GATE General Architecture for Text Engineering

Module 3: GATE Cloud

Introduction to GATE Cloud

Ian Roberts (i.roberts@sheffield.ac.uk)

GATE General Architecture for Text Engineering

This session will be recorded

Recorded video will be available after this session

GATE General Architecture for Text Engineering

GATE Cloud Overview

- Online platform operated by the GATE team
- https://cloud.gate.ac.uk
- Two principal strands
  - Dedicated servers
  - Processing pipelines
    - Free rate limited REST API
    - PAYG batch processing
- Originally launched around 2010

GATE General Architecture for Text Engineering

Dedicated servers - briefly

- Rent a virtual server pre-installed with our software
- Pay per hour of running time
- Two types of server available
  - Mímir indexing and search engine for GATE documents
  - Twitter data collector
- Covered in more detail this afternoon in Social Media module

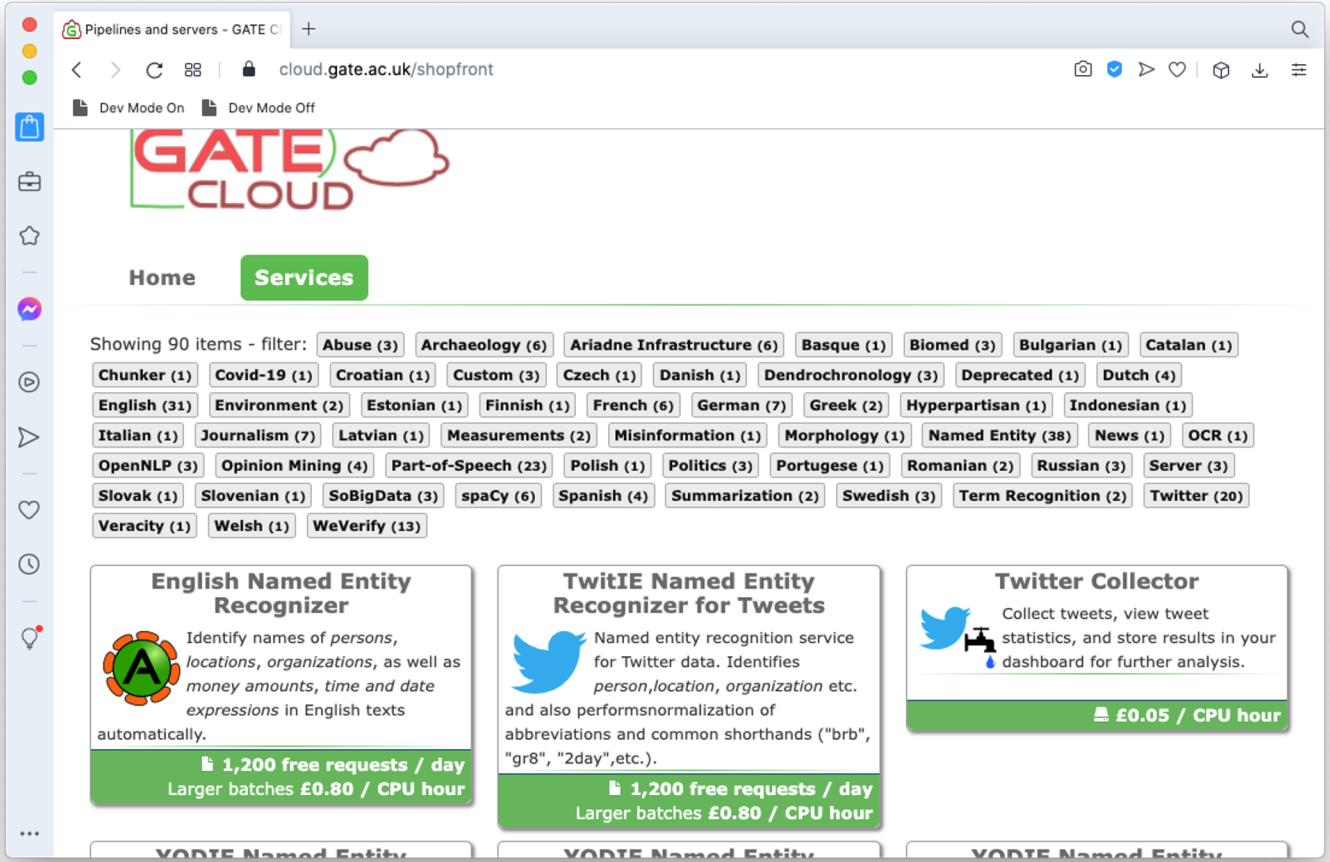
GATE General Architecture for Text Engineering

GATE Cloud Pipelines

- Recall: in GATE you build complex processing *pipelines* out of simple components
- GATE Cloud lets us publish “pipeline-as-a-service”
- 80+ pipelines available
- https://cloud.gate.ac.uk/shopfront

