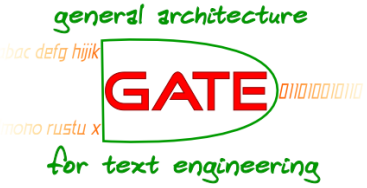
# Open-source Information Extraction with GATE for UN GA Resolutions

Adam Funk* Kalina Bontcheva Mark A. Greenwood
Ian Roberts
University of Sheffield

26th April 2019


The University Of Sheffield.






* a.funk@sheffield.ac.uk

# Contents

# 1. Summary

Our submission to the UNGA Resolutions challenge consists of the following items:

- a GATE application (pipeline for natural language processing and information extraction) developed for this project;
- a configuration file for Mímir semantic indexing and a sample configuration file for running a GCP batch job to process and index the documents;
- a Mímir information retrieval index with a "power user" search interface for semantic and textual queries on 2883 documents processed with that pipeline;
- a more user-friendly custom web front end for exploring sentence-level co-occurrences of keywords and semantic annotations in the Mímir index.

The software artifacts developed specifically for this project are licensed under the GPL-3.0 and available on GitHub. The GATE Developer, GCP, and Mímir tools used in the project are licensed under the LGPL-3.0 and are also available on GitHub as well as from the GATE website, which also contains the documentation.

GATE (General Architecture for Text Engineering) is a mature software toolkit for natural language processing (NLP), information extraction (IE), and related tasks. It includes GATE Developer, a GUI for developing applications, as well as libraries that allow those applications to be used headlessly and embedded in services.

# 2. Demonstrations

## 2.1. Exploring keyword and semantic annotation co-occurrences

A prototype for exploring sentence-level co-occurrences of keywords and semantic annotations is available here:

`http://demos.gate.ac.uk/unga/search`

This interface is a Grails web application that splits the lists of space-separated keywords and uses the search selection (preamble paragraphs only, operative paragraphs only, or whole documents) to "explode" a Mímir query for sentence collocation with the possible pairs drawn from the two lists (the Cartesian product), then compiles the results of all the queries and displays a matrix of the resulting counts. Multi-word keyphrases can be enclosed in double quotes to keep them from being split up.

For example, filling in the form with the lists `Cuba "United States"` and `embargo $Person $Date` and selecting *operative paragraphs only*, as shown in Figure 1, generates and runs the six Mímir queries shown in Figure 2. The numbers of sentences found are presented in a matrix with a heat map, as shown in Figure 3. Blank cells in the table indicate zero matches; any non-zero cell can be clicked and produces a list of sentences for that cell's query, grouped by document. The document filenames are links to a copy of the PDF on the server.

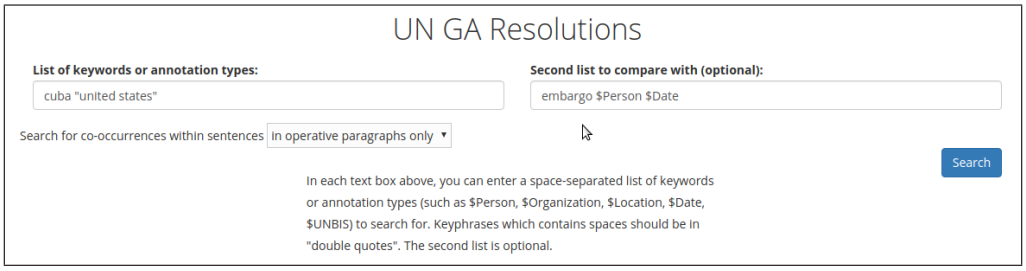



Figure 1: Example of a search

```
{Sentence} OVER ("Cuba" AND "embargo") IN {Operative}
{Sentence} OVER ("United States" AND "embargo") IN {Operative}
{Sentence} OVER ("Cuba" AND {Person}) IN {Operative}
{Sentence} OVER ("United States" AND {Person}) IN {Operative}
{Sentence} OVER ("Cuba" AND {Date}) IN {Operative}
{Sentence} OVER ("United States" AND {Date}) IN {Operative}
```

Figure 2: Mímir queries used for a search

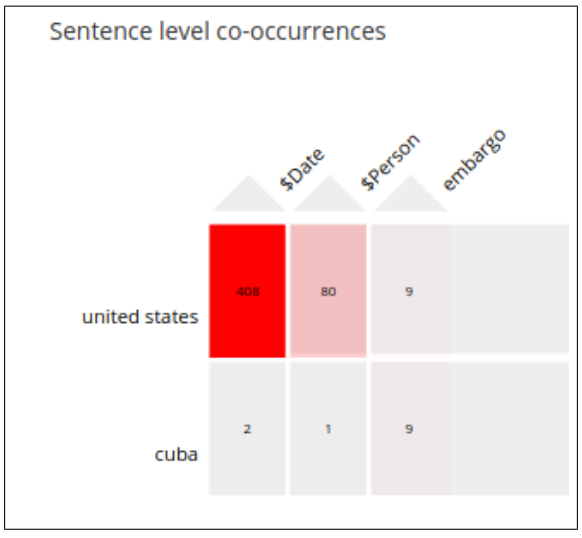



Figure 3: Example of a co-occurrence matrix

## 2.2. Information retrieval with keywords and semantic annotations

The Mímir index is available for direct searches here:

`https://demos.gate.ac.uk/unga/mimir/demo/search/index`

This interface is intended mainly for power users. The language allows queries over combinations of strings and annotations (as well as their features), using the operators `AND`, `OR`, `IN`, `OVER`, and `MINUS`, as illustrated in the following examples.

