



---

# BD003: Introduction to NLP

## Part 2 Information Extraction



# Contents

---

- This tutorial comprises the following topics:
- Introduction to Information Extraction
- ANNIE – GATE's IE tool
- Other tools for IE



# Named Entity Recognition: the cornerstone of IE

- Traditionally, NER is the identification of proper names in texts, and their classification into a set of predefined categories of interest
- Person
- Organisation (companies, government organisations, committees, etc)
- Location (cities, countries, rivers, etc)
- Date and time expressions

Various other types are frequently added, as appropriate to the application, e.g. newspapers, ships, monetary amounts, percentages.

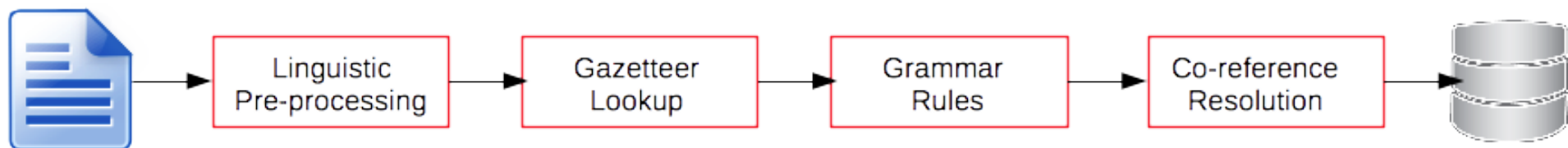


# Why is NE important?

- NE provides a foundation from which to build more complex IE systems:
  - Relations between NEs can provide tracking, ontological information and scenario building
  - Tracking (co-reference): “Dr Smith”, “John Smith”, “John”, “he”
  - Ontologies: “Athens, Georgia” vs “Athens, Greece”
  - Opinion mining: find what the opinions are about

# Typical NE pipeline

- Pre-processing (tokenisation, sentence splitting, morphological analysis, POS tagging)
- Entity finding (gazetteer lookup, NE grammars)
- Co-reference (alias finding, orthographic co-reference etc.)
- Export to database / XML / ontology





# Example of IE

---

John lives in London . He works there for Polar Bear Design .



# Basic NE Recognition

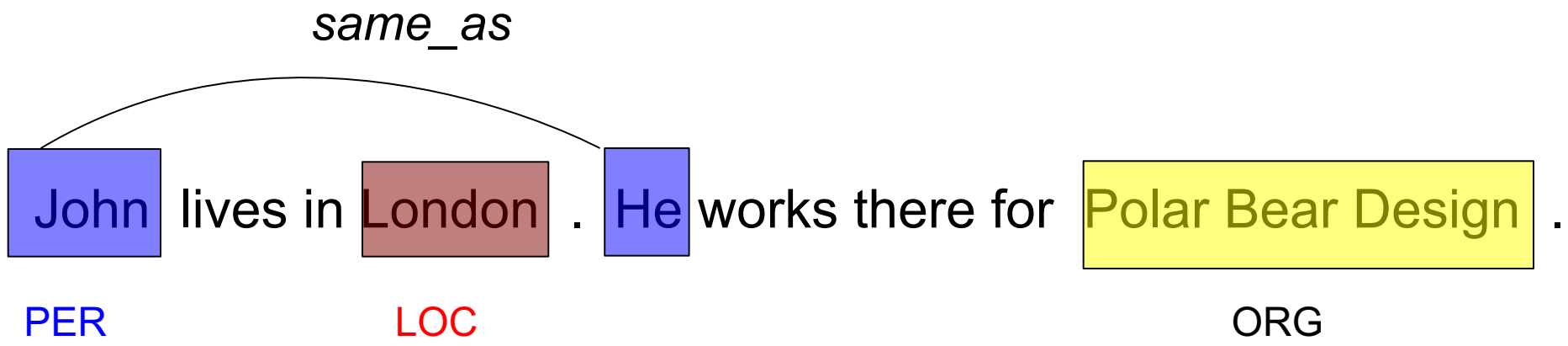
---

John lives in London . He works there for Polar Bear Design .