GATE General Architecture for Text Engineering

GATE Cloud Pipelines



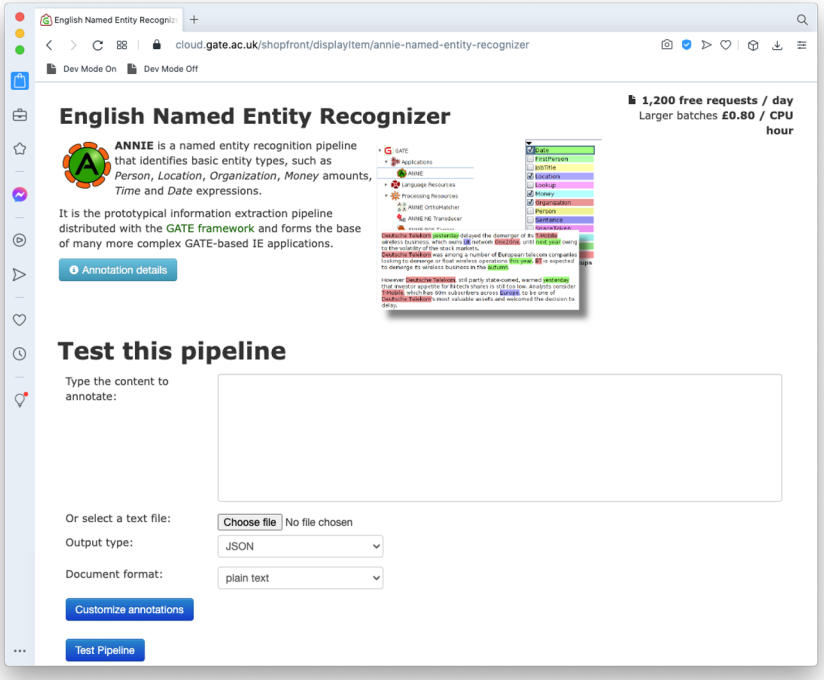
GATE General Architecture for Text Engineering

Using a GATE Cloud Pipeline

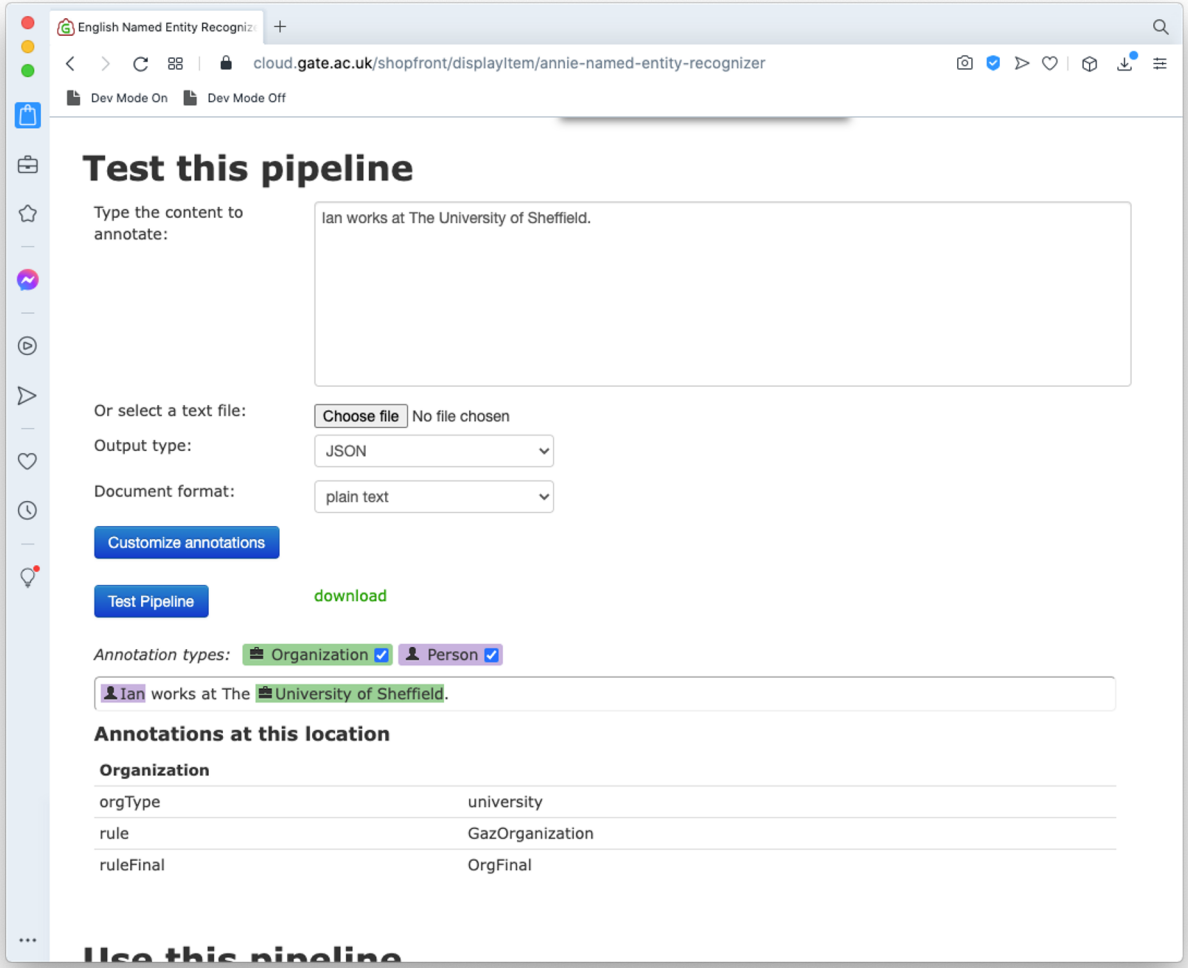
- Two interfaces
  - “Online” API
    - Send one text, get back annotations immediately
    - Free but quota-controlled
  - Batch “annotation job”
    - Process large bundles of data in parallel
    - Unlimited, pay-as-you-go
- We will focus mainly on online API

Trying it out

- You can try out any pipeline direct from the user interface
- Enter text you want to process
- Results shown as colour highlights, much like the GATE Developer UI
- Try it yourself!