- `embargo AND Cuba`
  Find documents that contain both words.
- `{Sentence} OVER (embargo AND Cuba)`
  Find sentences that contain both words.
- `Cuba IN {Preamble}`
  Find occurrences of the word in a preamble.
- `{Sentence} OVER ({Person} AND {UNBIS})`
  Find sentences that contain a person's name (or co-referenced pronoun) and a term from the UNBIS thesaurus.
- `{Operative} OVER ({Person gender="male"} AND {Organization})`
  Find operative sections that contain a person's name identified as male and an organization name.
- `{Preamble} OVER ({Person} AND {Organization} AND {UNBIS})`
  Find preamble sections that contain a personal name, an organization name, and a term from the thesaurus.
- `{Preamble} OVER (({Person} OR {Organization}) AND {UNBIS})`
  Find preamble sections that contain a personal name or an organization name (or both) as well as a term from the thesaurus.
- `{Preamble} OVER ({UNBIS} MINUS {Organization})`
  Find preamble sections that contain a term from the thesaurus but no organization names.
- `{UNBIS uri="http://metadata.un.org/thesaurus#1007055"}`
  or
  `{UNBIS label="WOOD FUELS"}` (using the prefLabel)
  Find matches for any of the prefLabel or altLabel values for this thesaurus entry; this will find case-insensitive matches for *wood fuels*, *firewood*, *fuel wood*, *fuelwood*, or *wood energy* in the documents, but the label given in the query must be in upper case (to match the value in the thesaurus).
- `{Preamble} OVER {UNBIS label="WOOD FUELS"}`
  Find preamble sections that contain such a match.

The Mímir query language has additional features and is explained in more detail in §5.1 of the Mímir Guide[1]; the GATE website[2] also has examples, other demos, and a video.

Mímir also provides a web service API, which can be used by custom software such as the exploration interface described in the previous section (it converts the lists of keywords and annotation types and the search selection pulldown menu to Mímir queries, runs them through the web service, and formats the results).

# 3. GATE application

GATE includes many processing resources (PRs) that can be used as they are, such as the components of the ANNIE information extraction pipeline; most PRs can also be customized, in particular *gazetteers* (lists of keywords and phrases to match in the text, with options to control case-sensitivity, whole or partial word matching, etc.) and *JAPE* (Java Annotation Patterns Engine), a language for matching regular expressions over annotations in the GATE document and adding new annotations and features.

The GATE document object model separates the plain text content and turns all the document markup into stand-off annotations, each of which is associated with a span of plain text (specified by character offsets). Each annotation has a type and a feature map of arbitrary key-value pairs (usually, but not necessarily, strings). This document model is based on the TIPSTER [2] model and the annotations form an *annotation graph* [1]. The document begins with one annotation set representing the original markup; GATE PRs can create new annotation sets and read, add, remove, and modify annotations.

The GATE application developed for this project processes each document with the following sets of components (grouped by functions, not individually in the exact order in which they run).

- GATE (using the Tika library) converts the PDF file to a GATE document.
- Standard NLP components carry out tokenization, sentence-splitting, part-of-speech (POS) tagging, and lemmatization. This set of PRs illustrates the flexibility of the GATE annotation system: the tokenizer and sentence-splitter add *Token*, *SpaceToken*, and *Sentence* annotations to the document; then the POS tagger adds *category* features to the *Token* annotations, and the lemmatizer subsequently adds *root* features, making use of the POS tags.
- The ANNIE information extraction components perform named entity recognition (for Person, Organization, Location, and Date entity types) and orthographic co-referencing. Co-referencing matches similar named entities, such as *Mr Smith* and *John Smith*.

[1] `https://gate.ac.uk/mimir/doc/mimir-guide.pdf`
[2] `https://gate.ac.uk/mimir/`

- A specialized paragraph-marking tool detects paragraph boundaries and creates annotations for subsequent PRs to use. (This is tuned to the specific format of the documents used in this prototype, but could be adapted for others.)
- A date normalizer adds a *normalized* feature in the format `31-Oct-2018` to each *Date* annotation.
- A custom gazetteer and set of JAPE rules mark terms in the UNBIS thesaurus, page headers and footers, and identify sections of preamble paragraphs and operative paragraphs, making use of the text structure. These match occurrences in the documents of the `skos:prefLabel` and `skos:altLabel` values with the *en* language code.

