

RuSSIR 2008

Taganrog, Russia. September 2008

Natural Language Processing for Information Access

Horacio Saggion
Department of Computer Science
University of Sheffield
England, United Kingdom
saggion@dc.shef.ac.uk

Overview of the course

- Introduction and Motivation (Day 1)
- NLP technology and tools (Day 1 & 2)
- Question Answering (Day 3 & 4)
- Text Summarization (Day 4 & 5)

Outline

- NLP for Information Access
- Information Retrieval
- Information Extraction
- Text Summarization
- Question Answering
- Cubreporter: a case study
- Other applications
- GATE tools for NLP
- Components
 - tokenisation
 - sentence splitting
 - part of speech tagging
 - named entity recognition
 - morphological analysis
 - parsing
- Demonstrations

Information Retrieval (Salton'88)

- Given a document collection and a user information needs
- Produces lists of documents matching the information need
- Information needs can be expressed as sets of keywords
- Documents are pre-processed in order to produce term indexes which contain information about where each term occurs in the collection - decisions have to be taken with regards to the definition of term

Information Retrieval

- Needs a method to measure the similarity between documents and queries
- The user has to read the documents in order to find the desired information
- IR can be applied to any domain
- Text Retrieval Conferences (since 1992) contributed to system development and evaluation (<http://trec.nist.gov>)

Information Extraction (Grishman'97)

- Pulls facts from the document collection
- Based on the idea of scenario template
 - some domains can be represented in the form of one or more templates
 - templates contain slots representing semantic information
 - IE instantiates the slots with values
- IE is domain dependent – a template has to be defined
- Message Understanding Conferences 1987-1997 fuelled the IE field and made possible advances in techniques such as Named Entity Recognition
- From 2000 the Automatic Content Extraction (ACE) Programme

Information Extraction

ALGIERS, May 22 (AFP) - At least 538 people were killed and 4,638 injured when a powerful earthquake struck northern Algeria late Wednesday, according to the latest official toll, with the number of casualties set to rise further ... The epicentre of the quake, which measured 5.2 on the Richter scale, was located at Thenia, about 60 kilometres (40 miles) east of Algiers, ...

DATE	21/05/2003
DEATH	538
INJURED	4,638
EPICENTER	Thenia, Algeria
INTENSITY	5.2, Richter

Information Extraction

- Template can be used to populate a data base
- Template can be used to generate a short summary of the input text

A 5.3 intensity earthquake in Algeria killed more than 500 people.
- Data base can be used to perform reasoning
 - What Algerian earthquake killed more people?

Information Extraction Tasks

- Named Entity recognition (NE)
 - Finds and classifies names in text
- Coreference Resolution (CO)
 - Identifies identity relations between entities in texts
- Template Element construction (TE)
 - Adds descriptive information to NE results
- Scenario Template production (ST)
 - Instantiate scenarios using TEs

Examples

- NE:
 - Thenia (Location), Algiers (Location), May 22 (Date), Wednesday (Date), etc.
- CO:
 - *a powerful earthquake* and *the quake*
 - *ALGIERS* and *Algiers*
- TE:
 - entity descriptions: Thenia is in Algeria, Algiers is capital of Algeria
- ST
 - combine entities in one scenario (as shown in the example)

Question Answering (Hirschman&Gaizauskas'01)

- Given a document collection (can be the Web) and a natural language question
- Extract the answer from the document collection
 - answer can in principle be any expression but in many cases questions ask for specific types of information such as person names, location, dates, etc.
- Open domain in general
- Text Retrieval Conferences Question Answering Track responsible for advances in the field of system development and evaluation (since 1999)
- From 2008 the Text Analysis Conference

QA Task (Voorhees'99)

- In the Text Retrieval Conferences (TREC) Question Answering evaluation, 3 types of questions are identified
- Factoid questions such as:
 - "Who is Tom Cruise married to?"
- List questions such as:
 - "What countries have atomic bombs?"
- Definition questions such as:
 - "Who is Aaron Copland?" or "What is aspirin?"
(Changed name to "other" question type)

Text Summarization (Mani'01)

- Given a document or set of documents
- Extract the most important content from it and present a condensed version of it
 - extracts vs abstracts
- Useful for decision making: read or not read; saves time; can be used to create surrogates; etc.
- Open domain, however domain knowledge proves important (e.g., scientific domain)
- Document Understanding Conferences (since 2000) contributed with much development in the field
- From 2008 the Text Analysis Conference

Integration of technologies for background gathering (Gaizauskas&al07)

- *Background gathering*: the task of collecting information from the news wire and other archives to contextualise and support a breaking news story
- Backgrounder components
 - similar events in the past; role players' profiles; factual information on the event
 - Collaboration with Press Association
 - 11 year archive with more than 8 million stories
 - Information access system comprising: Information Retrieval, Text Summarization, Question Answering, "Similar Event Search"

Background Examples

- Breaking News

"Powerful earthquake shook Turkey today"

- Past Similar Events

"Last year an earthquake measuring 6.3-magnitude hit southern Turkey killing 144 people."

- Extremes

"Europe's biggest quake hit Lisbon, Portugal, on November 1, 1755, when 60,000 people died as the city was devastated and giant waves 10 metres high swept through the harbour and on to the shore."

- Definitions

"Quakes occur when the Earth's crust fractures, a process that can be caused by volcanic activity, landslides or subterranean collapse. The resulting plates grind together causing the tremors."

Text Analysis Resources

- General Architecture for Text Engineering
 - (<http://gate.ac.uk>)
 - Tokenisation, Sentence Identification, POS tagging, NE recognition, etc.
- SUPPLE Parser
 - (<http://nlp.shef.ac.uk/research/supple>)
 - syntactic parsing and creation of logical forms
- Summarization Toolkit
 - (<http://www.dcs.shef.ac.uk/~saggion>)
 - Single and multi document summarization
- Lucene
 - (<http://lucene.apache.org>)
 - Text indexing and retrieval

Summarization System

- Scores sentences based on numeric features in both single and multi-document cases
 - position of sentence, similarity to headline, similarity to cluster centroid, etc.
 - values are combined to obtain the sentence score
- Sentences represented as terms of vectors used to compute similarities (e.g. cosine metric)
- Redundancy detection for MDS is based on *n-gram* overlap measure
- Press Association profiles are automatically identified
- Other profiles created using QA/summarization techniques

Question Answering

- Passage (i.e., paragraph) retrieval using question
- Question and passage analysis using SUPPLE and discourse interpretation (i.e., coreference)
 - expected answer type identified (EAT)
- Answer candidates in passages scored using
 - sentence score (overlap with question)
 - similarity of candidate answer to EAT (using WordNet)
 - count relations between candidate and question entities
 - merge scores across passages

Semantic Representations

- Leading paragraphs are parsed using SUPPLE and semantic representations created
 - *"The head of Australia's biggest bank resigned today"*
 - *head(e2), name(e4, 'Australia'), country(e4), of(e3, e4), bank(e3), adj(e3, biggest), of(e2, e3), resign(e1), lsubj(e1, e2)*
- Database records are created and used to support similar event search
- We can search for "resignation" events
 - "resign" -> leave job = quit, renounce, leave office
 - "head" -> person in charge = chief

Semantic Representations

- Word Senses are been generated using word centroids and cosine similarity (Aguirre & de Lacalle'03)
- "resign" transformed into "leave job" sense
~ renounce, leave office, step down, etc.
- "head" transformed into "person in charge"
~ chief
- Search based on matching semantic representations

Finding Stories

auto summaries

profiles

metadata

stories

The screenshot shows a web browser window titled 'CubReporter Search Results - Microsoft Internet Explorer'. The address bar shows 'http://don.dcs.shef.ac.uk/cubreporter/documentResult.jsp'. The search bar contains 'Tony Blair meets George Bush'. Below the search bar, there is a results summary: 'CubReporter Results 1 - 10 of about 1146995 for Tony+Blair+meets+George+Bush, Current Page: 1, Total: 114700 pages. (1)'. Below this, it says 'Your query includes the following people, click on to to view their profiles: George Bush Tony Blair'. There are two story entries listed. The first entry has a headline 'BLAIR PREPARES FOR BUSH MEETING' and a summary 'Mr Blair also announced, in a speech to a joint session of the Canadian Parliament, plans for a new political drive to improve relations between the EU and the North American Free Trade Area. The two first ladies, Mr Blairs wife Cherie and Laura Bush will also meet for the first time. Mrs Blair had forged a close bond with her fellow lawyer Hillary Clinton.' The second entry has a headline 'BLAIR AND BUSH FINALLY MEET' and a summary 'Both First Ladies wore dark overcoats and trousers, Mrs Bushs coat mid-length'. Both entries have a date of '2001-02-22' and a word count of '438'. There are links for 'Track it!' and 'Save!' next to the first entry.