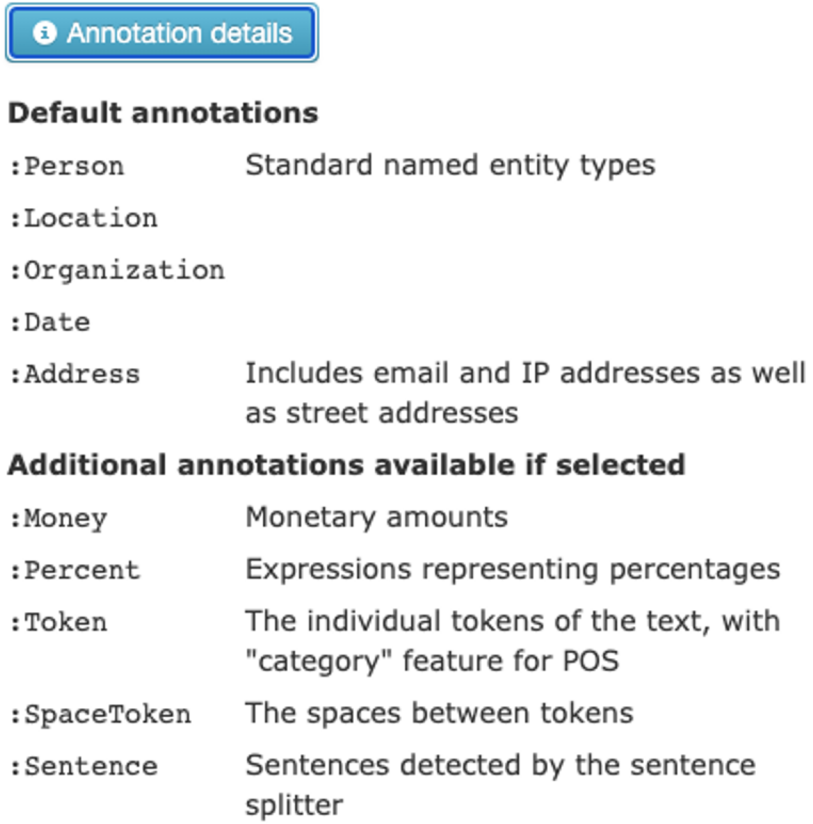
GATE General Architecture for Text Engineering



GATE General Architecture for Text Engineering

Setting parameters

- One pipeline can produce several different annotations
- Each pipeline specifies its “default” and “additional” annotation types
- Click “annotation details”

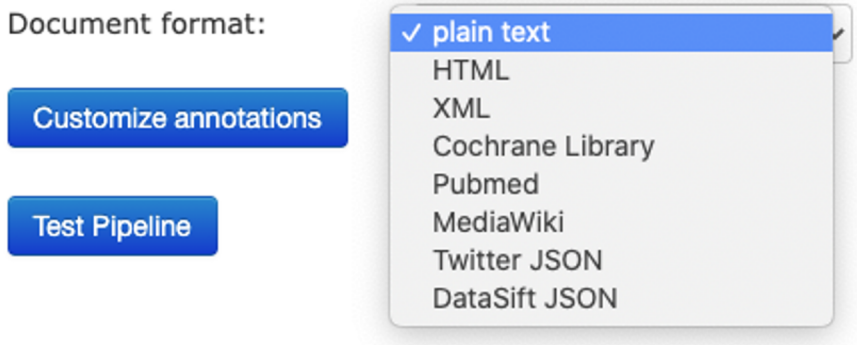


GATE General Architecture for Text Engineering

Setting parameters

- “Customize annotations” to select the types you want

- GATE Cloud services can handle any document type that GATE can parse - select “document format”



GATE General Architecture for Text Engineering

Example - processing HTML

Load an HTML file

Set the format

Text parsed out of the HTML

GATE General Architecture for Text Engineering

Output formats

- Choose JSON or XML output format
- Both produce the same clickable view but the “download” link gives access to the raw API response
- XML is the GATE standoff XML format - for examples see the documents you used in module 1
- JSON is based on Twitter data format

GATE General Architecture for Text Engineering

JSON example

- “entities” has a list of annotations of each type
- “indices”: `[start,end]` positions within the “text”, exactly* as in GATE Developer

```
{
  "text": "Ian works at The University of Sheffield",
  "entities": {
    "Organization": [
      {
        "indices": [17, 40],
        "orgType": "university",
        "rule": "GazOrganization",
        "ruleFinal": "OrgFinal"
      }
    ],
    "Person": [
      {
        "indices": [0, 3],
        "gender": "male",
        "kind": "firstName",
        "rule": "GazPersonFirst",
        "firstName": "Ian",
        "ruleFinal": "PersonFinal"
      }
    ]
  }
}
```

