

Information Extraction with GATE General Architecture for Text Engineering

**Some experiences developing a very small medical information
extraction prototype**

Johann Mitloehner
Vienna University of Economics, Austria
<http://mitloehner.net>

Some GATE Predefined Components

ANNIE - A Nearly-New Information Extraction

- tokenizer
- gazetteer
- sentence splitter
- part of speech tagger
- named entities transducer
- co-reference tagger

Other Components

Ontology viewer and editor

- accepts OWL ontologies in XML and N3

JAPE Java Annotation Patterns Engine

- regular expressions over annotations
- grammar files and rules, including Java code
- ontology-aware

GUI Interface

- Easy to construct processing pipeline with predefined components
- Easy to view input documents and results, including statistics
- Cumbersome to automate and include customized processing resources

Gazetteer Editor

File Options Tools Help

Messages medi onto p voting-example... ANNIE Gazetteer

airport.lst New List

List name	Major	Minor	Value
loc_key.lst	loc_key	post	Academy of Sciences
loc_prekey.lst	loc_key	pre	al-Jazeera
loc_prekey_lower.lst	loc_key	pre	Al-Jazeera
loc_relig.lst	location	relig	Al-Qaeda
ministry.lst	organization	governm	ANSA
months.lst	date	month	Association of South East Asian Nations
mountain.lst	location	region	Bank
new_cdg.lst	cdg		Bank of England
newspapers.lst	organization	newspap	British Chambers of Commerce
number_fold.lst	number_fold		Bureau of Hygiene and Tropical Medicine
numbers.lst	number		Campaign for Connecticut Families
ordinal.lst	date	ordinal	Catholic
org_base.lst	org_base		Catholics
org_key.lst	org_key		CBI
org_key_cap.lst	org_key	cap	C B I
org_pre.lst	org_pre		CEC
org_spur.lst	spur		C E C
organization.lst	organization		CEE
percent.lst	percent		C E E
person_ambig.lst	person_first	ambig	Center for Retailing Studies
person_ending.lst	person_ending		Central Elections Commission
person_female.lst	person_first	female	Centre for Economic and Policy Research
person_female_cap.lst	person_first	female	Church of England
person_full.lst	person_full		Church of Scotland
person_male.lst	person_first	male	Communist Party
person_male_cap.lst	person_first	male	COMMUNIST PARTY
person_relig.lst	person_full	relig	Confederation of British Industry
person_sci.lst	person_full	sci	CONSERVATIVE PARTY
person_spur.lst	spur		DATA CAPTURE AGENCY
phone_prefix.lst	phone_prefix		Democratic Communist Party
province.lst	location	province	

Filter: Case Ins. Regex Value

Gazetteer Editor Initialisation Parameters Gaze

Views built!

NE Transducer

The screenshot displays the ANNIE NE Transducer application window. The menu bar includes File, Options, Tools, and Help. The toolbar contains icons for file operations and editing. The main window is titled "ANNIE NE Transd..." and shows a project named "medi onto". The interface is divided into a left-hand tree view and a right-hand text area.

Tree View:

- main
 - first
 - firstname
 - name
 - name_post
 - date_pre
 - date
 - reldate
 - number
 - address
 - url_pre
 - url
 - email
 - identifier
 - jobtitle**
 - final
 - unknown
 - name_context
 - org_context
 - loc_context
 - clean

Text Area:

```
* Version 2, June 1991 (in the distribution as file licence.html,  
* and also available at http://gate.ac.uk/gate/licence.html).  
*  
* Diana Maynard, 10 Sep 2001  
*  
* $Id: jobtitle.jape 5921 2004-07-21 17:00:37Z akshay $  
*/  
  
Phase: jobtitle  
Input: Lookup Token  
Options: control = appelt  
  
Rule: jobtitle1  
(  
  {Lookup.majorType == jobtitle}  
  (  
    {Lookup.majorType == jobtitle}  
  )?  
)  
:jobtitle  
-->  
:jobtitle.JobTitle = {rule = "JobTitle1"}
```

At the bottom of the window, there are tabs for "Jape Viewer" and "Initialisation Parameters". A status bar at the very bottom reads "Views built!".

Ontology Editor

The screenshot displays the GATE Ontology Editor interface. The main window is titled "GATE" and contains a project tree on the left, a central "Classes & Instances" panel, and a right-hand "Resource Information" panel.