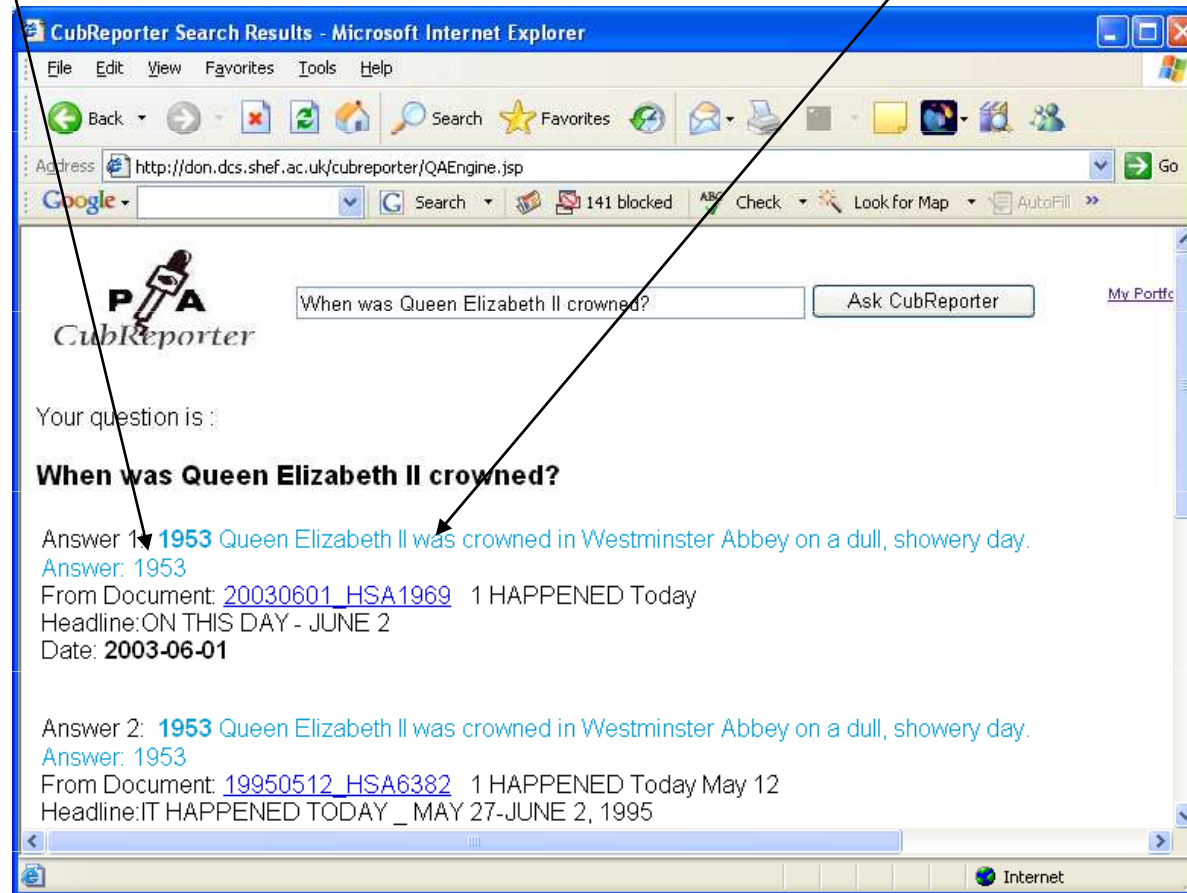
20010222_HSA7019 1 POLITICS Blair Nightlead
Headline:BLAIR PREPARES FOR BUSH MEETING
Summary:Mr Blair also announced, in a speech to a joint session of the Canadian Parliament, plans for a new political drive to improve relations between the EU and the North American Free Trade Area. The two first ladies, Mr Blairs wife Cherie and Laura Bush will also meet for the first time. Mrs Blair had forged a close bond with her fellow lawyer Hillary Clinton.
Content:By John Smith, Political Editor PA News in Ottawa, Canada Prime Minister Tony Blair will tomorrow head for Washington for his first summit with new US President George W Bush. Mr Blair will leave Canada in the early hours after a day of intense talks...
Date:2001-02-22 Word Count:438 [Track it] [Save]

20010223_HSA0470 1 POLITICS Blair Nightlead
Headline:BLAIR AND BUSH FINALLY MEET
Summary:Both First Ladies wore dark overcoats and trousers, Mrs Bushs coat mid-length

Getting Answers

answers

context



Getting Similar Events

“jet dropped bomb in Iraq”

jets drop bombs

bombs dropped

CubReporter Search Results - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Reload Print Mail News RSS Feeds

Address <http://don.dcs.shef.ac.uk/cubreporter/testResult1.jsp> Go

Google Search 141 blocked ABC Check Look for Map AutoFill Options

19990301_HSA0366 1 IRAQ Attack

Headline: **US JETS US BOMB IRAQI DEFENCES**

Summary: The US aircraft targeted Iraqi communication, radio relay and anti-aircraft artillery sites after being targeted several times by Iraqi radar, said spokesman Captain Mike Blass. Blass denied Iraqi reports that US planes hit or targeted an Iraqi oil pipeline running through to Turkey or one of its pumping stations.

Content: Air Force jets dropped more than 30 bombs on Iraqi military installations in the northern no fly zone today in a second consecutive day of attacks. The US aircraft targeted Iraqi communication, radio relay and anti-aircraft artillery sites after being...

Date: 1999-03-01 Word Count: 117 [Track it](#) [Save](#)

20040301_HSA0012 1 DEFENCE Bomb

Headline: **AIR CREW TO BLAME FOR DUMMY BOMB BLUNDER**

Summary: "Subsequent to the determination of air crew error, the crew was retrained. They have returned to flying status after satisfactorily demonstrating proficiency in actions required to safely conduct combat training. The "bomb crashed through an area of concrete close to industrial buildings on the Holme Industrial Estate, in Holme upon Spalding Moor, and left a hole 18ins in diameter.

Content: By Amy Caulfield, PA News A air crew responsible for dropping a dummy bomb on farmland by mistake has returned to flying after an investigation into the error, it was revealed today. The US military fighter jet, based at RAF Lakenheath in Suffolk, re...

