Introduction to Text Mining

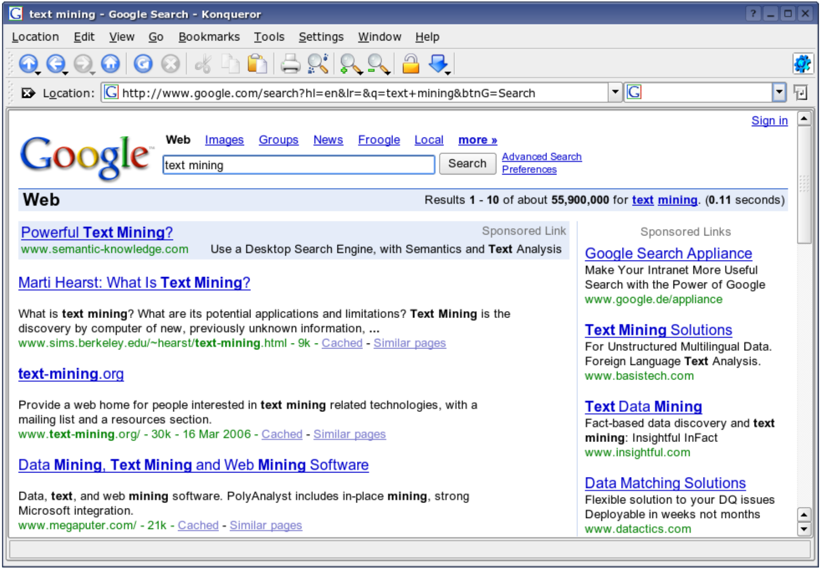
—Tutorial at EDBT'06—

René Witte

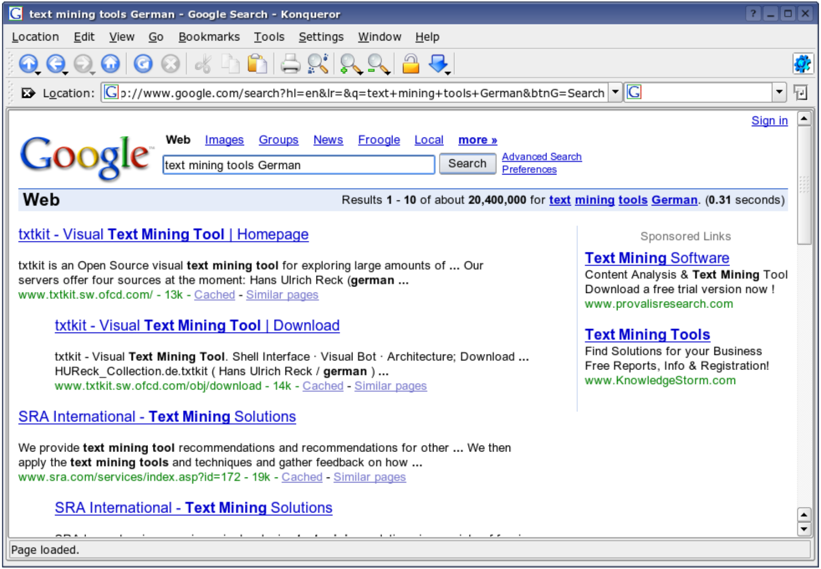
Faculty of Informatics
Institute for Program Structures and Data Organization (IPD)
Universität Karlsruhe, Germany
http://rene-witte.net

27.03.2006

Lack of Information?



Lack of Information?



Tutorial Overview

Today's Tutorial contains...

Introduction: Motivation, definitions, applications

Foundations: Theoretical background in Computational Linguistics

Technology: Technological foundations for building Text Mining systems

Applications: In-depth description of two application areas (summarization, biology) and overview on two others (question-answering, opinion mining)

Conclusions: the end.

Each part contains some references for further study.