PER LOC ORG



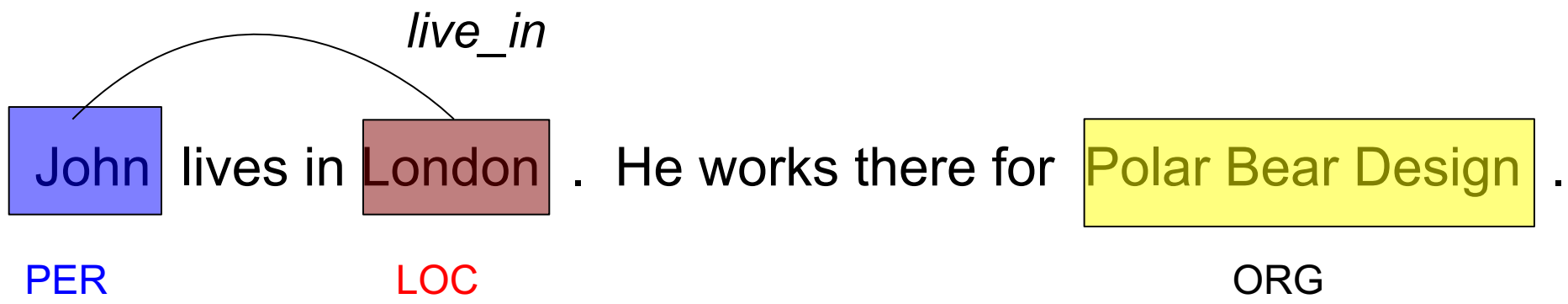
# Co-reference





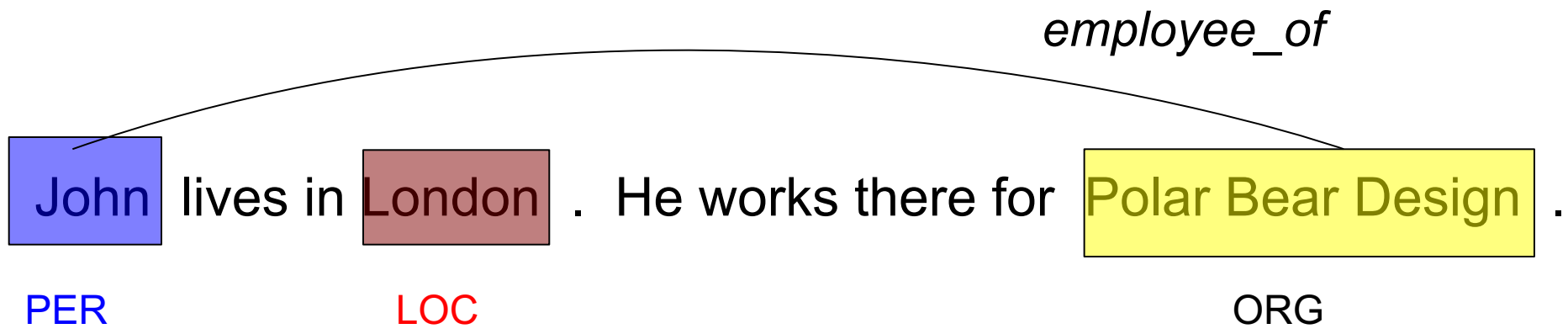


# Relations



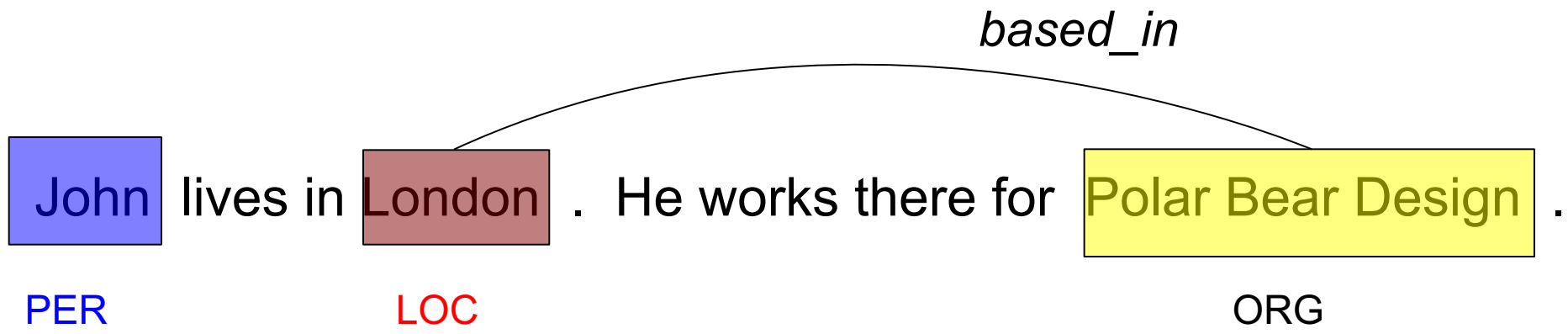


# Relations (2)





# Relations (3)





---

# **ANNIE: A Nearly New Information Extraction system**



# Nearly New Information Extraction

---

- ANNIE is a ready made collection of PRs that performs IE on unstructured text.
- ANNIE is “nearly new” because
  - It was based on an existing IE system, LaSIE
  - We rebuilt LaSIE because we decided that people are better than dogs at IE
  - Being 17 years old, it's not really new any more
  - The person who named it (not me) didn't really think it through (probably had too many beers...)



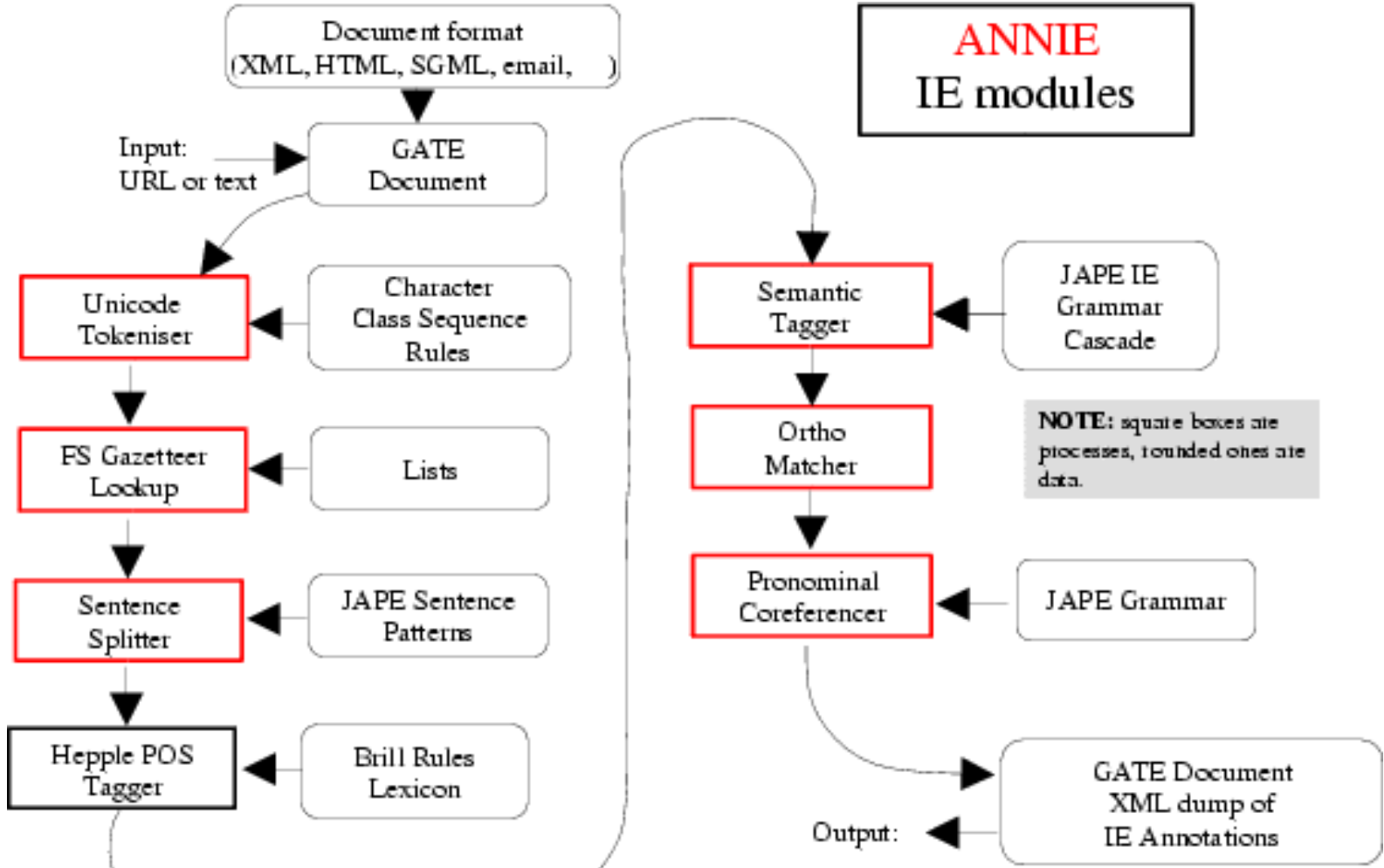
# What's in ANNIE?