* not quite exactly, if text has emoji

GATE General Architecture for Text Engineering

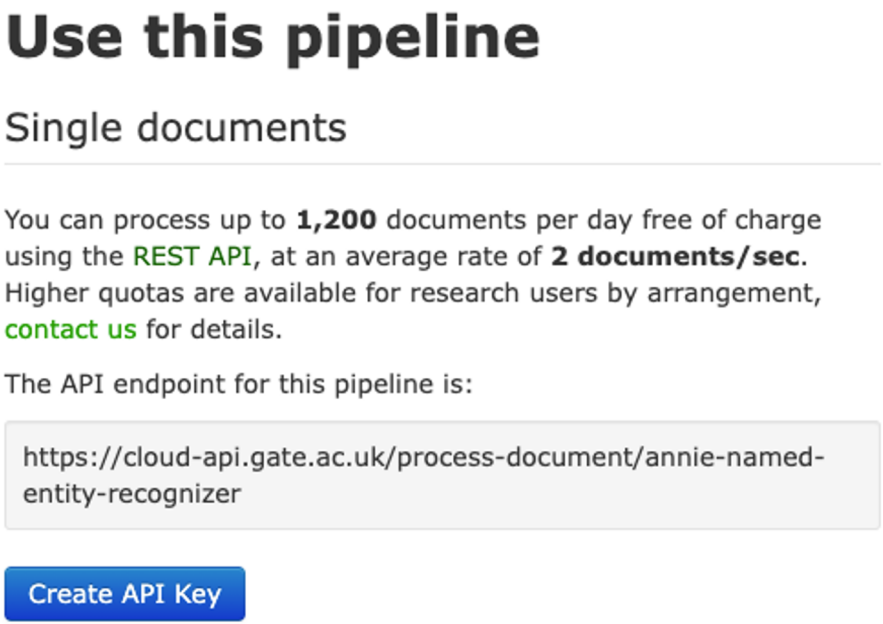
Using the API

- Try out interface is simply one front end on top of API
- You can call the same API directly from your own code, in any language with an HTTP client
- We have developed clients for
  - GATE Developer
  - Google Sheets

GATE General Architecture for Text Engineering

Using the API

- Each pipeline has its own endpoint URL
- HTTP POST to that URL with the document as body (with appropriate value for `Content-Type` header)
- JSON response by default, `Accept: application/xml` to get XML instead
- Full docs on cloud.gate.ac.uk



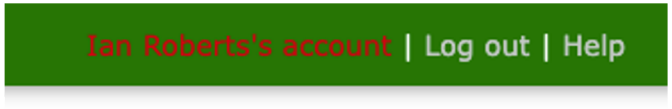
GATE General Architecture for Text Engineering

Authentication and rate limits

- Public pipelines work without authentication, but at very low daily quota and rate limit
  - 1 call per 10 seconds on average
  - ~100 calls per day
- The advertised 1,200 per day, 2 calls per second is the baseline for *registered* users
- To get this quota you must authenticate using an *API Key*

GATE Cloud API Keys

- If you haven't already, register now for an account on GATE Cloud
  - If you have already registered, log in
- Go to "<your name> account" in the top right
- Scroll to the bottom and "Manage your API keys"
- Generate a new key

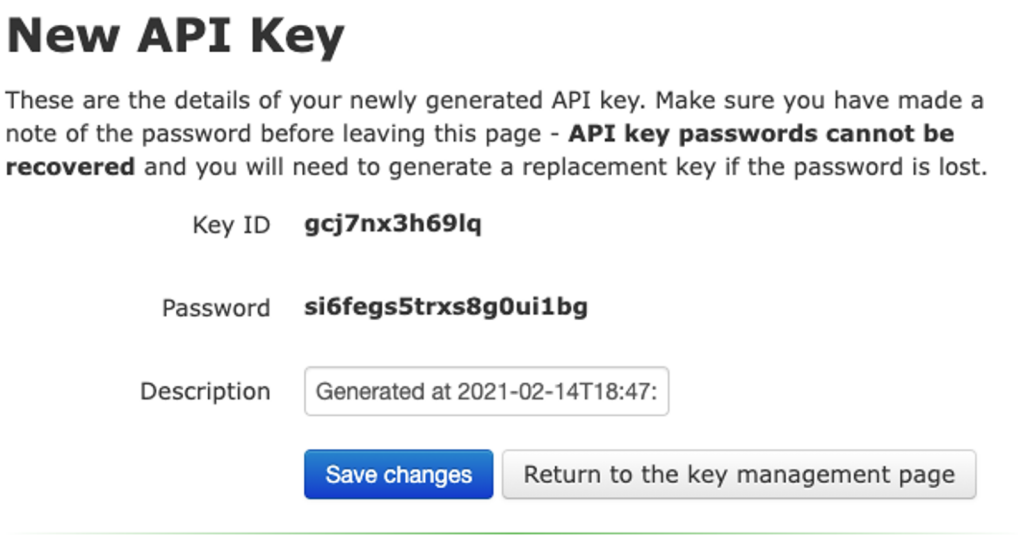


GATE General Architecture for Text Engineering

GATE Cloud API Keys

- API key is in two parts, the “key ID” and the “password”
- Copy and paste both into a text editor for future reference

- Note the warning: if you lose the password you must generate a new key (changing both ID and password)



GATE General Architecture for Text Engineering

Python example

- Open this notebook (Google Colab)
- Save a copy in your own Google Drive
- Example of how to call GATE Cloud with the `requests` module

