Module 6: ANNIC
The art and craft of JAPE rules

• You know by now how to write some simple JAPE rules

• The question is: how do you design them? How do you find patterns which are frequent in your test corpus?

• Given a dataset of tweets, how can you be sure that the JAPE LHS pattern you are about to implement doesn’t do more harm than good?
Motivation

- Need for a corpus analysis tool
- Useful for authoring of IE patterns for rules

ANNIC is an IR engine that can search over:
- Document Content (words)
- Metadata (Annotation types, features and values)
  - for example: Person.gender==”male”
ANNIC

- is based on Apache Lucene technology.
- can index any document supported by GATE
- is integrated in GATE as a Searchable Datastore
- has an advanced GUI that provides:
  - view of annotation mark-ups over the matched patterns
  - interactive way of developing new patterns
  - annotation statistics
How does it work?

- Initialization
  - where to store
  - what to index and what to exclude
  - context boundary (e.g. restricted within sentence or paragraph boundaries)
- Index actions linked with Datastore actions
  - when document is saved, index or re-index if already indexed
  - when document is deleted, delete it from the index
Hands-on 1: creating a datastore

- In GATE, right click on Datastores, then Create Datastore

![Create a datastore](image)

- Specify a new empty directory for the index

- By default, the annotation sets to be indexed are the default set (null) and the Key set (where by convention we put gold-standard annotations)

- We want to index only the PreProcess annotation set

- This needs to be specified at index creation time – we cannot change it later
Create Lucene Datastore (2)

- Click on the pencil button opposite Annotation Sets
- In the list box, delete the default values, type PreProcess and press the Add button
- Uncheck “Create Tokens Automatically
- Leave all else with default values
- Click OK, the new datastore is now ready to use
ANNIC: The Query Language

- JAPE –like LHS Pattern syntax
  - String within quotes or without quotes
    - e.g. “ubuntu”
  - \{AnnotationType\}
    - e.g. \{Person\}
  - \{AnnotationType == string\}
    - e.g. \{Organization == “University of Sheffield”\}
  - \{AT.featureName==value\}
    - e.g. \{Person.gender == male\}
  - \{AT.feature==value, AT.feature==value\}
    - e.g. \{Token.orth == “upperInitial”, Token.length == “3”\}
• Klene Operator + and * but they need to be quantified
  • \{\text{Person}\}\{\text{Token}\}\ast3\{\text{Organization}\} – find all Person and Organization annotations \text{within up to 3 tokens} of each other
• Logical $|$ (OR) operator
  • \{A\}(\{B\} \mid \{C\})
• Order of query terms is very important
Hands-on: ANNIC Pattern Searches

- Create a corpus and save it to the newly created Lucene Datastore
- Populate the corpus from the `corpus/annic-documents` directory
- Note: if you populate the corpus before saving it, all documents are loaded in memory – this might be bad if you have a lot!
- Double click on the datastore
- Click on the “Lucene Datastore Searcher” tab at the bottom
- Choose over which annotation set you wish to search (top right). Enter a test ANNIC query (e.g. `{Lookup}` or `{Hashtag}`) in the big search field, then press Search
Example: Building a Date pattern

- Let us first start by checking the `{Lookup}` annotations in the PreProcess set and the context in which they appear.
Seeing More Context

• Click the Configure button

• In the dialog box, keep adding rows for the annotation types (and optionally features) that you’d like displayed in the viewer

• A good set for our example is this:
Building Up A Date Pattern

• Let’s look for dates which contain a day of the week

• We start the query by typing \{Lookup.minorType=="day"\}

• 22 results are returned. The subsequent word is typically a Lookup of type month

• Expand the query: \{Lookup.minorType=="day"\} \{Lookup.minorType=="month"\}

• This still returns 22 results, which means we haven’t lost anything or introduced noise

• From inspection, we notice that what follows next is a number. These can be recognised from Token.kind == “number”

• Final Date LHS pattern: \{Lookup.minorType=="day"\} \{Lookup.minorType=="month"\}\{Token.kind=="number"\}
Example Results
Hands-on: Expand to include the time

- Double-click on the datastore, open the ANNIC GUI
- In the ANNIC GUI:
  - Expand the pattern to include the time expressions
Converting the Pattern to a JAPE Rule

• You might wish to create several different annotations from this JAPE LHS, e.g. Date, Time, and Offset

• Use different named blocks in the pattern to achieve this

• We leave this as homework, especially if you wish to link the year (which appears at the end) with the rest of the date

• A relevant PR here is the DateNormalizer:
  – http://gate.ac.uk/userguide/sec:misc-creole:datenormalizer