---

- The ANNIE application contains a set of core PRs:
  - Tokeniser
  - Sentence Splitter
  - POS tagger
  - Gazetteers
  - Named entity tagger (JAPE transducer)
  - Orthomatcher (orthographic coreference)
- There are also other useful PRs, which are not used in the default application, but can be added if necessary (chunkers, parsers etc.)



# Core ANNIE components





# Loading and running ANNIE

---

- Let's look again at the documents we annotated earlier with ANNIE
- Clicking on the annotations (right hand pane) in the default set, you should see a mixture of Named Entity annotations (Person, Location etc) and some other linguistic annotations (Token, Sentence etc.)
- Let's see what each component in ANNIE does
- Which components generate which annotations?
- **Have a guess!**





# Let's look at the PRs

- Each PR in the ANNIE pipeline creates some new annotations, or modifies existing ones
  - Document Reset → removes annotations
  - Tokeniser → Token annotations
  - Gazetteer → Lookup annotations
  - Sentence Splitter → Sentence, Split annotations
  - POS tagger → adds category features to Token annotations
  - NE transducer → Date, Person, Location, Organisation, Money, Percent annotations
  - Orthomatcher → adds match features to NE annotations



# Document Reset

---

- This PR should go at the beginning of (almost) every application you create
- It removes annotations created previously, to prevent duplication if you run an application more than once
- It does not remove the Original Markups set, by default
- You can configure it to keep any other annotation sets you want, or to remove particular annotation types only



# Document Reset Parameters

Loaded Processing resources

| Name | Type |
|------|------|
|------|------|

Selected Processing resources

| ! | Name                    | Type  |
|---|-------------------------|-------|
|   | Document Reset PR_00016 | Docur |

Run "Document Reset PR\_00016"?

Yes  No  If value of feature  is

Corpus: <none>

Runtime Parameters for the "Document Reset PR\_00016" Document Reset PR:

| Name                  | Type      | Required | Value  |
|-----------------------|-----------|----------|--------|
| annotationTypes       | ArrayList |          | []     |
| keepOriginalMarkupsAS | Boolean   |          | true   |
| setsToKeep            | ArrayList |          | [Key ] |

Run this Application

Specify any specific annotations to remove. By default, remove all.

Keep Original Markups set

Keep Key set



---

# Tokenisation and sentence splitting



# Tokenisation

---

- Tokenisation chops text into tokens (usually words, numbers and symbols)
- Typically separated (in English) by white space, but not always
- Tokens usually have features denoting things like orthography, kind (word/number/symbol) and maybe also things like part-of-speech or normalised versions (e.g. with misspellings or abbreviations)
- It's generally quite an easy task, but different tools generate tokens differently
  - should **20-02-16** be a single token?
  - What about **it's**?



# ANNIE Tokeniser

- Tokenisation based on Unicode classes
- Declarative token specification language
- Produces Token and SpaceToken annotations with features **orthography** and **kind**
- **length** and **string** features are also produced
- Rule for a lowercase word with initial uppercase letter:

```
"UPPERCASE_LETTER" LOWERCASE_LETTER"* >  
  Token; orthography=upperInitial; kind=word
```



# Document with Tokens

Annotation Sets Annotations List Annotations Stack Class Co-reference Editor Instance Text

Union Appeals For Talks To End BA Strike

Skip to navigation | Skip to content |  
Home | Contact Us | News Search;  
HubPage  
Airwise News  
Airport Guide  
Airwise Travel  
Search  
Union Appeals For Talks To End BA Strike  
March 22, 2010

Union leaders on Sunday called for talks with British Airways bosses to end strike action by cabin crew that has led to the cancellation of hundreds of flights and disrupted travel plans for thousands of passengers.

| Type  | Features   |
|-------|--|
| Token | { category=NNP, kind=word, length=5, orth=upperInitial, string=Union}    |
| Token | { category=NNPS, kind=word, length=7, orth=upperInitial, string=Appeals} |
| Token | { category=IN, kind=word, length=3, orth=upperInitial, string=For}       |
| Token | { category=NNS, kind=word, length=5, orth=upperInitial, string=Talks}    |
| Token | { category=TO, kind=word, length=2, orth=upperInitial, string=To}        |

- Date
- FirstPerson
- JobTitle
- Location
- Lookup
- Money
- Organization
- Percent
- Person
- Sentence
- SpaceToken
- Split
- Title
- Token
- Unknown
- Original markups



# ANNIE English Tokeniser

---

- The English Tokeniser is a slightly enhanced version of the Unicode tokeniser
- It comprises an additional JAPE transducer which adapts the generic tokeniser output for the POS tagger requirements
- It converts constructs involving apostrophes into more sensible combinations
- don't → do + n't
- you've → you + 've





# Looking at Tokens

---

- Remove ANNIE (right click on the name of the application (ANNIE) in the left hand pane -> Close recursively)
- Create a new application (corpus pipeline)
- Load a Document Reset and an ANNIE English Tokeniser
- Add them (in that order) to the application and run on the corpus
- View the Token and SpaceToken annotations
- What different values of the “kind” feature do you see?