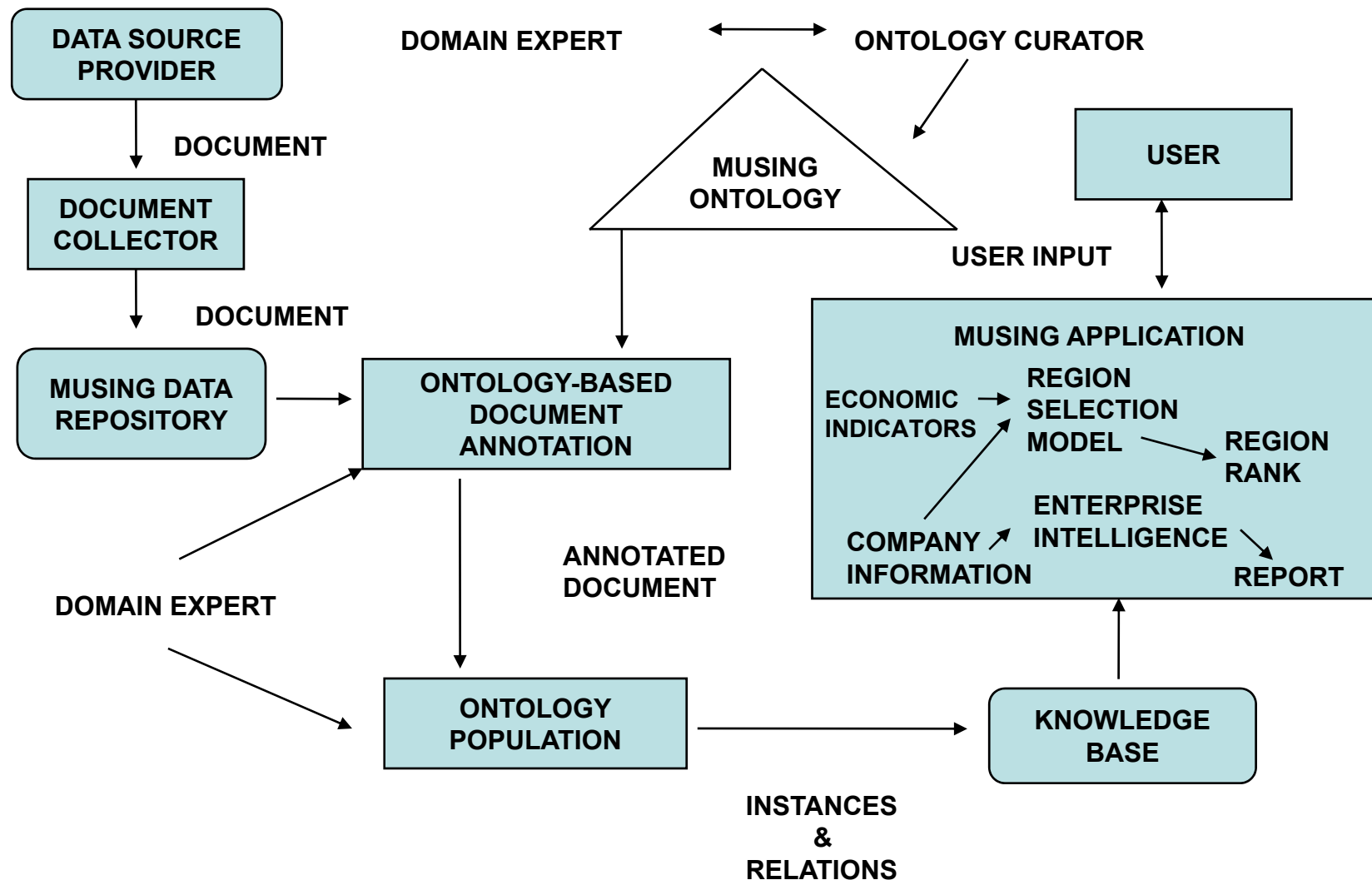
Date: 2004-03-01 Word Count: 289 [Track it](#) [Save](#)

Internet

Extracting information for business intelligence applications

- MUSING Project – 6FP European Commission (ICT)
 - integration of natural language processing and ontologies for business intelligence applications
 - extraction of company information
 - extraction of country/region information
 - identification of opinions in text for company reputation

Ontology-based IE in MUSING



Data Sources in MUSING

- Data sources include balance sheets, company profiles, press data, web data, etc. (some private data)
 - News papers from Italian financial news provider
 - Companies' web pages (main, "about us", "contact us", etc.)
 - Wikipedia, CIA Fact Book, etc.
- Ontology is manually developed through interaction with domain experts and ontology curators
 - It extends the PROTON ontology and covers the financial, international, and IT operational risk domain

Company Information in MUSING

GATE 4.0 build 2794

File Options Tools Help

Messages OWLIM Ontology LR_00016

Classes & Instances Properties

Classes and Instances

- Competitor
- Customer
- Object
 - Account
 - Activity
 - Agent
 - Group
 - BoardOfDirectors
 - Organization
 - Charity
 - CommercialOrganization
 - Company**
 - Airline
 - Bank
 - InsuranceCompany
 - MediaCompany
 - NewsAgency
 - PublishingComp:
 - TVCompany
 - PublicCompany
 - SportClub
 - SoccerClub
 - Telecom

- Division

generatedBy [EntitySource]

hasAccountingItems [ALL CLASSES]

hasAddress [Address]

hasAlias [Alias]

hasAuditSystem [AuditSystem]

hasBusinessLines [BusinessLine]

hasBusinessProfile [BusinessProfile]

hasContactInfo [ContactInformation]

hasCreditExposures [CreditExposure]

hasCreditHistory [CreditHistory]

hasDestinationCompany [ALL CLASSES]

hasEmail [Email]

hasEmployee [Person]

hasFax [PhoneNumber]

hasFinancialCommitments [FinancialCommitment]

hasFinancialRatios [FinancialRatio]

hasGuarantees [FinancialSecurity]

hasIT [EnterpriseIT]

hasImpact <http://www.w3.org/2001/XMLSchema#string>

hasInternetAddress [InternetAddress]

hasLeader [Person]

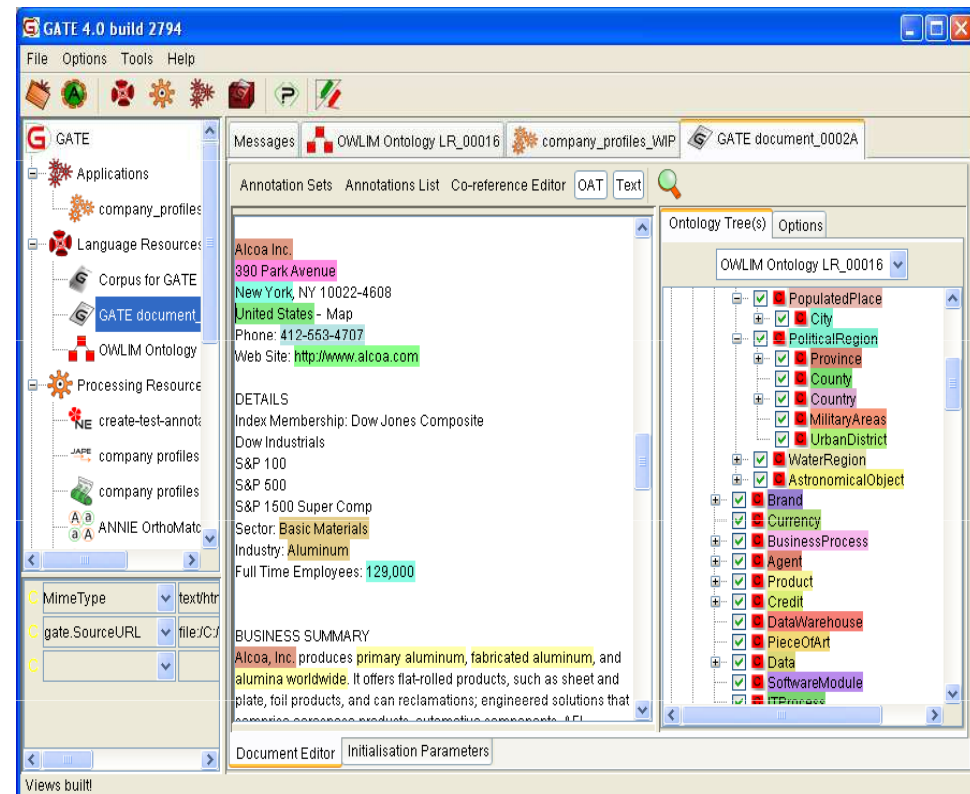
hasMainAlias [Alias]

GATE Ontology Editor Initialisation Parameters

Views built!

Extracting Company Information

- Extracting information about a company requires for example identify the Company Name; Company Address; Parent Organization; Shareholders; etc.
- These associated pieces of information should be asserted as properties values of the company instance
- Statements for populating the ontology need to be created ("Alcoa Inc" hasAlias "Alcoa"; "Alcoa Inc" hasWebPage "http://www.alcoa.com", etc.)