Project Tree (Left):

- GATE
 - Applications
 - ANNIE
 - p
 - Language Resources
 - Corpus for voting-example.xml_00064
 - voting-example.xml_00064
 - medi onto (selected)
 - steve3.txt_0005D
 - steve2.txt_0005C
 - steve1.txt_0005B
 - nigel3.txt_0005A
 - nigel2.txt_00059
 - nigel1.txt_00058
 - jane3.txt_00057
 - jane2.txt_00056
 - jane1.txt_00055

Classes & Instances Panel (Center):

Classes and Instances

- concept
 - body_part
 - left_eye
 - right_eye
 - condition
 - congested
 - irritated
 - disease
 - viral_disease
 - herpes_zoster (selected)
 - symptom

Resource Information Panel (Right):

- Resource Information
 - herpes_zoster herpes_zoster
 - URI http://gate.ac.uk/owlim#herpes_zoster
 - TYPE Ontology Instance
 - Direct Types
 - viral_disease viral_disease
 - All Types
 - concept concept
 - disease disease
 - viral_disease viral_disease
 - Same Instances
 - Property Types
 - seeAlso [ALL RESOURCES]
 - versionInfo [ALL RESOURCES]
 - comment [ALL RESOURCES]
 - label [ALL RESOURCES]
 - isDefinedBy [ALL RESOURCES]
 - Property Values
 - label herpes zoster

Processing Pipeline

The screenshot displays the GATE software interface. On the left, a tree view shows the project structure under 'GATE', including 'Applications' (ANNIE, p) and 'Language Resources' (Corpus for voting-example.xml_0006, voting-example.xml_00064, medi onto, and several text files like steve3.txt_0005D, jane1.txt_00055, etc.). The main window shows the 'Messages' tab with 'medi onto' and 'voting-example...' selected. It features two tables for processing resources:

Loaded Processing resources	
Name	Corpus
ANNIE	Corpus I
ANNIE OrthoMatcher	ANNIE O
GATE Morphological analyser_0018C	GATE Mc

Selected Processing resources	
Name	Value
Document Reset PR	Document Reset P
ANNIE English Tokeniser	ANNIE English Tok
ANNIE Sentence Splitter	ANNIE Sentence Sp
ANNIE POS Tagger	ANNIE POS Tagger
ANNIE Gazetteer	ANNIE Gazetteer
ANNIE NE Transducer	ANNIE NE Transdu

Below these tables, the 'Corpus' is set to 'Corpus for voting-example.xml_00064'. A message area shows 'No processing resource selected...'. At the bottom, there is a 'Run this Application' button and tabs for 'Serial Application Editor' and 'Initialisation Parameters'.

Document view

The screenshot displays the GATE (General Architecture for Text Engineering) software interface. The main window shows a document titled "voting-example.xml_00064" with the following text:

BBC News - Voters head to polls for UK general election

General election voting under way

Millions of people in constituencies across the UK are casting their votes in the general election.

Polling stations up and down the country opened at 0700 BST and people will be able to cast their ballots until 2200 BST.

More than 44 million people are registered to vote. The first declarations are expected at 2300 BST.

As well as picking MPs for Westminster, voters will elect councillors in 164 local authorities across England.

Voting in the general election will take place in 649 constituencies, with nearly 4,150 candidates standing for election across the country.

David Cameron was the first of the main UK party leaders to cast their vote. The Tory leader went to a community hall in Witney, Oxfordshire, shortly after 1030 BST, accompanied by his wife Samantha.

Labour leader Gordon Brown went to vote shortly after 1100 BST at a community centre close to his home in North Queensferry, Fife. His wife Sarah was with him.

The interface includes a menu bar (File, Options, Tools, Help), a toolbar, and a sidebar with a project tree. The project tree shows a hierarchy: Applications > ANNIE > p > Language Resources > Corpus for voting-example.xml_00064 > voting-example.xml_00064. Below the project tree, there are fields for MimeType (text/html) and gate.SourceURL (http://newsvote.bbc.co.uk/). The main document area has tabs for Annotation Sets, Annotations List, Annotations Stack, Co-reference Editor, OAT, RAT-C, RAT-I, and Text. The Annotations List tab is active, showing a table of annotations:

Type	Set	Start	End	Id	Rule
Organization		0	8	6035	{rule1=TheOrgXKey, rule2=OrgFinal}
Location		36	38	6036	{locType=country, rule1=Location1, rule2=LocFinal}
Location		140	142	6037	{locType=country, rule1=Location1, rule2=LocFinal}
Location		445	456	6041	{locType=city, rule1=Location1, rule2=LocFinal}
Location		520	527	6042	{locType=province, rule1=Location1, rule2=LocFinal}
Location		712	714	6044	{locType=country, rule1=Location1, rule2=LocFinal}
Organization		753	757	6125	{rule = OrgJobTitle}
Location		793	799	6130	{rule=UnknownLocRegion}

At the bottom of the window, there are tabs for Document Editor and Initialisation Parameters, and a "New" button.