# Sentence Splitting

---

- Sentence splitting chops text up into sentences
- Does certain punctuation (such as full stops, exclamation marks and question marks) denote the end of a sentence or something else?
- What else might it denote?
- Usual way to resolve this is to use lists of known abbreviations etc. and heuristics for the rest
- But what about tabbed numbers (1.) and quotations?
- What about multi-line addresses?



# ANNIE Sentence Splitter

---

- The default splitter finds sentences based on Tokens
- Creates Sentence annotations and Split annotations on the sentence delimiters
- Uses a gazetteer of abbreviations etc. and a set of JAPE grammars which find sentence delimiters and then annotate sentences and splits
- Load an ANNIE Sentence Splitter PR and add it to your application (at the end)
- Run the application and view the results



# Document with Sentences

Annotation Sets Annotations List Annotations Stack Class Co-reference Editor Instance Text

the opposition conservatives, ahead in opinion polls, have been turning up the pressure on Labour over its links to Unite, saying the government had failed to take action quickly enough because it did not want to alienate its financial backers.

"We deplore the strike, and the prime minister and the transport secretary have said that absolutely clearly," Foreign Secretary David Miliband told Sky News.

"The way to resolve these disputes is through negotiation, it is damaging for the company, it is damaging for the crews and it is damaging for the country."

The dispute arose because BA, which has 12,000 cabin crew, wants to save an annual GBP£62.5 million pounds (USD\$95 million) to help cope with a fall in demand, volatile fuel prices and increased competition from low-cost carriers.

A spokesman said there was no estimate yet as to how much the industrial action would cost the company.

| Type        | Features |
|-------------|----------|
| Sentence {} |          |
| Sentence {} |          |
| Sentence {} |          |
| Sentence {} |          |
| Sentence {} |          |

- Date
- FirstPerson
- JobTitle
- Location
- Lookup
- Money
- Organization
- Percent
- Person
- Sentence
- SpaceToken
- Split
- Title
- Token
- Unknown
- ▶ Original markups



---

# Shallow lexico-syntactic features



# POS tagging

---

- Part-of-speech (POS) tagging is concerned with tagging words with their part of speech, e.g. noun, verb, adjective
- These basic linguistic categories are typically divided into quite fine-grained tags, distinguishing between e.g. singular and plural nouns, and different tenses of verbs.
- For languages other than English, gender may also be included in the tag.
- The set of possible tags used is critical and varies between different tools, making interoperability between different systems tricky.

# POS tagsets

- Different tools use different sets of tags, usually depending on how the tools were trained
- Some have more fine-grained sets of features than others
- One very commonly used tagset for English is the Penn Treebank (PTB) – used in ANNIE

---

|         |                                      |
|---------|--------------------------------------|
| Context | There were 250 cyber-crime offences. |
| Token   | EX VBD CD NN NNS                     |

- Other popular sets include those derived from the Brown corpus and the LOB (Lancaster-Oslo/Bergen) Corpus



# How POS tagging works

- The POS tag is determined by taking into account not just the word itself, but also the context in which it appears.
- This is because many words are ambiguous, and reference to a lexicon is insufficient to resolve this.
- **Love** could be a noun or verb depending on the context
  - **I love fish**
  - **Love is all you need**
- Approaches typically use machine learning, because it is quite difficult to describe all the rules needed for determining the correct tag given a context





# ANNIE POS tagger

- 
- ANNIE POS tagger is a Java implementation of Brill's transformation based tagger
  - Trained on Wall Street Journal, uses Penn Treebank tagset
  - Default ruleset and lexicon can be modified manually (with a little deciphering)
  - Adds **category** feature to **Token** annotations
  - Requires Tokeniser and Sentence Splitter to be run first



# Morphological analysis

---

- Morphological analysis involves the identification and classification of the linguistic units of a word
- Typically breaks the word down into its root form and an affix
  - e.g. **walked** = **walk** (root) + **-ed** (affix)
- For English, typically applied to verbs and nouns and involve suffixes, but other languages use affixes and infixes



# Inflectional and derivational morphology

---

- Inflectional variants involve mood, tense, plurals etc.
  - run -> ran
  - dog -> dogs
- Derivational variants involve a change of syntactic category (part of speech)
  - work -> worker
  - loud -> loudness
- Morphological analysers for English typically only deal with inflectional morphology, and are often rule-based



# GATE morphological analyser

---

- Not an integral part of ANNIE, but can be found in the Tools plugin as an “added extra”
- Flex based rules: can be modified by the user (instructions in the User Guide)
- Generates **root** feature on Token annotations
- Requires Tokeniser to be run first
- Requires POS tagger to be run first if the considerPOSTag parameter is set to true



# Running the Morphological Analyser

- 
- Add an ANNIE POS Tagger to your app
  - Add a GATE Morphological Analyser after the POS Tagger
  - If this PR is not available, load the Tools plugin first
  - Re-run your application
  - Examine the features of the Token annotations
  - New features of category and root have been added



---

# Gazetteers



# Gazetteers

- 
- Gazetteers are plain text files containing lists of names (e.g rivers, cities, people) used to help with many NLP tasks such as NER
  - Each gazetteer has an index file listing all the lists, plus features of each list (`majorType`, `minorType`, and `language`)
  - Lists can be modified either internally using the Gazetteer Editor, or externally in your favourite editor
  - Gazetteers generate by default `Lookup` annotations with relevant features corresponding to the list matched
  - Note that the name of the annotation produced for each list can be changed from `Lookup` (option in the index file)



# Running the ANNIE Gazetteer

---

- Various different kinds of gazetteer are available: we'll look at the default ANNIE gazetteer
- Add the ANNIE Gazetteer PR to the end of your pipeline
- Re-run the pipeline
- Look for “Lookup” annotations and examine their features





# ANNIE gazetteer - contents

---

- Double click on the ANNIE Gazetteer PR (under Processing Resources in the left hand pane) to open it
- Select “Gazetteer Editor” from the bottom tab
- In the left hand pane (linear definition) you see the index file containing all the lists
- In the right hand pane you see the contents of the list selected in the left hand pane
- Each entry can be edited by clicking in the box and typing
- New entries can be added by typing in the “New list” or “New entry” box respectively