General Architecture for Text Engineering

– GATE (Cunningham&al'02)

- Framework for development and deployment of natural language processing applications
- A graphical user interface allows users (computational linguists) access, composition and visualisation of different components and experimentation
- A Java library (gate.jar) for programmers to implement and pack applications

Component Model

- Language Resources (LR)
 - data
- Processing Resources (PR)
 - algorithms
- Visualisation Resources (VR)
 - graphical user interfaces (GUI)
- Components are extendable and user-customisable
 - for example adaptation of an information extraction application to a new domain
 - to a new language where the change involves adaptation of a module for word recognition and sentence recognition

Documents in GATE

- A document is created from a file located somewhere in your disk or in a remote place or from a string
- A GATE document contains the “text” of your file and sets of annotations
- When the document is created and if a format analyser for your type is available “parsing” (format) will be applied and annotations will be created
 - xml, sgml, html, etc.
- Documents also store features, useful for representing metadata about the document
 - some features are created by GATE

Documents in GATE

- Annotations have
 - types (e.g. Token)
 - belong to particular annotation sets
 - start and end offsets – where in the document
 - features and values which are used to store orthographic, grammatical, semantic information, etc.
- Documents can be grouped in a Corpus
- Corpus is other language resource in GATE which implements a set of documents

/ names in text



Annotation Guidelines

- People need clear definition of what to annotate in the documents, with examples
- Typically written as a guidelines document
- Piloted first with few annotators, improved, then “real” annotation starts, when all annotators are trained
- Annotation tools require the definition of a formal DTD (e.g. XML schema)
 - What annotation types are allowed
 - What are their attributes/features and their values
 - Optional vs obligatory; default values

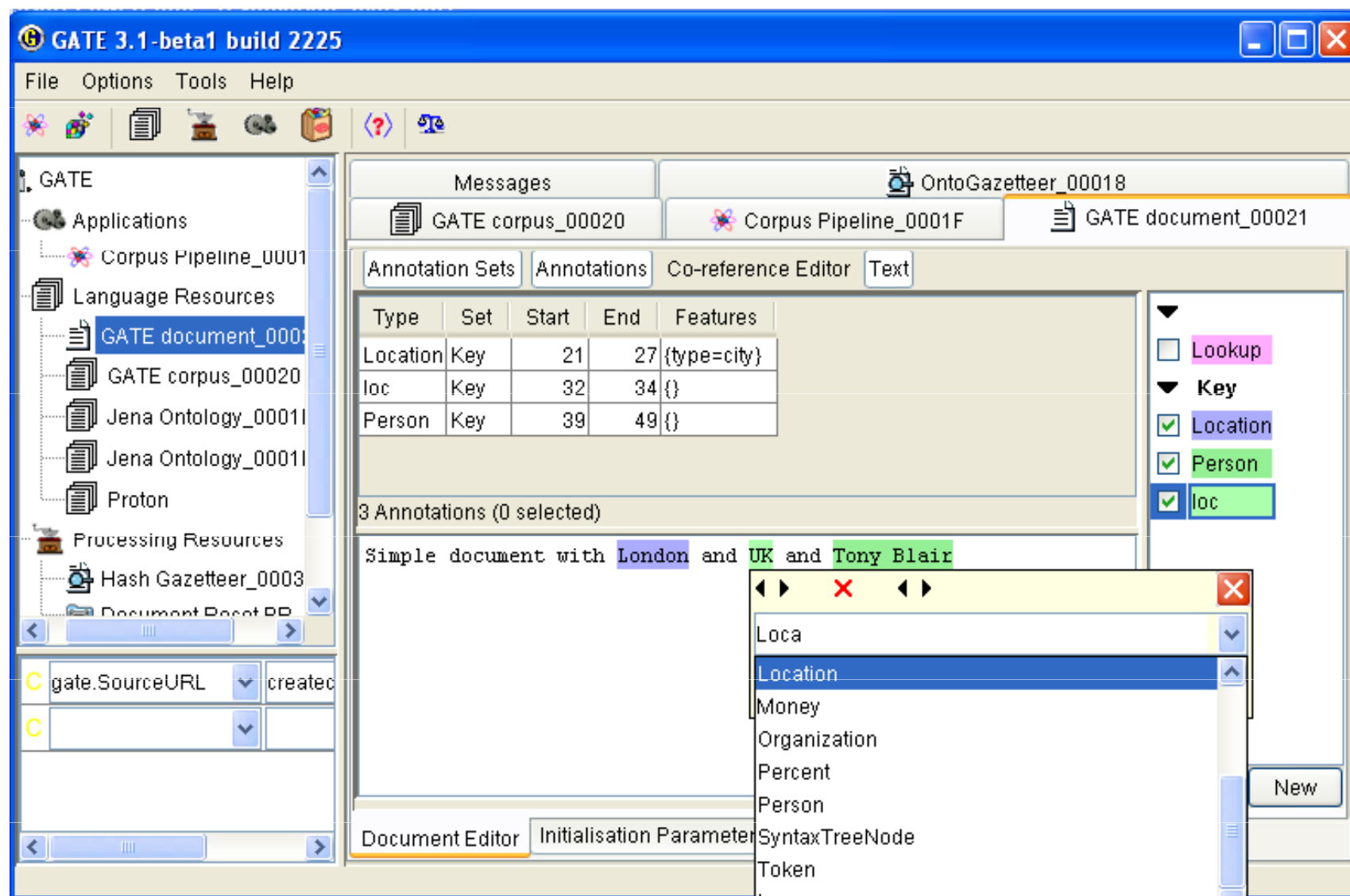
Annotation Schemas

```
<?xml version="1.0"?>
<schema
  xmlns="http://www.w3.org/2000/10/XMLSchema">
  <!-- XSchema definition for email-->
  <element name="Email" />
</schema>
```

Annotation Schemas

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2000/10/XMLSchema">
  <!-- XSchema definition for token-->
  <element name="Address">
    <complexType>
      <attribute name="kind" use="optional">
        <simpleType>
          <restriction base="string">
            <enumeration value="email"/>
            <enumeration value="url"/>
            <enumeration value="phone"/>
            <enumeration value="ip"/>
            <enumeration value="street"/>
            <enumeration value="postcode"/>
            <enumeration value="country"/>
            <enumeration value="complete"/>
          </restriction> ...
        </simpleType>
      </attribute>
    </complexType>
  </element>
</schema>
```

Manual Annotation in GATE GUI



Annotation in GATE GUI

The following tasks can be carried out manually in the GATE GUI:

- Adding annotation sets
- Adding annotations
- Resizing them (changing boundaries)
- Deleting
- Changing highlighting colour
- Setting features and their values

Preserving and exporting results

- Annotations can be stored as stand-off markup or in-line annotations
- The default method is standoff markup, where the annotations are stored separately from the text, so that the original text is not modified
- A corpus can also be saved as a regular or searchable (indexed) datastore

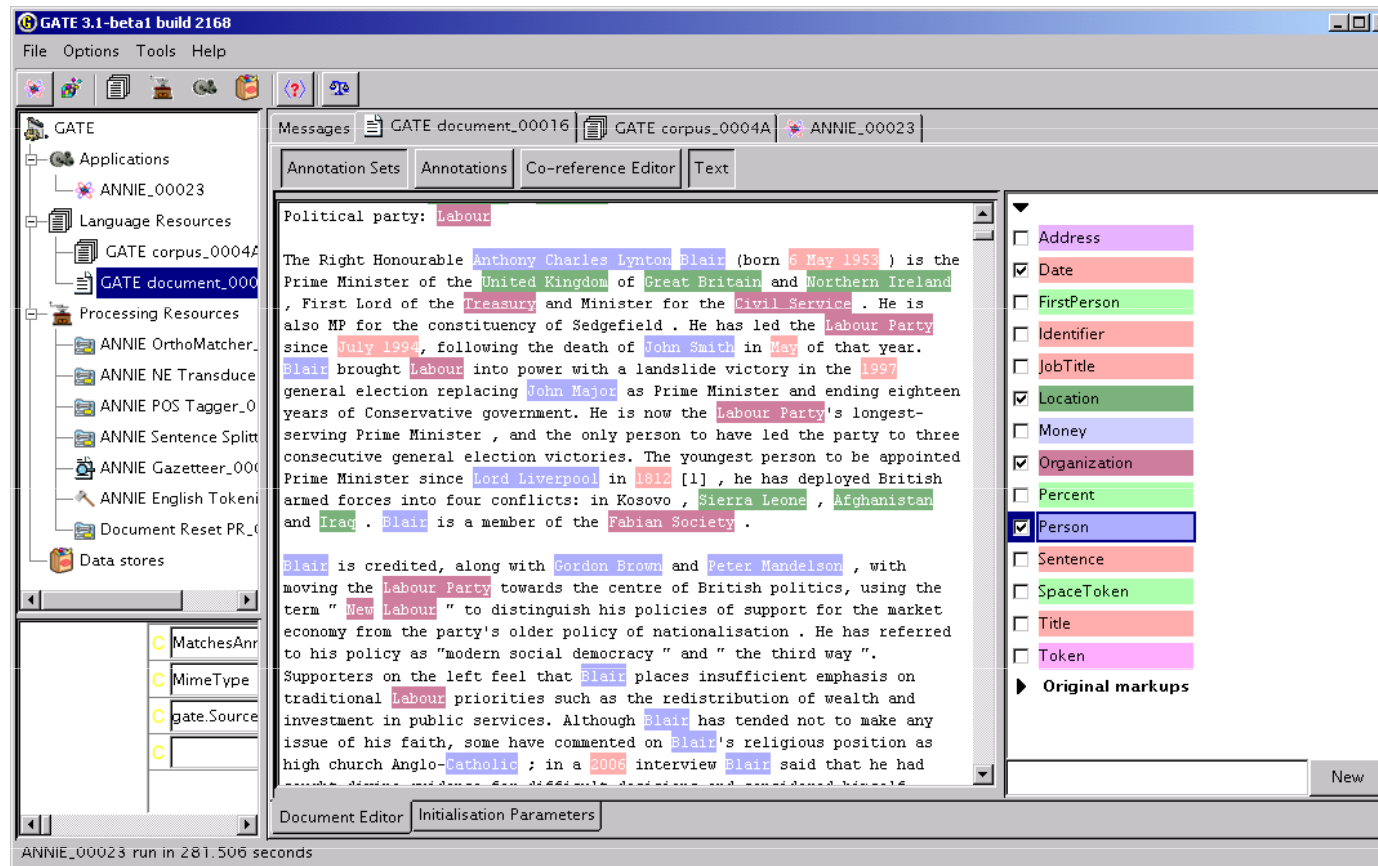
Corpora and System Development

- “Gold standard” data created by manual annotation
- Corpora are divided typically into a training, sometimes testing, and unseen evaluation portion
- Rules and/or ML algorithms developed on the training part
- Tuned on the testing portion in order to optimise
 - Rule priorities, rules effectiveness, etc.
 - Parameters of the learning algorithm and the features used
- Evaluation set – the best system configuration is run on this data and the system performance is obtained
- No further tuning once evaluation set is used!

Applications in GATE

- Applications are created by sequencing processing resources
- Applications can be run over a Corpus of documents – corpus pipeline
 - so each component is applied to each document in the corpus in sequence
- Applications may not have a corpus as input, but different parameters – pipeline

Name Entity Recognition



Text Processing Tools

- Document Structure Analysis
 - different document parsers take care of the structure of your document (xml, html, etc.)
 - Tokenisation
 - Sentence Identification
 - Parts of speech tagging
 - Morphological analysis
-
- All these language resources have as runtime parameter a GATE document, and they will produce annotations over it
 - Most resources have initialisation parameters

Creole

- a Collection of REusable Objects for Language Engineering ~ Language Resources + Processing Resources
- creole.xml provides details about available components, the java class that implements the resource, and jar file where it is found

Example of resource

<RESOURCE>