The result is an annotation set containing the following output annotation types and features to be indexed in Mímir.

- *Person* with *gender* (if identifiable from the forename or title)
- *Organization* with *orgType*
- *Location* with *locType*
- *Date* with *normalized* value
- *Title* with *title* and *number* of the resolution
- *ResolutionAdoption* with normalized *date*
- *Preamble* annotations over the preamble paragraphs on each page
- *Operative* annotations over the operative paragraphs on each page
- *UNBIS* annotations with *uri* and *label* (prefLabel) features for matches to prefLabel and altLabel values in the UNBIS thesaurus

## 4. Software availability

The GATE application and the exploration interface are available from GitHub[3]. The GATE application can also be loaded directly in GATE by using the plugin manager with the following registration details.

| | |
|---|---|
| Group | `uk.ac.gate.plugins` |
| Artifact | `unga` |
| Version | `1.0-SNAPSHOT` |

[3] `https://github.com/GateNLP/gateplugin-UNGA`
`https://github.com/GateNLP/UNGA-search`

This will add the UN GA application to the *Ready Made Applications* menu. More details are available in Chapter 3 of the GATE User Guide. [3]

GATE Developer is licensed under the LGPL-3.0 and available on GitHub[4] and from the GATE website[5], which also contains the GATE Developer documentation.

The GATE Cloud Parallelizer (GCP) is a tool for running GATE applications headlessly over documents from various sources (typically files on disk) and storing the output in a Mímir index (in this case) or elsewhere. It is available with documentation from GitHub[6] under the LGPL-3.0 licence.

GATE Mímir (Multi-paradigm Information Management Index and Repository) is a system for indexing documents with GATE annotations and then searching with combinations of text, annotations, and annotation features. It is available from GitHub[7] under the LGPL-3.0; documentation and examples are on the GATE website.

## 5. Prospects for further development

This prototype is based on the document structures typically observed in the PDFs of UN GA resolutions that we downloaded, but could be adapted for other formats. GATE supports various forms of conditional processing, so that the document type can be detected early in the application and used to control the behaviour of subsequent parts of the pipeline.

The gazetteer derived from the UNBIS thesaurus for this prototype contains only the alternative and preferred labels given in the RDF itself, but we have experience of using word embeddings and corpora of relevant documents to enrich ontology-based gazetteers [4] with keywords and phrases similar to those used in the ontology itself, as we have done in the KNOWMAK[8] project. Mímir can also support ontology-aware queries using SPARQL, as exemplified in our BBC News demo; the main Mímir demo page[9] includes the Mímir interface and example queries, as well as the more user-friendly *People in the News* page, which uses SPARQL Mímir queries behind the scenes.

For this prototype, we batch-processed a collection of documents on disk with output to a Mímir index for searching and exploration. We also have experience in embedding GATE applications in web services in various ways, so that documents can be fed to a service individually over HTTP, or the service can received a list of HTTP or file URLs of document to process.

The exploratory interface currently lists the matched sentences only, grouped by document, with links to copies of the PDFs on our server. Further application development and integration with the digital library would allow better rendering and the use of document titles instead of filenames. We are also planning improvements in the Mímir service

[4] `https://github.com/GateNLP/gate-core`
[5] `https://gate.ac.uk/`
[6] `https://github.com/GateNLP/gcp`
[7] `https://github.com/GateNLP/mimir`
[8] `https://www.knowmak.eu/`
[9] `http://demos.gate.ac.uk/mimir/`

API which would allow interfaces such as this one to highlight the interior matches (e.g., the exact sections of text that match `Cuba` or `$Person`) within the larger match.

GATE also has resources available for noun- and verb-phrase chunking (which might be useful for more sophisticated semantic analysis), pronominal co-referencing, processing other languages, and term extraction over a corpus. The GATE team has experience in developing custom applications for sentiment analysis, event detection, Semantic Web uses, and enrichment of documents for intelligent archiving and retrieval.

## References


[1] Steven Bird and Mark Liberman. A formal framework for linguistic annotation. *Speech communication*, 33(1):23–60, 2001.

[2] Hamish Cunningham, Kevin Humphreys, and Robert Gaizauskas. GATE—a TIPSTER-based general architecture for text engineering. In *Proceedings of the TIPSTER Text Program (Phase III) 6 Month Workshop*. Morgan Kaufmann, 1997.

[3] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, Adam Funk, Angus Roberts, Danica Damljanovic, Thomas Heitz, Mark A. Greenwood, Horacio Saggion, Johann Petrak, Yaoyong Li, Wim Peters, and Leon Derczynski. *Developing Language Processing Components with GATE Version 8 (a User Guide)*. University of Sheffield, April 2019. URL `http://gate.ac.uk/sale/tao/index.html`.

[4] D. Maynard, B. Lepori, and P. Laredo. Using ontologies to map between research and policy data: opportunities and challenges. In *Proceedings of 17th International Conference on Scientometrics and Informetrics (ISSI)*, Rome, September 2019.