GATE Embedded

Use GATE libs in Java applications

- easy, just set CLASSPATH
- BUT lib component names do not correspond to GUI names
- > Best to proceed from sample code

```
00071905@CSSE2702:~/medical:543> cat makefile
GATE=$(HOME)/GATE-6.1
```

```
Pipe.class: Pipe.java
```

```
javac -classpath $(GATE)/bin/gate.jar:$(GATE)/lib/* Pipe.java
```

```
import gate.*;
import gate.util.*;
import gate.creole.*;
import gate.creole.ontology.*;
import gate.creole.gazetteer.*;
import java.io.*;
import java.util.*;
import java.net.*;
```

```
public class Pipe {
    private gate.Corpus corpus;
    public static void main(String[] args) throws Exception {
        new Pipe(args);
    }
```

```
Pipe(String[] files) throws Exception {
    Gate.init();
    Gate.getCreoleRegister().registerDirectories(
        new File(System.getProperty("user.dir")).toURL());
    System.out.println("\n== OBTAINING DOCUMENTS ==");
    createCorpus(files);
    System.out.println("\n== USING GATE TO PROCESS THE DOCUMENTS ==");
    runProcessingResources();
    System.out.println("\n== DOCUMENT FEATURES ==");
    display();
}
```

```
// Create corpus from documents on command line
```

```
private void createCorpus(String[] files) throws GateException {  
    corpus = Factory.newCorpus("Transient Gate Corpus");  
  
    for (int file = 0; file < files.length; file++) {  
        System.out.print("\t " + (file + 1) + " " + files[file]);  
        try {  
            corpus.add(Factory.newDocument(new File(files[file]).toURL()));  
            System.out.println(" -- success");  
        } catch(gate.creole.ResourceInstantiationException e) {  
            System.out.println(" -- failed (" + e.getMessage() + ")");  
        } catch(Exception e) {  
            System.out.println(" -- " + e.getMessage());  
        }  
    }  
}
```

```
// construct the processing pipeline and run it on the corpus

private void runProcessingResources() throws Exception {
    SerialAnalyserController pipeline = (SerialAnalyserController)Factory
        .createResource("gate.creole.SerialAnalyserController");

    // add processing resources to the pipeline: start with tokenizer

    ProcessingResource tokeniser = (ProcessingResource)
        Factory.createResource("gate.creole.tokeniser.DefaultTokeniser",
            Factory.newFeatureMap());

    pipeline.add(tokeniser);
}
```

```
// gazetteer needs parameters, have to construct feature map first

FeatureMap feats = Factory.newFeatureMap();
FeatureMap params = Factory.newFeatureMap();
URL mappingURL = new URL("file:" + System.getProperty("user.home") +
    "/medical/gaz/mappings.def");
URL listsURL = new URL("file:" + System.getProperty("user.home") +
    "/medical/gaz/lists.def");
params.put("mappingURL", mappingURL);
params.put("listsURL", listsURL);
params.put("gazetteerName", "com.ontotext.gate.gazetteer.HashGazetteer");

OntoGazetteer ontoGazetteer = (OntoGazetteer) Factory.createResource(
    "gate.creole.gazetteer.OntoGazetteerImpl", params, feats, "OntoGazetteer");

pipeline.add(ontoGazetteer);
```

```
// ontology and jape transducer
```

```
FeatureMap fm = Factory.newFeatureMap();  
fm.put("n3URL", new URL("file:" + System.getProperty("user.home") +  
    "/medical/medi.n3"));  
fm.put("defaultNameSpace", "");  
Ontology ontology = (Ontology) Factory.createResource(  
    "gate.creole.ontology.owlim.OWLIMOntologyLR", fm, null, "medi");
```

```
fm = Factory.newFeatureMap();  
fm.put("encoding", "ISO-8859-1");  
fm.put("grammarURL", new URL("file:" + System.getProperty("user.home") +  
    "/medical/medi.jape"));  
fm.put("ontology", ontology);  
ProcessingResource tr = (ProcessingResource)  
    Factory.createResource("gate.creole.Transducer", fm);
```

```
pipeline.add(tr);
```

```
// all done lets rock
```

```
pipeline.setCorpus(corpus);  
pipeline.execute();
```

```
}
```

```
// display specific annotations only
```

```
private void display() throws Exception {  
    Iterator docs = corpus.iterator();  
    while (docs.hasNext()) {  
        Document doc = (Document) docs.next();  
        feature("Symptom", doc);  
        feature("Disease", doc);  
    }  
}
```

```
// need to use document and offsets to get the text inside the annotation
```

```
void feature(String feat, Document doc) throws Exception {  
    Iterator it = doc.getAnnotations().get(feat).iterator();  
    while (it.hasNext()) {  
        System.out.println(feat + ": " + doc.getSourceUrl());  
        Annotation a = (Annotation) it.next();  
        System.out.println(doc.getContent().getContent(a.getStartNode().getOffset(),  
a.getEndNode().getOffset()));  
        System.out.println();  
    }  
}
```



```
00071905@CSSE2702:~/medical:530> cat gaz/lists.def
```

```
bodyparts.lst:bodypart:precise  
conditions.lst:condition:imprecise  
viral_diseases.lst:viral_disease:unreliable
```

```
# col 2 is majorType, col 3 is minorType (feature names for annotation)
```

```
00071905@CSSE2702:~/medical:531> cat gaz/bodyparts.lst
```

```
eyes  
limbs  
left eye  
right eye
```

```
# --> "left eye" gets a "Lookup" annotation with feature majorType=bodypart
```

```
00071905@CSSE2702:~/medical:532> cat gaz/mappings.def
```

```
bodyparts.lst::bodypart  
conditions.lst::condition  
viral_diseases.lst::viral_disease
```

```
# mappings connect gazetteer list names to ontology class names  
# note that there is no "diseases" list
```