```
<NAME>ANNIE English Tokeniser</NAME>
<CLASS>gate.creole.tokeniser.DefaultTokeniser</CLASS>
<COMMENT>
    A customisable English tokeniser
    (http://gate.ac.uk/sale/tao/#sec:en-tokeniser).
</COMMENT>
<PARAMETER NAME="document"
    COMMENT="The document to be tokenised" RUNTIME="true">
    gate.Document
</PARAMETER>
<PARAMETER NAME="annotationSetName" RUNTIME="true"
    COMMENT="The annotation set to be used for the generated annotations"
    OPTIONAL="true">
    java.lang.String
</PARAMETER>
<PARAMETER NAME="tokeniserRulesURL"
    DEFAULT="resources/tokeniser/DefaultTokeniser.rules"
    COMMENT="The URL for the rules file" SUFFIXES="rules">
    java.net.URL
</PARAMETER>
<PARAMETER NAME="transducerGrammarURL"
    DEFAULT="resources/tokeniser/postprocess.jape"
    COMMENT="The URL for the postprocessing transducer"
    SUFFIXES="jape">
    java.net.URL
</PARAMETER>
<PARAMETER NAME="encoding"
    COMMENT="The encoding used for reading the definitions"
    DEFAULT="UTF-8">
    java.lang.String
</PARAMETER>
<ICON>tokeniser</ICON>
```

</RESOURCE>

Tokenisation

- Identify different words in text: numbers, symbols, words, etc.
 - not only sequences between spaces or separators (in English)
 - 2,000 is not "2" "," "000"
 - I've is "I" and "'ve"
 - \$20 is "\$" and "20"


Tokenisation in GATE

- Rule-based LHS > RHS
 - LHS is a regular expression over character classes
 - RHS specifies annotation to be created and features to be asserted for the created annotation
 - "DECIMAL_DIGIT_NUMBER"+ >Token;kind=number;
 - (SPACE_SEPARATOR) >SpaceToken;kind=space;
 - (CONTROL) >SpaceToken;kind=control;
 - "UPPERCASE_LETTER" (LOWERCASE_LETTER (LOWERCASE_LETTER|DASH_PUNCTUATION|FORMAT))* > Token;orth=upperInitial;kind=word;
- Tokeniser produces a Token type of annotation

Tokenisation in GATE

- Features produced by the tokeniser
 - string: the actual string of the token
 - orth: orthographic information
 - length: the length of the string
 - kind: the type of token (word, symbol, punctuation, number)
- SpaceToken is another annotation produced
 - features: kind (control or space), length, string

Sentence Splitter

- 
- Decide where a sentence ends
 - “The court ruled that Dr. Smith was innocent.”
 - A rule based mechanism
 - uses a list of known abbreviations to help identify end of sentence uses (e.g. Dr)
 - a period is a sentence break if it is preceded by a non-abbreviation and followed by an uppercase common word
 - The Splitter in GATE produces a Sentence type of annotation

Parts of Speech Tagging (Hepple'00)

- Associate a part of speech tag to each word
 - tags from the Penn Treebank including punctuation
- Based on Brill's tagger but a different learning approach used for rule acquisition – no need for re-annotating the corpus at each iteration during learning
- Two steps during tagging
 - initial guess based on lexicon (contain most likely tag)
 - correction based on a list of rules (contextual)

POS tags used

CC - coordinating conjunction: "and", "but", "nor", "or", "yet", plus, minus, less, times (multiplication), over (division). Also "for" (because) and "so" (i.e., "so that").

CD - cardinal number

DT - determiner: Articles including "a", "an", "every", "no", "the", "another", "any", "some", "those".

EX - existential there: Unstressed "there" that triggers inversion of the inflected verb and the logical subject; "There was a party in progress".

FW - foreign word

IN - preposition or subordinating conjunction

JJ - adjective: Hyphenated compounds that are used as modifiers; happy-go-lucky.

JJR - adjective - comparative: Adjectives with the comparative ending "-er" and a comparative meaning. Sometimes "more" and "less".

JJS - adjective - superlative: Adjectives with the superlative ending "-est" (and "worst"). Sometimes "most" and "least".

JJSS - -unknown-, but probably a variant of JJS

-LRB- - -unknown-

LS - list item marker: Numbers and letters used as identifiers of items in a list.

MD - modal: All verbs that don't take an "-s" ending in the third person singular present: "can", "could", "dare", "may", "might", "must", "ought", "shall", "should", "will", "would".

NN - noun - singular or mass

NNP - proper noun - singular: All words in names usually are capitalized but titles might not be.

NNPS - proper noun - plural: All words in names usually are capitalized but titles might not be.

NNS - noun - plural

PDT - predeterminer: Determinerlike elements preceding an article or possessive pronoun; "all/PDT his marbles", "quite/PDT a mess".

POS - possessive ending: Nouns ending in "'s" or "'".

PP - personal pronoun

PRPR\$ - unknown-, but probably possessive pronoun

PRP - unknown-, but probably possessive pronoun

PRP\$ - unknown, but probably possessive pronoun, such as "my", "your", "his", "his", "its", "one's", "our", and "their".

RB - adverb: most words ending in "-ly". Also "quite", "too", "very", "enough", "indeed", "not", "n't", and "never".

RBR - adverb - comparative: adverbs ending with "-er" with a comparative meaning.

RBS - adverb - superlative

RP - particle: Mostly monosyllabic words that also double as directional adverbs.

STAART - start state marker (used internally)

SYM - symbol: technical symbols or expressions that aren't English words.

TO - literal to

UH - interjection: Such as "my", "oh", "please", "uh", "well", "yes".

VBD - verb - past tense: includes conditional form of the verb "to be"; "If I were/VBD rich...".

VBG - verb - gerund or present participle

VCN - verb - past participle

VBP - verb - non-3rd person singular present

VB - verb - base form: subsumes imperatives, infinitives and subjunctives.

VBZ - verb - 3rd person singular present

WDT - wh-determiner

WP\$ - possessive wh-pronoun: includes "whose"

WP - wh-pronoun: includes "what", "who", and "whom".

WRB - wh-adverb: includes "how", "where", "why". Includes "when" when used in a temporal sense.

Parts of Speech Tagging

- Two resources
 - lexicon collected from corpus with <word, list of valid tags>
 - employs VBZ
 - empty JJ VB VBP
 - some heuristics for unknown words
 - rules for correcting tagging mistakes
 - NN VBG PREVWD before
 - rules instantiate patterns such as:
 - Change tag A to tag B if Condition
- The GATE tagger produces a feature category for each token in the document, the value of the feature is the name of the POS tag

Morphological Analysis in GATE

- For each noun and verb in the document identifies lemma and affix which are stored in the Token annotation ('root', 'affix')
- A set of rules for regular cases is used
- A set of irregular cases which explicitly indicate how to decompose the word is also used

Stemming in GATE

- Removing prefixes and suffixed of a word
 - produces a feature 'stem' in the Token annotation
 - John -> stem=john
 - tells -> stem=tell
 - considered -> stem=consid (root=consider)
 - leaving -> stem=leav (root=leave)
 - had -> stem=had (root=have)
- Available for English and other languages (e.g. Spanish)

Named Entity Recognition

- It is the cornerstone of many NLP applications – in particular of IE
- Identification of named entities in text
- Classification of the found strings in categories or types
- General types are Person Names, Organizations, Locations
- Others are Dates, Numbers, e-mails, Addresses, etc.
- Domains may have specific NEs: film names, drug names, programming languages, names of proteins, etc.

NER problems



- There are problems even with well known categories
- “Ambrose Chapel” it’s not a name it is a place!!!!
- Ambiguity is one problem
 - “Paris” can be a city or a person
 - Paris (for Paris Hilton, the Person); Paris Hilton hotel (the place)
 - London can be a place or an organization (the government)

Approaches to NER

- Knowledge-based approach
 - Word level information is useful in recognising entities:
 - capitalization, type of word (number, symbol)
 - Specialized lexicons (Gazetteer lists) usually created by hand; although methods exist to compile them from corpora
 - List of known continents, countries, cities, person first names
 - On-line resources are available to pull out that information

Approaches to NER

- In GATE Gazetteers lists entries may contain some useful semantic information
 - for example one may associate some features and values to entry names
 - features can be used in grammars or can be used to enrich system output
 - gazetteer lists are organized in index files

Approaches to NER

- Knowledge-based approach
 - Rules are used to combine different evidences
 - a known first name followed by a sequence of words with upper initial may indicate a person name
 - a upper initial word followed by a company designator (e.g., Co., Ltd.) may indicate a company name
- A cascade approach is generally used where some basic names are first identified and are latter combined into more complex names

Gazetteers in GATE

- Lists store keywords (one keyword per line)
 - list of male names (person_male.lst)
Aaron
Abraham
....
- Set of lists compiled and a finite state machine is created which operates on the strings
- The machine produces annotations of type Lookup when the keyword is found in text
- 60k entries in 80 types:
 - organization; artifact; location; amount_unit; manufacturer;

Gazetteer in GATE

- Sets of lists are organized in a main lists file
- Each list specifies attributes MajorType and MinorType and Language, having major and minor types gives some flexibility to grammar rules
 - government.lst:organization:government
 - department.lst:organization:department
 - person_male.lst:person_first:male
 - person_female.lst:person_first:female
- Attributes are used to help identification of more complex entities (for example discriminating when possible between a male or female name)
- List entries may be entities or parts of entities, or they may contain contextual information (e.g. job titles often indicate people)

Named Entity Grammar in GATE

- Implemented in the JAPE language (part of GATE)
 - Regular expressions over annotations
 - Provide access and manipulation of annotations produced by other modules
- Rules are stored in grammar files
- Grammar files are compiled into Finite State Machines
- A main grammar files specifies how different grammars should be executed (phases)
 - constitute a cascade of FSTs over annotations

NER in GATE

-
- Rules are hand-coded, so some linguistic expertise is needed here
 - uses annotations from tokeniser, POS tagger, and gazetteer modules
 - use of contextual information
 - rule priority based on pattern length, rule status and rule ordering
 - Common entities: persons, locations, organisations, dates, addresses.

JAPE Language

- A JAPE grammar rule consists of a left hand side (LHS) and a right hand side (RHS)
 - LHS= what to match (the pattern)
 - RHS = how to annotate the found sequence
- A JAPE grammar is a sequence of grammar rules
- Grammars are compiled into finite state machines
- Rules have priority (number)
- There is a way to control how to match
 - options parameter in the grammar files

LHS of JAPE rules

- The LHS of the rule contains patterns to be matched, in the form of annotations (and optionally their attributes).
- Annotation types to be recognized must be declared at the beginning of the phase
- Annotations may be combined using traditional operators [| * + ?]

LHS of JAPE rules

- There is no negative operator
- More than one pattern can be matched in a single rule
- Left and right context (not to be annotated) can be matched
- LHS has labels to be referred to in RHS

Examples of LHS patterns

- //identify a token with upper initial
(`{Token.orth == upperInitial}`):upper
- //recognise a sequence of one upper initial word followed by a location designator (e.g. Ennerdale Lake)
(`{Token.orth == upperInitial}`
`{Lookup.majorType == loc_designator}`) :location
- //same but with upper initial or all capitals
((`{Token.orth == upperInitial}` | `{Token.orth == allCaps}`)
`{Lookup.majorType == loc_designator}`) :location

Example of RHS

`({Token.orth == upperInitial}
 {Lookup.majorType ==
 lake_designator}) :location`

→

`:location.Location = { type = "lake" }`

- Indicates annotation type to be produced 'Location' and features and values for that annotation type

Macros in JAPE grammars

Macro: ONE_DIGIT
({Token.kind == number, Token.length == "1"})

Macro: TWO_DIGIT
({Token.kind == number, Token.length == "2"})

Macro: FOUR_DIGIT
({Token.kind == number, Token.length == "4"})

Macro: DAY_MONTH_NUM
(ONE_DIGIT | TWO_DIGIT)

In the LHS of the rule one can use the macro name:

(DAY_MONTH_NUM):annotate -> :annotate.DAY = {}

Example of RHS (context)

Rule: Date

