What is GATE

- A framework to develop Natural Language Processing (NLP) applications
  - A model for document content and document annotations
  - An API that implements that model
  - A bunch of software components for text processing

2

What a GATE application does?

- The main purpose of a GATE application is to produce annotations over parts of a text document
  - Token
  - POS
  - Named Entity
  - …

3

How can I create a GATE application?

- A GATE application consists in a sequence of "processing resources" applied over the document(s) you want to annotate
- How can I get a processing resource?
  - A lot of processing resources are available on GATE
  - You can program your own processing resources in Java using the GATE API
  - Other mechanisms (see later)
    - Gazetteer
    - JAPE

4

GATE model for annotations

- Annotation ID
- Annotation name (type)
- Document fragment referred to by the annotation
  - Start Node
  - End Node
- Features
  - Pairs (attribute, value)

5

Example GATE application: ANNIE

ANNIE

- A Nearly-New Information Extraction System
- Produce a number of annotations over a text document

7

ANNIE Components (I)

- Document Reset
  - Removes previous annotations
- Tokeniser
  - Recognises Token annotations (numbers, words, punctuation, space tokens)
- Gazetteer
  - Contains lists of names of some type
    - Months, cities, countries, companies, …

8

ANNIE Components (II)

- Sentence Splitter
  - Recognises sentences in document
- POS Tagger
  - Adds POS feature to Tokens
  - Composed of:
    - Lexicon
    - Ruleset to disambiguate
      - Example: To have a rest, to rest
- NE transducer
  - Recognise several types of named entities
  - Based on JAPE
- Orthomatcher
  - Adds identity relations over named entities
  - Runs only over entities of the same type or unknown

9

Tools to develop processing resources

Gazetteer

- A Gazetteer is composed of a number of lists
- Each list is a plain text file with one entry per line
- An index file (lists.def) points to all the lists
- For each list
  - List file
  - Major type
  - Optionally: minor type, language
    - company_fr.lst:organization:company:french
- Longest match
  - Santiago
  - Santiago de Compostela

11

JAPE

- A JAPE grammar recognises regular expressions over annotations and perform actions over recognised patterns
- Composed of a number of rules
- Each rule: (pattern --> actions)

12

Sample JAPE rule

```
Rule: IPAddress
(
{Token.kind == number}
{Token.string == "."}
{Token.kind == number}
{Token.string == "."}
{Token.kind == number}
{Token.string == "."}
{Token.kind == number}
)
:ipAddress -->
:ipAddress.Address = {kind = "ipAddress"}
```

13