# Gazetteer editor

The screenshot shows the GATE Developer 7.1-ANNIE Gazetteer Editor interface. The main window displays a list of definition files on the left and a table of entries for the selected 'city.lst' file on the right. The table has columns for 'List name', 'Major', 'Minor', and 'Value'. The 'city.lst' file is selected, and its entries are shown in the table. The table shows entries for 'city' and 'country' with their respective major and minor categories. The 'Value' column contains the names of the cities and countries.

| List name           | Major         | Minor          | Value     |
|---------------------|---------------|----------------|-----------|
| abbreviations.lst   | stop          |                | Aaccra    |
| adbc.lst            | adbc          |                | Aalborg   |
| airports.lst        | location      | airport        | Aarhus    |
| charities.lst       | organization  |                | Ababa     |
| city.lst            | location      | city           | Abadan    |
| city_cap.lst        | location      | city           | Abakan    |
| company.lst         | organization  | company        | Aberdeen  |
| company_cap.lst     | organization  | company        | Abha      |
| country.lst         | location      | country        | Abi Dhabi |
| country_abbrev.lst  | location      | country_abbrev | Abidjan   |
| country_adj.lst     | country_adj   |                | Abilene   |
| country_cap.lst     | location      | country        | Abu       |
| currency_prefix.lst | currency_unit | pre_amount     | Abu Dhabi |
| currency_unit.lst   | currency_unit | post_amount    | Abuja     |
| date_key.lst        | date_key      |                | Acapulco  |
| date_unit.lst       | date_unit     |                | Acarigua  |
| day.lst             | date          | day            | Accra     |

definition file entries

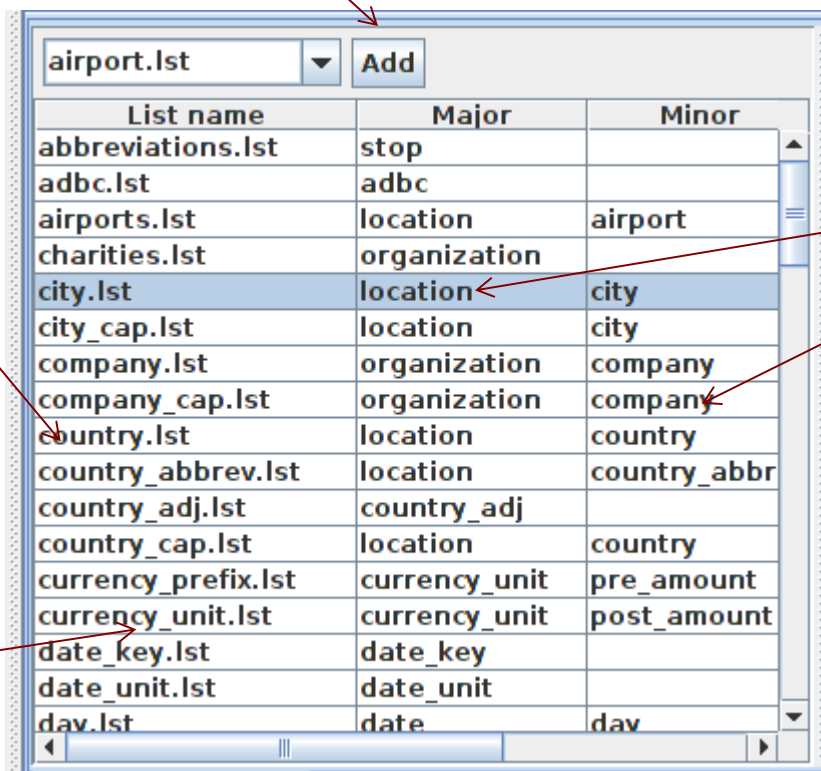
entries for selected list

# Modifying the definition file

add a new list

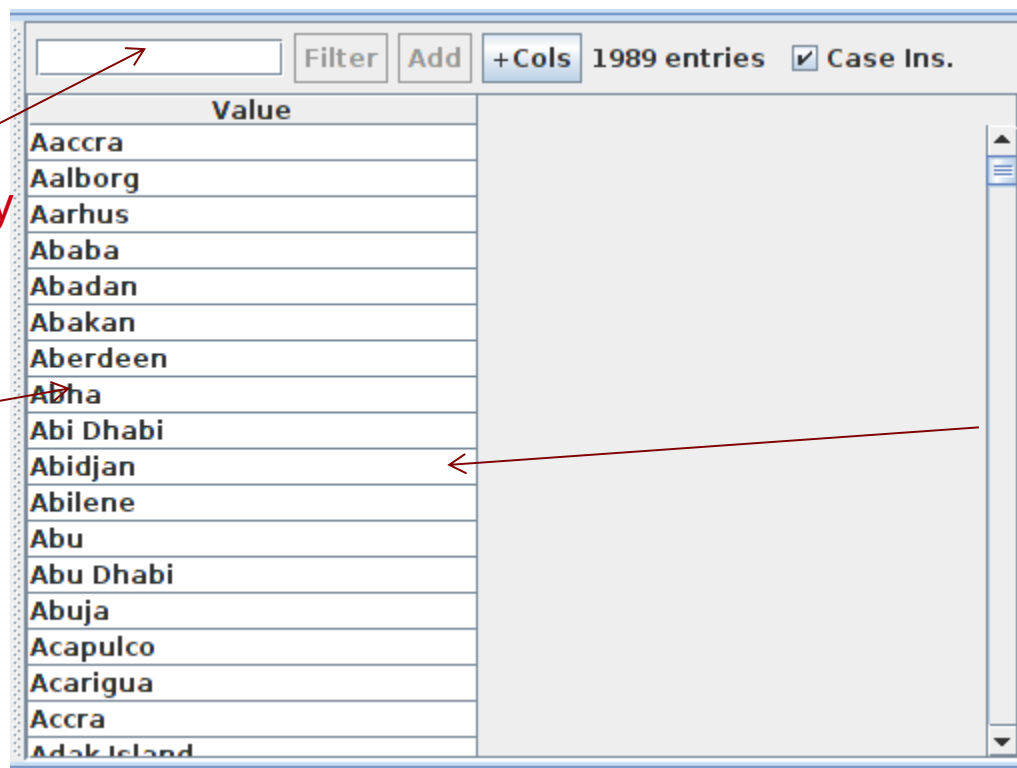
edit an existing list name by typing here

delete a list by right clicking on an entry and selecting Delete

A screenshot of the GATE definition file editor. At the top, there is a dropdown menu showing "airport.lst" and an "Add" button. Below this is a table with three columns: "List name", "Major", and "Minor". The table contains various entries such as "abbreviations.lst", "adbc.lst", "airports.lst", "charities.lst", "city.lst", "city\_cap.lst", "company.lst", "company\_cap.lst", "country.lst", "country\_abbrev.lst", "country\_adj.lst", "country\_cap.lst", "currency\_prefix.lst", "currency\_unit.lst", "date\_key.lst", "date\_unit.lst", and "day.lst". The "city.lst" row is highlighted in blue. Red arrows point from the text annotations to the "Add" button, the "city.lst" row, and the "Delete" button.