```
({Token.string == "Date"} {Token.string == ":"}):context  
(({Token.kind == "number", Token.length == "2"})  
 {Token})  
({Token.kind == "number", Token.length == "2"})  
 {Token})  
({Token.kind == "number", Token.length == "2"}):annotate  
-->  
:annotate.Date = { type = "dd/mm/yy format" }
```

Further processing in RHS

- Java code can be included in the RHS of the rule
- It is a powerful mechanism which can help add semantic information to the annotations
 - for example extracting information from the context

Available Java objects

- bindings: labels used in LHS are available
- doc: the GATE document which is being process
- annotations: all GATE document annotations produced until that stage
- inputAS, outputAS: phase input and output annotations

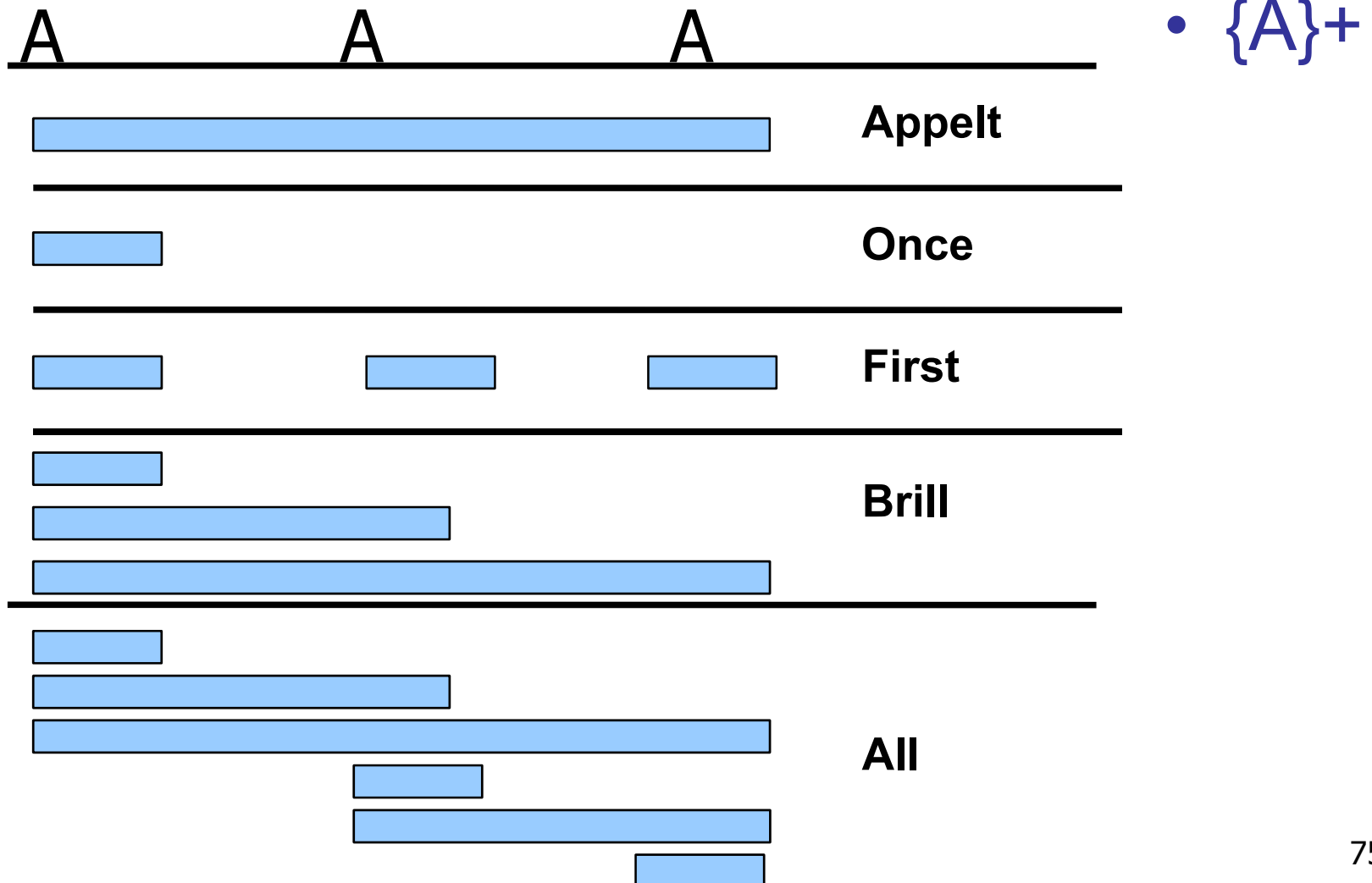
JAPE Application modes

- Matching control for rules
 - Brill (fires all matches)
 - First (shortest match fires)
 - Once (Phase exits after first match)
 - All (as for Brill, but matching continues from offset following the current one, not from the end of the last match)
 - Appelt (priority ordering: longest match fires, then explicit rule priority, then first defined rule fires)

Matching algorithms and Rule Priority

- Rules compete **within** a single phase (.jape file)
- styles of matching:
 - Brill (fire every rule that applies)
 - First (shortest rule fires)
 - Appelt (use of priorities)
 - Once (as soon as a rule fires, matching stops)
- Appelt priority is applied in the following order
 - Longest pattern
 - Explicit priority (default = -1)
 - First defined rule

JAPE Application Modes



Using phases

- Grammars usually consist of several phases, run sequentially
- A definition phase (conventionally called *main.jape*) lists the phases to be used, in order
- Only the definition phase needs to be loaded
- Temporary annotations may be created in early phases and used as input for later phases
- Annotations from earlier phases may need to be combined or modified

Coreference Resolution

- Name coreference
 - matches *similar* names in text, e.g. “Dr. Jacob Smith” and “Smith”
 - creates a ‘matches’ annotation which allows you to extract a chain of ‘equivalent’ names
- Pronominal coreference
 - solves references to named entities of pronouns in English (tokens marked with POS category PRP or PRP\$)

Coreference Resolution

- Orthographic co-reference can improve NE results by assigning entity type to previously unclassified names, based on relations with classified NEs
- May not reclassify already classified entities
- Classification of unknown entities very useful for surnames which match a full name, or abbreviations, e.g. "Bonfield" will match "Sir Peter Bonfield"

ANNIE System

- A Nearly New Information Extraction System
 - recognizes named entities in text
 - “packed” application combining/sequencing the following components: document reset, tokeniser, splitter, tagger, gazetteer lookup, NE grammars, name coreference
 - can be used as starting point to develop a new named recogniser

Some NE Annotated Corpora

- MUC-6 and MUC-7 corpora - English
- CONLL shared task corpora
<http://cnts.uia.ac.be/conll2003/ner/> - NEs in English and German
<http://cnts.uia.ac.be/conll2002/ner/> - NEs in Spanish and Dutch
- TIDES surprise language exercise (NEs in Cebuano and Hindi)
- ACE – English -
<http://www ldc.upenn.edu/Projects/ACE/>

The MUC-7 corpus

- 100 documents in SGML
- News domain
- 1880 Organizations (46%)
- 1324 Locations (32%)
- 887 Persons (22%)
- Inter-annotator agreement very high (~97%)
- http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_proceedings/marsh_slides.pdf

The MUC-7 Corpus (2)

<ENAMEX TYPE="LOCATION">CAPE CANAVERAL</ENAMEX>,
<ENAMEX TYPE="LOCATION">Fla.</ENAMEX> &MD;
Working in chilly temperatures <TIMEX
TYPE="DATE">Wednesday</TIMEX> <TIMEX
TYPE="TIME">night</TIMEX>, <ENAMEX
TYPE="ORGANIZATION">NASA</ENAMEX> ground crews
readied the space shuttle Endeavour for launch on a Japanese
satellite retrieval mission.

<p>

Endeavour, with an international crew of six, was set to blast off
from the <ENAMEX
TYPE="ORGANIZATION|LOCATION">Kennedy Space
Center</ENAMEX> on <TIMEX
TYPE="DATE">Thursday</TIMEX> at <TIMEX
TYPE="TIME">4:18 a.m. EST</TIMEX>, the start of a 49-
minute launching period. The <TIMEX TYPE="DATE">nine
day</TIMEX> shuttle flight was to be the 12th launched in
darkness.

Performance Evaluation

- Evaluation metric – mathematically defines how to measure the system's performance against a human-annotated, gold standard
- Scoring program – implements the metric and provides performance measures