Part I

Introduction

Information Overload

Too much (textual) information

- We now have electronic books, documents, web pages, emails, blogs, news, chats, memos, research papers, . . .
- . . . all of it immediately accessible, thanks to databases and Information Retrieval (IR)
- An estimated 80–85% of all data stored in databases are natural language texts
- But humans did not scale so well. . .

This results in the common perception of Information Overload.

Example: The BioTech Industry

Access to information is a serious problem

- 80% of biological knowledge is **only** in reasearch papers
- finding the information you need is prohibitively expensive

Humans do not scale well

- if you read 60 research papers/week. . .
- . . .and 10% of those are interesting. . .
- . . .a scientist manages 6/week, or 300/year

This is not good enough

- MedLine adds more than 10 000 abstracts each *month*!
- Chemical Abstracts Registry (CAS) registers 4000 entities **each day**, 2.5 million in 2004 alone

[cf. Talk by Robin McEntire of GlaxoSmithKline at KBB'05]

Example: The BioTech Industry

Access to information is a serious problem

- 80% of biological knowledge is **only** in reasearch papers
- finding the information you need is prohibitively expensive

Humans do not scale well

- if you read 60 research papers/week. . .
- . . .and 10% of those are interesting. . .
- . . .a scientist manages 6/week, or 300/year

This is not good enough

- MedLine adds more than 10 000 abstracts each *month*!
- Chemical Abstracts Registry (CAS) registers 4000 entities **each day**, 2.5 million in 2004 alone

[cf. Talk by Robin McEntire of GlaxoSmithKline at KBB'05]

Example: The BioTech Industry

Access to information is a serious problem

- 80% of biological knowledge is **only** in reasearch papers
- finding the information you need is prohibitively expensive

Humans do not scale well

- if you read 60 research papers/week...
- ...and 10% of those are interesting...
- ...a scientist manages 6/week, or 300/year

This is not good enough

- MedLine adds more than 10 000 abstracts each *month*!
- Chemical Abstracts Registry (CAS) registers 4000 entities **each day**, 2.5 million in 2004 alone

[cf. Talk by Robin McEntire of GlaxoSmithKline at KBB'05]

Definitions

One usually distinguishes

- Information Retrieval
- Information Extraction
- Text Mining

Text Mining (Def. *Wikipedia*)

Text mining, also known as intelligent text analysis, text data mining or knowledge-discovery in text (KDT), refers generally to the process of extracting interesting and non-trivial information and knowledge from unstructured text. Text mining is a young interdisciplinary field which draws on information retrieval, data mining, machine learning, statistics and computational linguistics. As most information (over 80%) is stored as text, text mining is believed to have a high commercial potential value.

Definitions

One usually distinguishes

- Information Retrieval
- Information Extraction
- Text Mining

Text Mining (Def. *Wikipedia*)

Text mining, also known as intelligent text analysis, text data mining or knowledge-discovery in text (KDT), refers generally to the process of extracting interesting and non-trivial information and knowledge from unstructured text. Text mining is a young interdisciplinary field which draws on information retrieval, data mining, machine learning, statistics and computational linguistics. As most information (over 80%) is stored as text, text mining is believed to have a high commercial potential value.

Definitions

One usually distinguishes

- Information Retrieval
- Information Extraction
- Text Mining

Text Mining (Def. *Wikipedia*)

Text mining, also known as intelligent text analysis, text data mining or knowledge-discovery in text (KDT), refers generally to the process of extracting interesting and non-trivial information and knowledge from unstructured text. Text mining is a young interdisciplinary field which draws on information retrieval, data mining, machine learning, statistics and computational linguistics. As most information (over 80%) is stored as text, text mining is believed to have a high commercial potential value.

Definitions

One usually distinguishes

- Information Retrieval
- Information Extraction
- Text Mining

Text Mining (Def. *Wikipedia*)

Text mining, also known as intelligent text analysis, text data mining or knowledge-discovery in text (KDT), refers generally to the process of extracting interesting and non-trivial information and knowledge from unstructured text. Text mining is a young interdisciplinary field which draws on information retrieval, data mining, machine learning, statistics and computational linguistics. As most information (over 80%) is stored as text, text mining is believed to have a high commercial potential value.