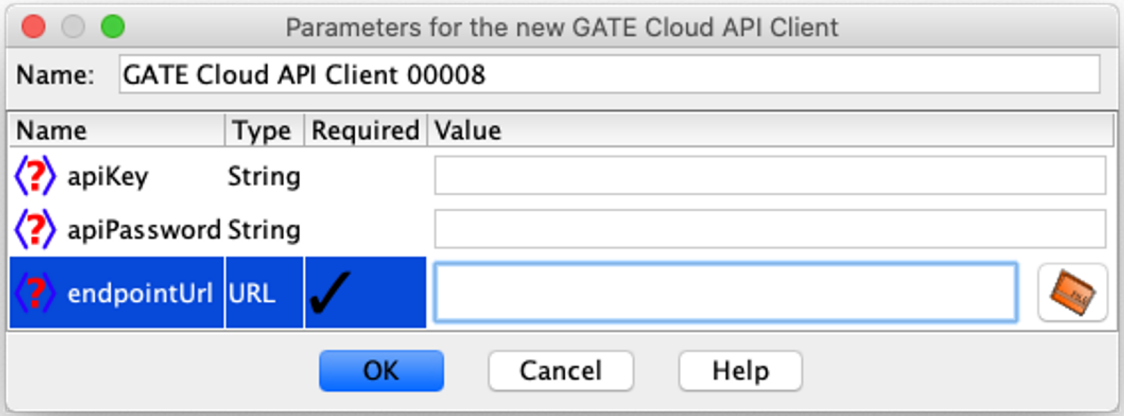
GATE General Architecture for Text Engineering

Calling GATE Cloud in Developer

- We provide a GATE PR to call a GATE Cloud pipeline
- Start GATE Developer
- Load the “GATE Cloud Client” plugin
- Create a new “GATE Cloud API Client” processing resource

Calling GATE Cloud in Developer

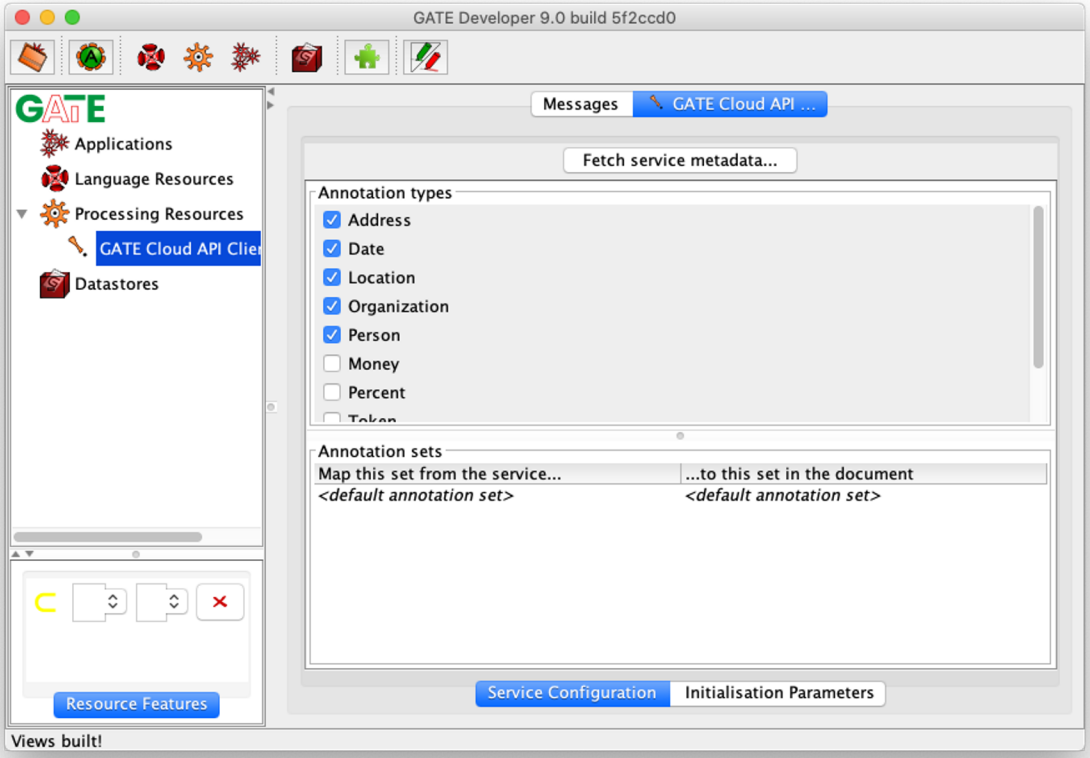
- Parameters
  - **apiKey** - the key ID you just created
  - **apiPassword** - the matching password
  - **endpointUrl** - the URL copied from "use this pipeline"



GATE General Architecture for Text Engineering

Configuring the PR

- Double-click the new PR in the tree, then “fetch service metadata”
- Annotation type checkboxes as on “test this pipeline”
- Annotation sets…



GATE General Architecture for Text Engineering

Annotation sets and types

- Recall: in GATE each document has one or more annotation sets, each containing annotations of one or more types
- Think back to those “annotation details”
- See the colon - “:”
- Annotations are specified to GATE Cloud as *selector* expressions “`SetName:AnnotationType`”
- Most pipelines use the default set (which has no name)

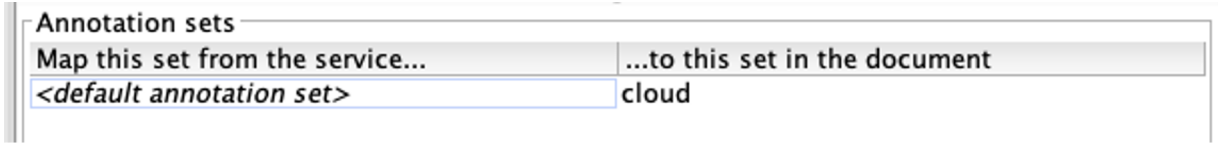
Default annotations

```
:Person
:Location
:Organization
```

GATE General Architecture for Text Engineering

Mapping annotation sets

- By default, cloud client PR puts annotations that come back from service into the same set the service used
- But this is configurable
  - e.g. call ANNIE, but put its output into a set called “cloud”