edit the major and minor Types by typing here

# Modifying a list



add a new entry  
by typing here

edit an  
existing entry  
by typing here

Delete an entry by  
right clicking and  
selecting "Delete"



# Editing gazetteer lists

- The ANNIE gazetteer has about 60,000 entries arranged in 80 lists
- Each list reflects a certain category, e.g. airports, cities, first names etc.
- List entries might be entities or parts of entities, or they may contain contextual information (e.g. job titles often indicate people)
- **Click on any list to see the entries**
- Note that some lists are not very complete!
- **Try adding, deleting and editing existing lists, or the list definition file**
- **To save an edited gazetteer, right click on the gazetteer name in the tabs at the top or in the resources pane on the right, and select “Save and Reinitialise” before running the gazetteer again.**
- **Try adding a word from a document you have loaded (that is not currently recognised as a Lookup) into the gazetteer, re-run the gazetteer and check the results.**



# Editing gazetteers outside GATE

- You can also edit both the definition file and the lists outside GATE, in your favourite text editor
- If you choose this option, you will need to reinitialise the gazetteer in GATE before running it again
- To reinitialise any PR, right click on its name in the Resources pane and select “Reinitialise”
- Note the difference between “Reinitialise” and “Save and Reinitialise”
  - “Renitialise” ignores any unsaved changes you made in the gazetteer editor in GATE, but incorporates changes made to the def file outside GATE
  - “Save and reinitialise” ignores any changes made outside GATE, but incorporates changes made within GATE



# List attributes

---

- When something in the text matches a gazetteer entry, a Lookup annotation is created, with various features and values
- The ANNIE gazetteer has the following default feature types: majorType, minorType, language
- These features are used as a kind of classification of the lists: in the definition file features are separated by “:”
- e.g. the “city” list has majorType “location” and minorType “city”
- Note that the way you define majorType and minorType is entirely up to you – there is no right or wrong thing to put here
- Later, in the JAPE grammars, we can refer to all Lookups of type location, or we can be more specific and refer just to those of type “city” or type “country”



---

# Named Entity Recognition





# NE transducer

- Gazetteers can be used to find terms that suggest entities
- However, the entries can often be ambiguous
  - “May Jones” vs “May 2010” vs “May I be excused?”
  - “Mr Parkinson” vs “Parkinson's Disease”
  - “General Motors” vs. “General Smith”
- Hand-crafted grammars can be used to define patterns over the Lookups and other annotations
- These patterns can help disambiguate, and they can combine different annotations, e.g. Dates can be comprised of day + number + month
- NE transducer consists of a number of grammars written in the JAPE language



# Rules for NER

---

- A typical simple pattern-matching rule might try to match all university names
- How could we match e.g. [University of Essex](#) with rules?
  - Match the string “[University of](#)” followed by a city name (from gazetteer)
  - We could generalise [University of](#) to “anything matched by a gazetteer containing names of words for educational establishments (school, college, etc.)”
- But what about:
  - [Sheffield Hallam University](#)
  - [University College London](#)
  - [Doncaster School for the Deaf](#)
  - [School for Scandal](#)
- It’s not always easy!

# Example JAPE rules for Organisations

---

- Ealing police

```
(  
{Lookup.majorType == location} | {Lookup.majorType == country_adj}  
{Lookup.majorType == organization} [1,2]  
)
```

- Royal Tuscan

```
(  
{Lookup.majorType == org_pre}  
{Token.orth == upperInitial}+  
{Lookup.majorType == org_ending}?  
)
```



# ANNIE NE Transducer

---

- Load an ANNIE NE Transducer PR
- Add it to the end of the application
- Run the application
- Look at the annotations
- You should see some new annotations such as Person, Location, Date etc.
- These will have features showing more specific information (eg what kind of location it is) and the rules that were fired (for ease of debugging)
- Double click on the ANNIE NE Transducer in LH pane (under Processing Resources) to see all grammar rules



---

# Co-reference



# Using co-reference

- 
- Different expressions may refer to the same entity
  - Orthographic co-reference module (orthomatcher) matches proper names and their variants in a document
  - **Mr Smith** and **John Smith** will be matched as the same person
  - **International Business Machines Ltd.** will match **IBM**



# Orthomatcher PR

- Performs co-reference resolution based on orthographical information of entities
- Produces a list of annotation IDs that form a co-reference “chain”
- List of such lists stored as a document feature named “MatchesAnnots”
- Improves results by assigning entity type to previously unclassified names, based on relations with classified entities
- May not reclassify already classified entities
- Classification of unknown entities very useful for surnames which match a full name, or abbreviations, e.g. **Bonfield** <Unknown> will match **Sir Peter Bonfield** <Person>
- A pronominal PR is also available