 - For each document and over the entire corpus
 - For each type of NE

The Evaluation Metric

- Precision = correct answers/answers produced
- Recall = correct answers/total possible correct answers
- Trade-off between precision and recall
- F-Measure = $(\beta^2 + 1)PR / \beta^2R + P$
[van Rijsbergen 75]
- β reflects the weighting between precision and recall, typically $\beta=1$

The GATE Evaluation Tool

Annotation Diff Tool

Select the KEY doc: ft-airlines

Select the KEY annotation set: Key

Select the RESPONSE doc: ft-airlines

Select the RESPONSE annot set: Default set

Select annot. type: Location

Features: ☐ All ☐ Some ☒ None

String - Key	Start - Key	End - Key	String - Response	Start - Response	End - Response
England	2358	2365	England	2358	2365
UK	258	260	UK	258	260
Hampshire	2638	2647			
Swanwick	2886	2894			
Europe	746	752	Europe	746	752
Wales	2370	2375	Wales	2370	2375
UK	2801	2803	UK	2801	2803
Swanwick	2628	2636			
UK	931	933	UK	931	933

Precision strict: 1.0000 Recall strict: 0.6667 F-Measure strict: 0.8000
 Precision average: 1.0000 Recall average: 0.6667 F-Measure average: 0.8000
 Precision lenient: 1.0000 Recall lenient: 0.6667 F-Measure lenient: 0.8000

Regression Testing

- Need to track system's performance over time
- When a change is made to the system we want to know what implications are over the entire corpus
- Why: because an improvement in one case can lead to problems in others
- GATE offers automated tool to help with the NE development task over time

Document Indexing

- Indexing with Lucene
 - Populate a corpus
 - Create a Data Store (java serialisation)
 - Save corpus to DS
 - the corpus become “indexable” by Lucene
 - Index the corpus using document content
 - To search use the Information Retrieval plug-in
 - SearchPR processing resource (put it in a “pipeline”)
 - Specify parameters of search (corpus, query, etc.) and run
 - double clicking on SearchPR displays the results

Annotations in Context (ANNIC)

- Create a linguistic/semantic index
 - Create a Lucene Searchable DS
 - Populate a corpus
 - Apply ANNIE to the corpus
 - Save corpus to DS
 - Search in Context in the DS GUI

Machine Learning Approach

- Given a corpus annotated with named entities we want to create a classifier which decides if a string of text is a NE or not
 - ...<person>Mr. John Smith</person>...
 - ...<date>16th May 2005</date>
- The problem of recognising NEs can be seen as a classification problem

Machine Learning Approach

- Each named entity instance is transformed for the learning problem
 - ...<person>Mr. John Smith</person>...
 - Mr. is the beginning of the NE person
 - Smith is the end of the NE person
- The problem is transformed in a binary classification problem
 - is token begin of NE person?
 - is token end of NE person?
- Context is used as features for the classifier

Parsing with SUPPLE (Gaizauskas&al'05)

- Sheffield University Prolog Parser for Language Engineering
- A bottom-up parser for English which produces syntactic and semantic sentence representations
- An attribute-value context-free grammar of English is used to derive syntactic representations (it includes a question grammar for QA applications)
 - categories in the grammar have attributes and values which can be instantiated during parsing

Parsing with SUPPLE

- The grammar covers the following constituents
 - prepositional phrases; noun phrases; core verbs; verb phrases; relative clauses; sentences; questions
- The input to the parsing process is a 'chart' where both lexical items and multiword expressions (named entities) are allowed
- The output is the best possible 'parse' of the sentence, this can be partial

Parsing with SUPPLE

- Semantics is constructed compositionally as the sentence is parsed
- nouns and verbs are represented as normalised unary predicates (cat, eat, etc.)
- Identifiers (ei) are used to refer to an entity or an event and are produced for each noun and verb
 - cat(e1), eat(e2)
- binary predicates represent relations or attribute values of the entities or events; they are a fixed inventory used to represent grammatical and semantic relations
 - lsubj(X,Y), lobj(X,Z), of(X,Y), name(X,Z),...

Parsing with SUPPLE

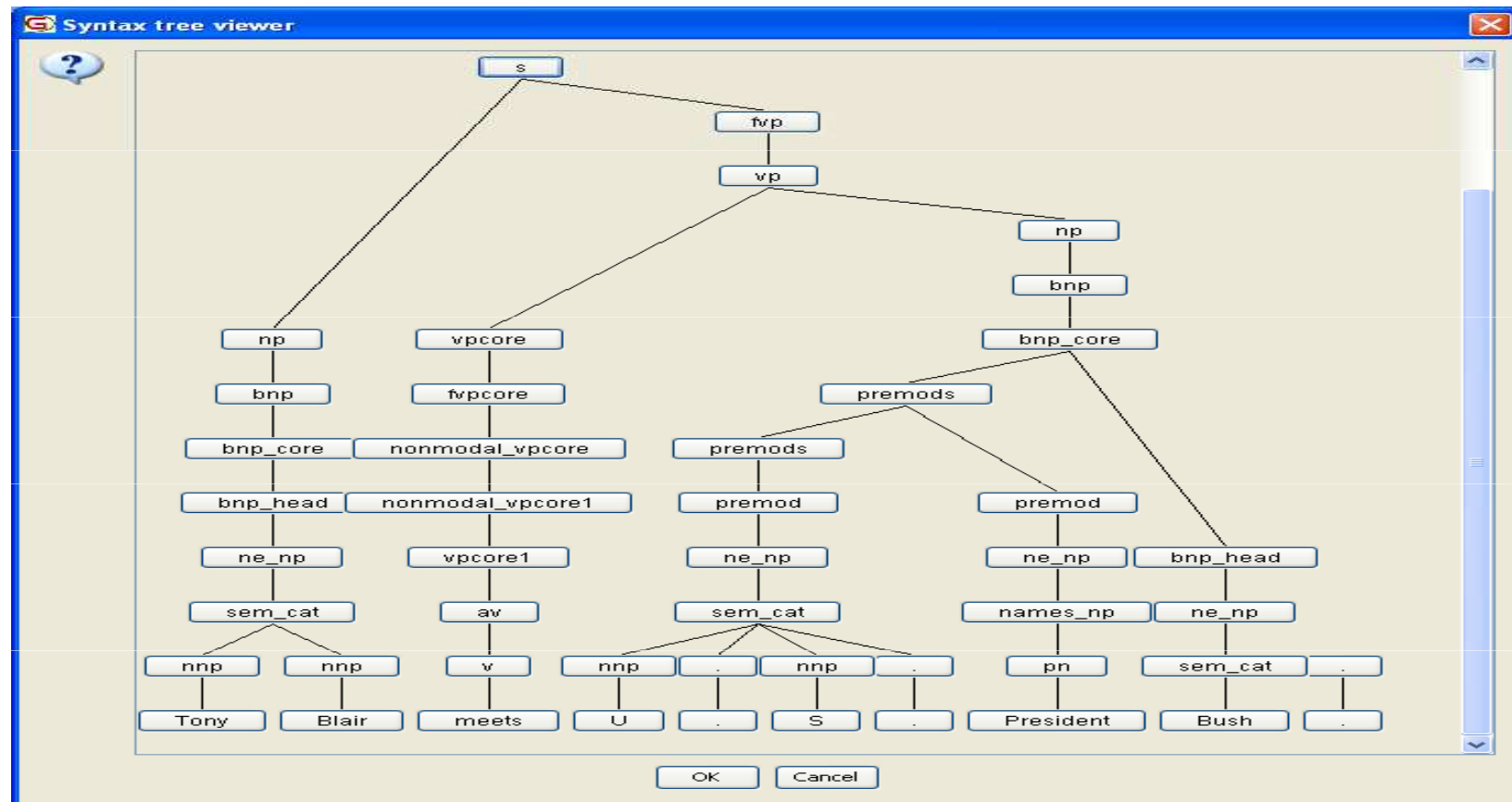
- Example

- Tony Blair meets U.S. President Bush.
 - identifies Tony Blair and Bush as Person type and U.S. is a Location type
 - wraps those constituents so that SUPPLE does not have to analyse them
 - rest of elements in the sentence are passed as words with POS, roots, number, gender, etc.

- Syntactic Annotation (string)