GATE General Architecture for Text Engineering

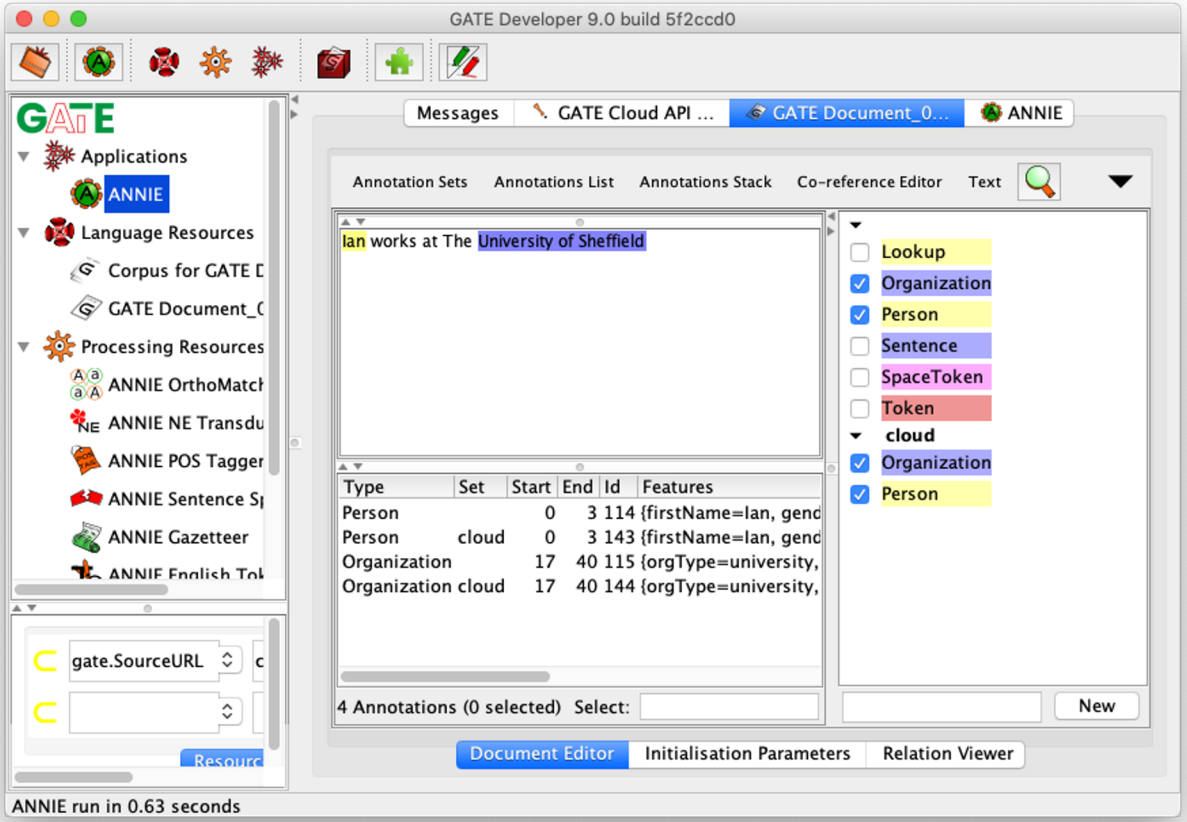
Try it

- Configure your GATE Cloud Client PR to map ANNIE default set to local “cloud” set
- Load a document
- Create a corpus (shortcut - you can right-click on the document, then “new corpus with this document”)
- Load the default ANNIE application
- Add the GATE Cloud Client PR to the end of the ANNIE pipeline
- Run the app over your (one document) corpus

GATE General Architecture for Text Engineering

Try it

Cloud ANNIE has found the same annotations as local ANNIE



GATE General Architecture for Text Engineering

Try a different service

- ANNIE is a good test, but not one you'd really use in this way
- There are other pipelines on GATE Cloud that are not available for you to run locally
- Delete your Cloud Client PR
- Create another one calling a different service
  - e.g. YODIE
- YODIE uses the "Shef" annotation set by default

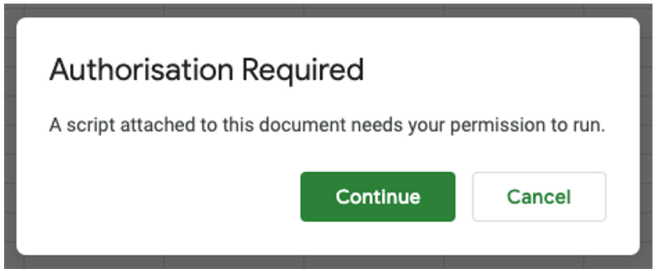
GATE General Architecture for Text Engineering

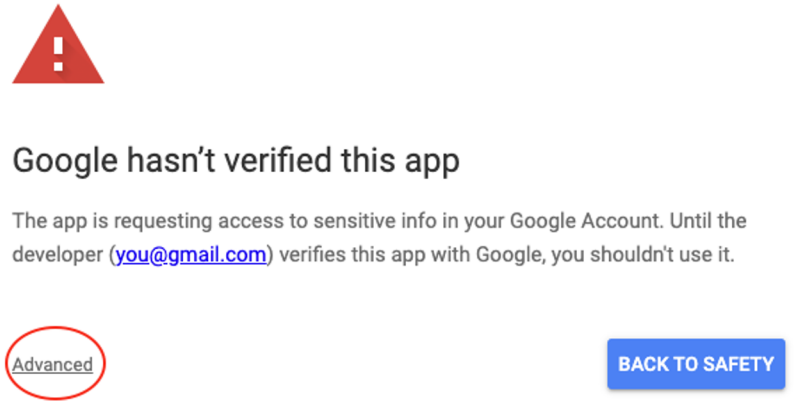
Google Sheets

- We have recently developed an add-on for Google Sheets to process text in a spreadsheet
- Particularly useful for people who are comfortable with spreadsheets but not programming
- Currently in beta - not yet a published add-on you can install to your own sheets
- For testing purposes take a copy of this Google Sheet
  - You can only *view* the original, not edit it
  - go to File → “Make a copy”