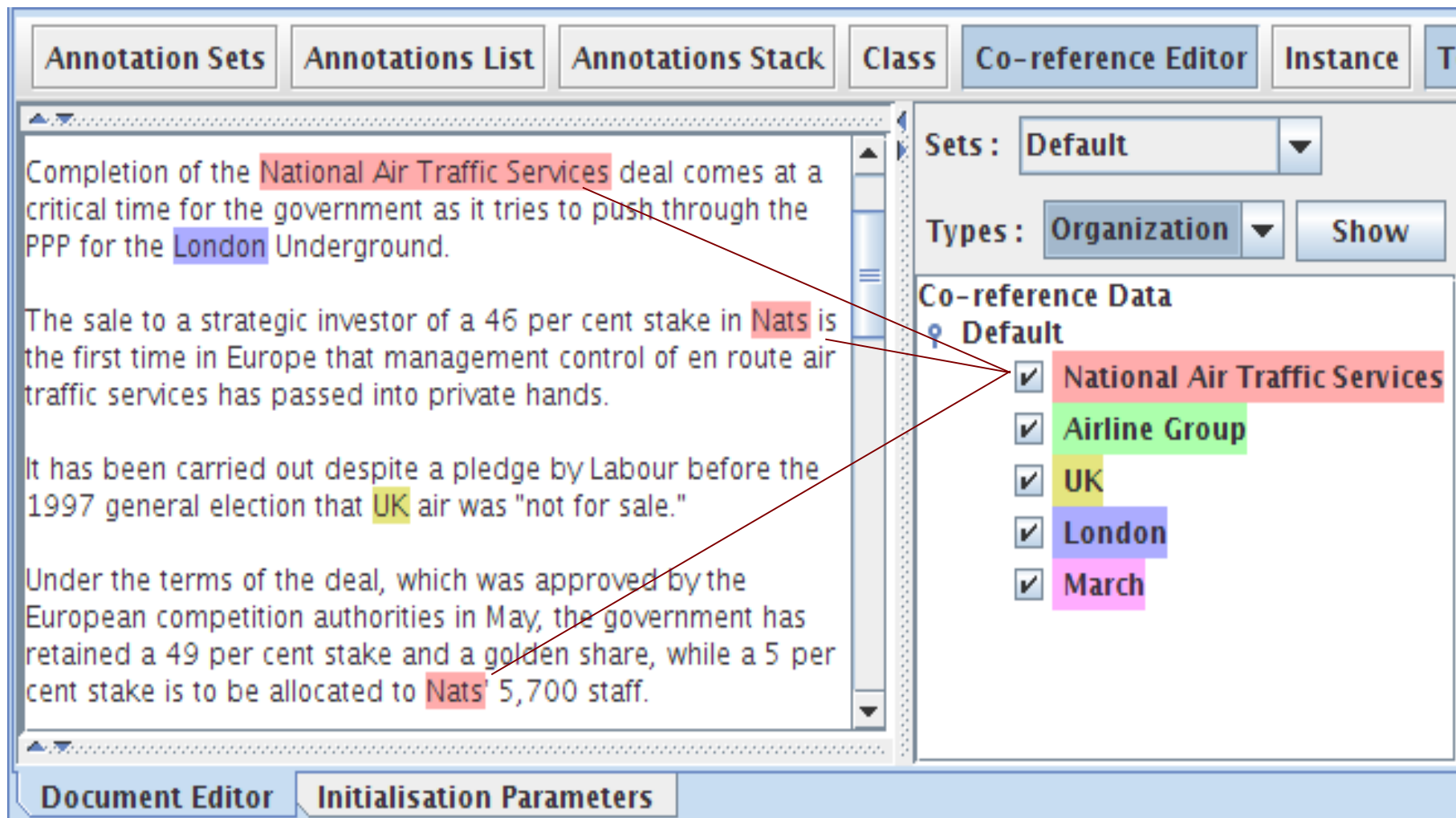
# Looking at co-reference

---

- Add a new PR: ANNIE OrthoMatcher
- Add it to the end of the application
- Run the application
- In a document view, open the co-reference editor by clicking the button above the text
- All the documents in the corpus should have some co-reference, but some may have more than others



# Coreference editor

The screenshot shows the GATE Coreference Editor window. At the top, there are several tabs: 'Annotation Sets', 'Annotations List', 'Annotations Stack', 'Class', 'Co-reference Editor' (which is active), 'Instance', and 'T...'. The main text area on the left contains four paragraphs of text with several words highlighted in colored boxes: 'National Air Traffic Services' (red), 'London' (blue), 'Nats' (red), 'UK' (yellow), and 'March' (pink). On the right side, there is a control panel. It includes a 'Sets:' dropdown menu set to 'Default', a 'Types:' dropdown menu set to 'Organization', and a 'Show' button. Below this is a section titled 'Co-reference Data' with a 'Default' label and a question mark icon. A list of five items is shown, each with a checked checkbox and a colored background matching the text highlights: 'National Air Traffic Services' (red), 'Airline Group' (green), 'UK' (yellow), 'London' (blue), and 'March' (pink). Red lines connect the highlighted text in the main area to their corresponding entries in the Co-reference Data list. At the bottom of the window, there are two tabs: 'Document Editor' and 'Initialisation Parameters'.



# Using the co-reference editor

- 
- Select the annotation set you wish to view (Default)
  - A list of all the co-reference chains that are based on annotations in the currently selected set is displayed
  - Select an item in the list to highlight all the member annotations of that chain in the text (you can select more than one at once)
  - Hovering over a highlighted annotation in the text enables you to Delete an item from the co-reference chain
  - Try it!



## Other NLP tools

- 
- There are lots of other NLP tools available in GATE (and elsewhere)
  - Term extraction – extracting key terms from text
  - Parsers – analysing the sentence structure
  - NP and VP chunking – shallow form of parsing (usually more accurate and efficient than full parsing)
  - Summarisation – making short summaries / abstracts of longer text



# TermRaider

- GATE plugin for detecting single and multi-word terms
- Terms are ranked according to three possible scoring systems:
  - $\text{tf.idf} = \text{term frequency (number of times the term occurs in the corpus)} \div \text{document frequency (number of documents in which the term occurs)}$
  - augmented  $\text{tf.idf}$  = after scoring  $\text{tf.idf}$ , the scores of hypernyms are boosted by the scores of hyponyms
  - Kyoto domain relevance =  $\text{document frequency} \times (1 + \text{nbr of hyponyms in the corpus})$



# TermRaider: Methodology

---

- After linguistic pre-processing (tokenisation, lemmatisation, POS tagging etc.), nouns and noun phrases are identified as initial term candidates
- Noun phrases include post-modifiers such as prepositional phrases, and are marked with head information for determining hyponymy. Nested nouns and noun phrases are all marked as candidates.
- Term candidates are then scored in 3 ways.
- The results can be viewed in the GATE GUI, exported as RDF, or saved as CSV files
- The viewer can be used to adjust the cutoff parameter. This is used to determine the score threshold for a term to be considered valid
- Terms can also be shown as a tag cloud



# Deciding what is a term

---

- Because TermRaider ranks every possible candidate term, you probably don't want to use all candidate terms if you're annotating terms in a text
- We therefore provide a cutoff mechanism to select what score should determine whether something is a term or not
- The last PR in TermRaider is a JAPE grammar which takes a feature “threshold” and a value, by default set to 45, and annotates candidates as “Term” only if the value of the augmented tf.idf is above the threshold.