What to mine?

Emails, Instant Messages, Blogs, ...

Look for:

- Entities (Persons, Companies, Organizations, ...)
- Events (Inventions, Offers, Attacks, ...)

Biggest existing system: ECHELON (UKUSA)

What to mine? (II)

News: Newspaper articles, Newswires, ...

Similar to last, but additionally:

- collections of articles (e.g., from different agencies, describing the same event)
- contrastive summaries (e.g., event described by U.S. newspaper vs. Arabic newspaper)
- also needs temporal analysis
- main problems: cross-language and cross-document analysis

Many publicily accessible systems, e.g. *Google News* or *Newsblaster*.

What to mine? (III)

(Scientific) Books, Papers, ...

- detect new trends in research
- automatic curation of research results in Bioinformatics

need to deal with highly specific language

Software Requirement Specifications, Documentation, ...

- extract requirements from software specification
- detect conflicts between source code and its documentation

Web Mining

extract and analyse information from web sites

- mine companies' web pages (detect new products & trends)
- mine Intranets (gather knowledge, find "illegal" content, ...)

problems: not simply plain text, also hyperlinks and hidden information ("deep web")

What to mine? (III)

(Scientific) Books, Papers, ...

- detect new trends in research
- automatic curation of research results in Bioinformatics

need to deal with highly specific language

Software Requirement Specifications, Documentation, ...

- extract requirements from software specification
- detect conflicts between source code and its documentation

Web Mining

extract and analyse information from web sites

- mine companies' web pages (detect new products & trends)
- mine Intranets (gather knowledge, find "illegal" content, ...)

problems: not simply plain text, also hyperlinks and hidden information ("deep web")

What to mine? (III)

(Scientific) Books, Papers, ...

- detect new trends in research
- automatic curation of research results in Bioinformatics

need to deal with highly specific language

Software Requirement Specifications, Documentation, ...

- extract requirements from software specification
- detect conflicts between source code and its documentation

Web Mining

extract and analyse information from web sites

- mine companies' web pages (detect new products & trends)
- mine Intranets (gather knowledge, find "illegal" content, ...)

problems: not simply plain text, also hyperlinks and hidden information ("deep web")

Typical Text Mining Tasks

Classification and Clustering

- Email Spam-Detection, Classification (Orders, Offers, . . .)
- Clustering of large document sets (`vivisimo.com`)
- Creation of topic maps (`www.leximancer.com`)

Web Mining

- Trend Mining, Opinion Mining, Novelty Detection
- Ontology Creation, Entity Tracking, Information Extraction

"Classical" NLP Tasks

- Machine Translation (MT)
- Automatic Summarization
- Question-Answering (QA)

Typical Text Mining Tasks

Classification and Clustering

- Email Spam-Detection, Classification (Orders, Offers, . . .)
- Clustering of large document sets (`vivisimo.com`)
- Creation of topic maps (`www.leximancer.com`)

Web Mining

- Trend Mining, Opinion Mining, Novelty Detection
- Ontology Creation, Entity Tracking, Information Extraction

"Classical" NLP Tasks

- Machine Translation (MT)
- Automatic Summarization
- Question-Answering (QA)

Typical Text Mining Tasks

Classification and Clustering

- Email Spam-Detection, Classification (Orders, Offers, ...)
- Clustering of large document sets (`vivisimo.com`)
- Creation of topic maps (`www.leximancer.com`)

Web Mining

- Trend Mining, Opinion Mining, Novelty Detection
- Ontology Creation, Entity Tracking, Information Extraction

"Classical" NLP Tasks

- Machine Translation (MT)
- Automatic Summarization
- Question-Answering (QA)

Information Overload, Part II

Can't you just summarize this for me?

Create "intelligent assistants" that retrieve, process, and condense information for you.

We already have: Information Retrieval

We need: Technologies to process the retrieved information

One example is Automatic Summarization to condense a single document or a set of documents.