00071905@CSSE2702:~/medical:535> cat medi.n3

@prefix : <http://gate.ac.uk/owlim#> .

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

@prefix owl: <http://www.w3.org/2002/07/owl#> .

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

@prefix protons: <http://proton.semanticweb.org/2005/04/protons#> .

@prefix protont: <http://proton.semanticweb.org/2005/04/protont#> .

:concept a owl:Class .

:symptom a owl:Class .

:body_part a owl:Class .

:condition a owl:Class .

:disease a owl:Class .

:condition rdfs:subClassOf :concept .

:symptom rdfs:subClassOf :concept .

:disease rdfs:subClassOf :concept .

:body_part rdfs:subClassOf :concept .

:viral_disease a owl:Class .

:viral_disease rdfs:subClassOf :disease .

:left_eye a :body_part .

:right_eye a :body_part .

:herpes_zoster a :viral_disease .

:congested a :condition .

:irritated a :condition .

```
00071905@CSSE2702:~/medical:537> cat medi.jape
```

```
Phase: OntoMatching
```

```
Input: Token Lookup
```

```
Options: control = appelt
```

```
# annotate body parts affected by conditions as "Symptom"
```

```
Rule: Symptom
```

```
(
```

```
  ({Lookup.majorType == bodypart}):part
```

```
  ({Token.string != ""})[0,3]
```

```
  ({Lookup.majorType == condition}):cond
```

```
):partcond
```

```
-->
```

```
:partcond.Symptom = { majorType = medical, minorType = unreliable }
```

```
# annotate anything that belongs to the class "disease" according to ontology
```

```
Rule: Disease
```

```
  ({Lookup.class == disease}):dis
```

```
-->
```

```
:dis.Disease = { majorType = medical, minorType = speculative }
```

00071905@CSSE2702:~/medical:539> cat gcm1.txt

RE: #####
##, Kojonup WA 6395 Tel: #####
DOB: ##/##/##### Our Ref: 25886

Thank you for referring this lady who had a left sided herpes zoster involving the maxillary division of her trigeminal nerve.

This occurred last May, and she is really still recovering. At this point of time she still has post hepatic pain and has a partial seventh nerve palsy on the left side.

Her visual acuity uncorrected was right and left 6/7.5 slightly better in the left than the right.

The left eye was a little congested, and she had a slow blink rate with some degree of exposure keratitis present.

The intraocular pressures were normal, and dilated examination showed no fundus pathology. The optic discs were healthy.

I told Mrs ##### that whilst the virus had burnt out, she had residual symptoms which may go on for some time.

In the meantime I have pointed out the importance of preserving the cornea of the left eye and the need for frequent lubrication and artificial tears.

With kind regards,
Yours sincerely,

```
00071905@CSSE2702:~/medical:538> java Pipe *.txt
```

```
== OBTAINING DOCUMENTS ==
```

```
1) gcm1.txt -- success
```

```
2) gcm2.txt -- success
```

```
...
```

```
== USING GATE TO PROCESS THE DOCUMENTS ==
```

```
HashGazetteer is being initialized!
```

```
Loading OWLIMOntologyLR, doSetAutoLabel is true
```

```
ruleSet=owl-max, partialRdfs=false
```

```
== DOCUMENT FEATURES ==
```

```
Symptom: file:/home/res.labs/5/00071905/linux/medical/gcm1.txt
```

```
left eye was a little congested
```

```
Disease: file:/home/res.labs/5/00071905/linux/medical/gcm1.txt
```

```
herpes zoster
```

```
# note that "herpes zoster" is annotated as "viral_disease" by the gazetteer
```

```
# only via the ontology the class "disease" can be inferred
```

```
00071905@CSSE2702:~/medical:540> cat gaz/viral_diseases.lst
```

```
herpes zoster
```