```
best_parse=( s ( np ( bnp ( bnp_core (
bnp_head ( ne_np ( sem_cat "Tony Blair" ) )
) ) ) ) ( fvp ( vp ( vpcore ( fvpcore (
nonmodal_vpcore ( nonmodal_vpcore1 (
vpcore1 ( av ( v "meets" ) ) ) ) ) ) ) ( np (
bnp ( bnp_core ( premods ( premods (
premod ( ne_np ( sem_cat "U.S." ) ) ) ) (
premod ( ne_np ( names_np ( pn "President"
) ) ) ) ) ( bnp_head ( ne_np ( sem_cat "Bush"
) ) ) ) ) ) ) )
```

Parsing with SUPPLE



Parsing with SUPPLE

- Semantic Annotation (array of strings)
qlf=[name(e2,'Tony Blair'), person(e2),
realisation(e2,offsets(0,10)), meet(e1),
time(e1,present), aspect(e1,simple),
voice(e1,active), lobj(e1,e3), name(e3,'Bush'),
person(e3), name(e4,'U.S.'), location(e4),
country(e4), realisation(e4,offsets(17,21)),
qual(e3,e4), ne_tag(e5,offsets(22,31)),
name(e5,'President'),
realisation(e5,offsets(22,31)), qual(e3,e5),
realisation(e3,offsets(17,36)),
realisation(e1,offsets(11,36)), lsubj(e1,e2)]

Parsing with SUPPLE

- A wrapper is provided in GATE
 - given a text which has been POS-tagged and Morphologically analysed, maps the tokens in each sentence to the input expected by SUPPLE
 - read the syntactic and semantic information from files and stores the information into the GATE documents as
 - parse, semantics, syntax tree nodes
- Can be run with SICStus prolog, SWI prolog, and PrologCafe (Java implementation)