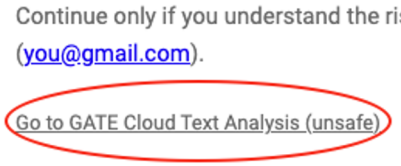
Authorisation

- In your copy of the sheet, “Add-ons” → “GATE Cloud Text Analysis” → “Open sidebar”
- First time you do this it will request authorisation
  - Your email will appear as the developer, this is normal!

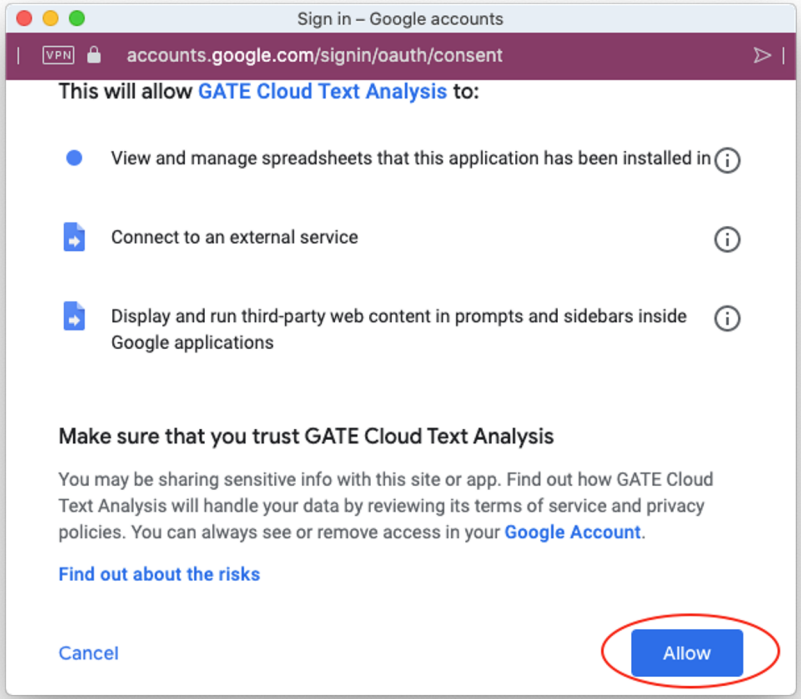




Authorisation



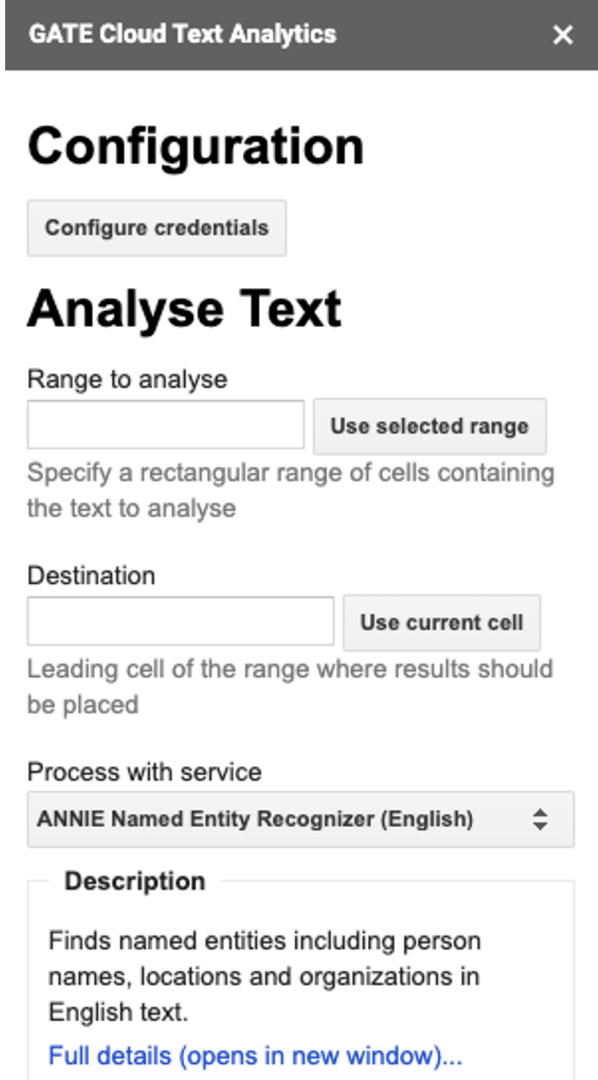
- When the add-on is published properly then we will be the developer and it won't be considered "unsafe"



GATE General Architecture for Text Engineering

Using the add-on

- Once authorised, “open sidebar” again
- “Configure credentials” lets you supply your API key ID and password
- Intended workflow
  - Choose a column of texts to process
  - Typically, set destination immediately to the right of the top-most input cell
    - For input B2:B9, destination would be C2
  - Choose the service to use
  - Configure annotations you want to extract using column headings



