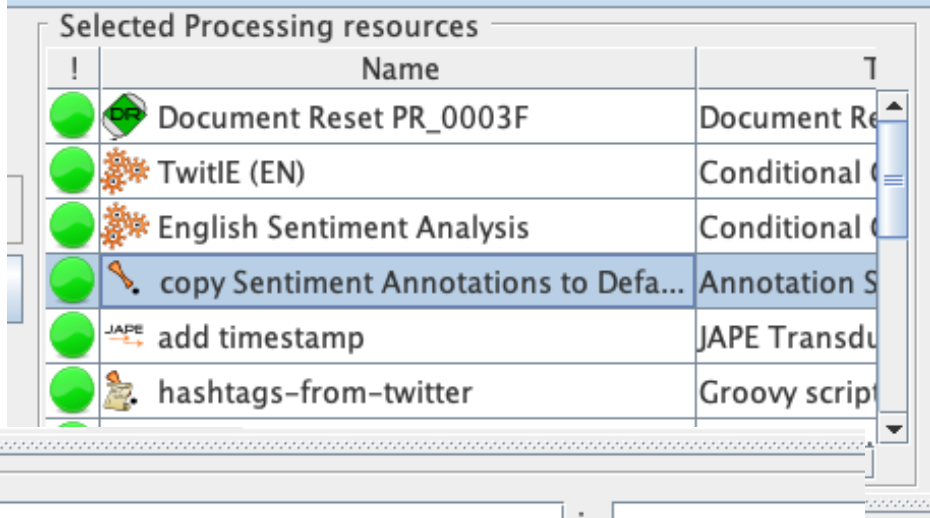
GATE General Architecture for Text Engineering

Adventurous hands-on (2): Combine the application with sentiment

- Add the sentiment application to the abuse application
- Hint: check Module 7 for how to combine two applications
- Hint: you may wish to copy or move some of the relevant Sentiment annotations from the Sentiment set into your working set
  - What PR do you need for this?
  - What parameters would you set?

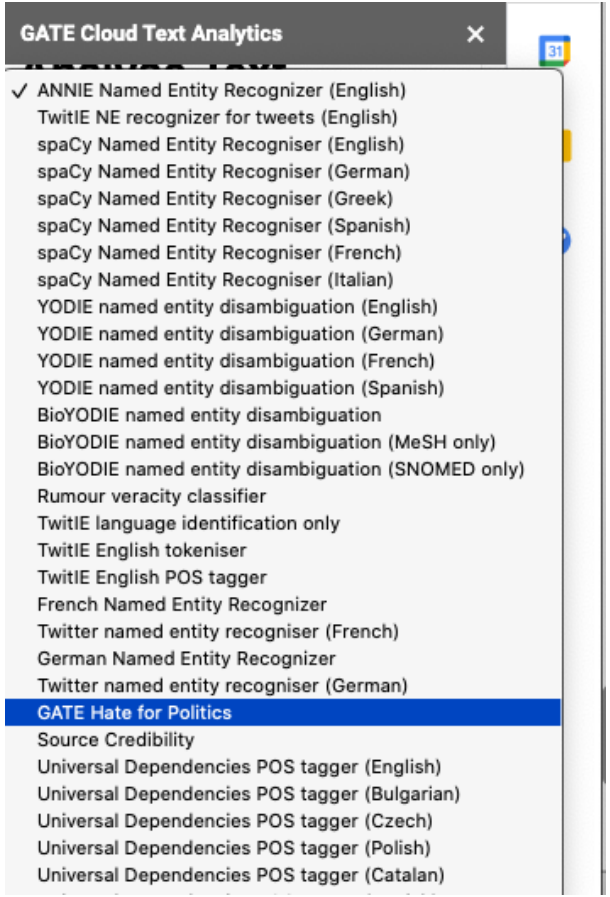
My solution

- Yours could be different



Hands-on with Google Sheets

- You can also find GATE-Hate as a Google Sheets app
- If you did module 3, you can try experimenting with the app in the same spreadsheet as you used before



GATE General Architecture for Text Engineering

Further reading

- Our work on social media and abuse finding https://gate-socmedia.group.shef.ac.uk/election-analysis-and-hate-speech/ (includes the full video from earlier)