File Options Tools Help

Messages TermRaider-Engl... JAPE termCandidateTh... HyponymyTermban...

ft-BT-loop-01-aug-2001.xml  
ft-BT-briefing-02-aug-2001.xml  
ft-BT-07-aug-2001.xml\_0000A  
china-sea1.txt\_00009  
GATE Corpus\_00008

Processing Resources

termCandidateThreshold

kyotoCopier  
augTfidfCopier  
tfidfCopier  
augmentation  
deduplicateMW  
multiwordJape  
selectTokens  
orthomatcher

threshold 45.0

Resource Features

Jape Viewer Initialisation Parameters

term-candidate-threshold

Phase: TermCandidate  
Input: SingleWord MultiWord  
Options: control = all

Rule: TermCandidate

```

((SingleWord){MultiWord}):match
-->
:match {
  Annotation ann = gate.Utills.getOnlyAnn(matchAnnots);
  FeatureMap oldf = ann.getFeatures();
  double threshold = 50.0; // fallback
  if (ctx.getPRFeatures().containsKey("threshold")) {
    threshold = Double.parseDouble(ctx.getPRFeatures().get("threshold").toString());
  }

  // Note that this reads a feature called 'threshold' on the PR itself.
  // To edit the feature in the GATE GUI, show the termCandidateThreshold PR
  // & look in the lower left corner. If the feature is missing,
  // the fallback given above is used.

  if (oldf.containsKey("tfidfAug") &&
      (((Double) oldf.get("tfidfAug")) > threshold)) {
    Long start = ann.getStartNode().getOffset();
    Long end = ann.getEndNode().getOffset();
  }
}

```

# Term candidates in a document

The screenshot shows the GATE Developer 7.2 interface. The main window displays a document with several paragraphs of text. The text is annotated with green highlights, indicating term candidates. The TermCandidate panel on the right shows the following details for a selected candidate:

| Property             | Value           |
|----------------------|-----------------|
| canonical            | figure          |
| category             | NN              |
| head                 | figure          |
| kind                 | word            |
| kyotoDomainRelevance | 43.761814424715 |
| length               | 6               |
| orth                 | lowercase       |
| root                 | figure          |
| string               | figure          |
| tfdidf               | 45.806158824792 |
| tfdidfAug            | 51.601390028462 |

The document text includes the following paragraphs:

larger within the party.

3.) "For a representative figure among reality-based Republicans I go with David Frum, the former speechwriter for George W. Bush and a conservative who cannot stomach what has happened to his party. Rather than become a Democrat or claim some sort of ideological conversion, Frum has taken up his pen, as with: When Did the GOP Touch With Reality? " There he writes:

Few of us have the self-knowledge and emotional discipline to say one thing while meaning another. If we say something often enough, we begin to believe it. We don't usually delude others until after we have first deluded ourselves. Some of the smartest and most sophisticated people know "canny investors, erudite authors" sincerely and passionately believe that President Barack Obama has gone far beyond conventional American liberalism and is willfully and relentlessly driving the United





# Try TermRaider in GATE

- Load the TermRaider plugin in GATE
- Load a corpus (around 20-100 documents on a similar topic is ideal, e.g. the news texts from the hands-on file)
- Load TermRaider from the “Ready-made Applications” and run it on the corpus
- Inspect the results (click on “SingleWord”, “MultiWord” or “Candidate Term” in the document viewer)
- Try the Term Cloud viewer
- Change the threshold (open the termCandidateThreshold PR in GATE and then modify the value of “threshold” in the box in the bottom left corner)



# Other NLP Toolkits

---

- GATE is not the only NLP toolkit (though we think it's the best!)
- Others include:
  - OpenCalais
  - UIMA
  - LingPipe
  - OpenNLP
  - Stanford Tools
- All integrated into GATE as plugins

- 
- UIMA is an NL engineering platform developed by IBM
  - Shares some functionality with GATE, but is complementary in most respects.
  - Interoperability layer has been developed to allow UIMA applications to be run within GATE, and vice versa, in order to combine elements of both.
  - Emphasis is on architectural support, including asynchronous scaleout (deploying many copies of an application in parallel)
  - Much narrower range of resources provided than GATE

<http://incubator.apache.org/uima/>

# OpenCalais



- 
- Web service for semantic annotation of text.
  - The user submits a document to the web service, which returns entity and relations annotations in RDF, JSON or some other format.
  - Typically, users integrate OpenCalais annotation of their web pages to provide additional links and ‘semantic functionality’.
  - OpenCalais annotates both relations and entities, although the GATE plugin only supports entities.

<http://www.opencalais.com>



# LingPipe

- 
- Provides set of IE and data mining tools largely ML-based. Has a set of models trained for particular tasks/corpora.
  - Limited ontology support: can connect entities found to databases and ontologies
  - Advantage: ML models can suggest more than one output, ranked by confidence. The user can choose number of suggestions generated.
  - Disadvantage: ML models only apply to specific tasks and domains.

<http://alias-i.com/lingpipe/index.html>



# Try them out!

---

- You can try some of these out by selecting them from the Ready-Made Applications
- You may need to first load the relevant plugin
  - Click the green jigsaw piece icon to get the Plugin manager and then select the plugin you want to load



# Summary

---

- Introduced the core NLP components used in typical text analysis tasks
- Demonstrated the ANNIE pipeline in GATE as an example of how to build up an Information Extraction pipeline
- Experimented with other tools within GATE
- Next, we'll show how to evaluate and compare results, and you can play with some further NLP components