GATE General Architecture for Text Engineering

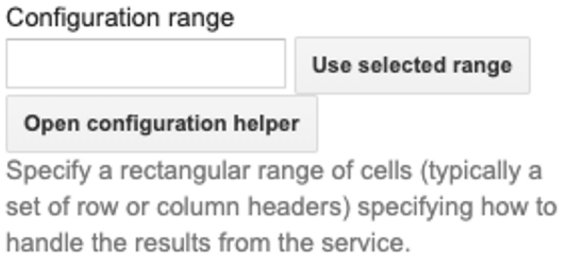
Demo

- The “BBC example” sheet has a few BBC news articles
- Select the cells with input text (B2 to B9)
- Under “Range to analyse” click “use selected range”
- We will put the output alongside - click into cell C2, then under destination click “use current cell”
  - Results from processing B2 will go in C2, D2, …
  - B3 will go in C3, D3, …
  - Etc.
- Select ANNIE as the service to use

GATE General Architecture for Text Engineering

Configuring annotations

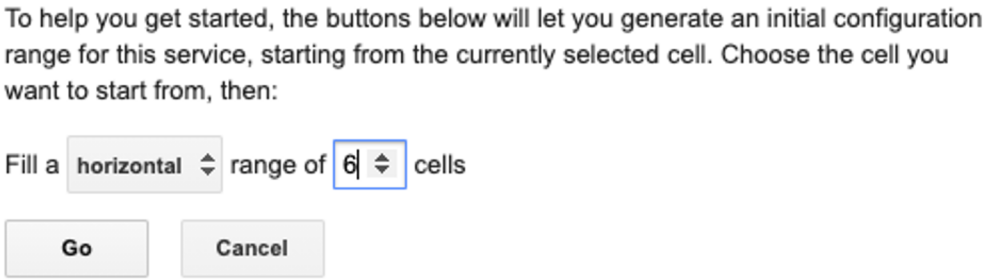
- One service can produce many different annotations
- Approach taken here is to map output into several columns
- “Configuration range” (typically column headers) defines what to extract
- Scroll sidebar down, click “Open configuration helper”



GATE General Architecture for Text Engineering

Configuring annotations

- Click into cell C1, fill a horizontal range of 6 cells



- You will see C1 to H1 filled with headings, and the range selected - click “Use selected range”
  - Column headings are annotation types

GATE General Architecture for Text Engineering

Explanation

- To summarise: you've now configured the tool to
  - process the text in B2:B9
  - using ANNIE
  - extract the annotations specified by C1:H1
  - and put the results in cells starting at C2
- "Submit job"
  - The tool will post each input cell in turn to GATE Cloud
  - If the input set were bigger, you'd see a percentage progress bar
  - Tool is careful to respect rate limit & quota, and will slow down processing or wait if necessary for quota to reset

GATE General Architecture for Text Engineering

More advanced configuration

- For a column heading like “Person”, tool outputs the text under each Person annotation in the API response
- But we know GATE annotations have other features
- Configuration is actually a pattern language that can extract any combination of features for each annotation

GATE General Architecture for Text Engineering

More advanced configuration

`AnnotationType pattern`

- `pattern` can be any sequence of feature names separated by spaces or punctuation

`Person gender`	→	male
`Location text (locType)`	→	Sheffield (city)

- `text` is a “magic” feature name representing the text under the annotation (like `@string` in JAPE)

GATE General Architecture for Text Engineering

More advanced configuration

- Configuration helper creates drop-down options on the headers but these are not exhaustive
- You can edit the headings to fit your requirements
- Use “test this pipeline” to see what features are available
  - “Full details” link under service description will take you there

GATE General Architecture for Text Engineering

Another example - YODIE

- Let's run a different pipeline on the same texts
- Range to analyse is still `B2:B9`
- Service: "YODIE named entity disambiguation (English)"
- Destination: use cell `I2`
- Configuration helper: click in `I1` and generate a one-cell range, then "use selected range"
- Drop down `I1` to see example configs
  - All the same `Mention` annotation type, but different combinations of features - `inst` is the DBPedia instance URI

GATE General Architecture for Text Engineering

Another example - YODIE

- Submit the job and let it process
- Note “Myanmar” and “Burma” both map to the same concept in DBPedia
  - It’s not perfect - “the WHO” is treated as the band, not the World Health Organization

GATE General Architecture for Text Engineering

Batch processing overview

- If you want to process large numbers of documents in a hurry
  - Upload documents as ZIP files
  - or collect data from Twitter (see module 4)
  - Reserve an “annotation job”, either for a standard pipeline, or upload your own
  - Processing done on Amazon EC2 with many VMs in parallel
  - Output as JSON or (ZIP files of) XML like the online API
  - Pay for the compute time you use - buy credit vouchers from the University of Sheffield online shop

GATE General Architecture for Text Engineering

Summary

- GATE Cloud offers a REST API to process text with any of our published pipelines
- API can be called by any client that speaks HTTP(S)
  - “Test this pipeline”
  - PR from GATE Developer
  - Your own code in Python/Java/golang/etc.
  - Google Sheets
- Everyone gets a basic free quota, enhanced quotas available for research users on request
- PAYG batch mode for larger jobs

GATE General Architecture for Text Engineering

Links

- Home page: https://cloud.gate.ac.uk
- API spec: https://cloud.gate.ac.uk/info/help/online-api.html
- Java library: https://github.com/GateNLP/cloud-client

Thank you!