For example. . .

Mrs. Coolidge: *What did the preacher discuss in his sermon?*

President Coolidge: *Sin.*

Mrs. Coolidge: *What did he say?*

President Coolidge: *He said he was against it.*

Information Overload, Part II

Can't you just summarize this for me?

Create "intelligent assistants" that retrieve, process, and condense information for you.

We already have: Information Retrieval

We need: Technologies to process the retrieved information

One example is Automatic Summarization to condense a single document or a set of documents.

For example. . .

Mrs. Coolidge: *What did the preacher discuss in his sermon?*

President Coolidge: *Sin.*

Mrs. Coolidge: *What did he say?*

President Coolidge: *He said he was against it.*

Automatic Summarization

Example source (newspaper article)

HOUSTON – The Hubble Space Telescope got smarter and better able to point at distant astronomical targets on Thursday as spacewalking astronauts replaced two major pieces of the observatory's gear. On the second spacewalk of the shuttle Discovery's Hubble repair mission, the astronauts, C. Michael Foale and Claude Nicollier, swapped out the observatory's central computer and one of its fine guidance sensors, a precision pointing device. The spacewalkers ventured into Discovery's cargo bay, where Hubble towers almost four stories above, at 2:06 p.m. EST, about 45 minutes earlier than scheduled, to get a jump on their busy day of replacing some of the telescope's most important components. . . .

Summary (10 words)

Space News: [the shuttle Discovery's Hubble repair mission,
the observatory's central computer]

Automatic Summarization

Example source (newspaper article)

HOUSTON – The Hubble Space Telescope got smarter and better able to point at distant astronomical targets on Thursday as spacewalking astronauts replaced two major pieces of the observatory's gear. On the second spacewalk of the shuttle Discovery's Hubble repair mission, the astronauts, C. Michael Foale and Claude Nicollier, swapped out the observatory's central computer and one of its fine guidance sensors, a precision pointing device. The spacewalkers ventured into Discovery's cargo bay, where Hubble towers almost four stories above, at 2:06 p.m. EST, about 45 minutes earlier than scheduled, to get a jump on their busy day of replacing some of the telescope's most important components. . . .

Summary (10 words)

Space News: [the shuttle Discovery's Hubble repair mission,
the observatory's central computer]

Dealing with Text in Natural Languages

Problem

How can I automatically create a summary from a text written in natural language?

Solution: Natural Language Processing (NLP)

Current trends in NLP:

- deal with "real-world" texts, not just limited examples
- requires robust, fault-tolerant algorithms (e.g., partial parsing)
- shift from rule-based approches to statistical methods and machine learning
- focus on "knowledge-poor" techniques, as even shallow semantics is quite tough to obtain

Dealing with Text in Natural Languages

Problem

How can I automatically create a summary from a text written in natural language?

Solution: Natural Language Processing (NLP)

Current trends in NLP:

- deal with "real-world" texts, not just limited examples
- requires robust, fault-tolerant algorithms (e.g., partial parsing)
- shift from rule-based approches to statistical methods and machine learning
- focus on "knowledge-poor" techniques, as even shallow semantics is quite tough to obtain

Part II

Foundations

Take your PP-Attachement out of my Garden Path!

Understanding Computational Linguists

Text Mining is concerned with processing documents written in natural language:

- this is the domain of *Computational Linguistics (CL)* and *Natural Language Processing* (NLP)
- practical application, with more of an engineering perspective, also called *Language Technology (LT)*
- Text Mining (TM) is concerned with concrete practical applications (compare: “Information Systems” and “Databases”)

Hence, we need to review some concepts, terminology, and foundations from these areas.

Computational Linguistics 101

Classical Categorization

To deal with the complexity of natural langauge, it is typically regarded on several levels (cf. Jurafsky & Martin):

- **Phonology** the study of linguistic sounds
- **Morphology** the study of meaningful components of words
- **Syntax** the study of structural relationships between words
- **Semantics** the study of meaning
- **Pragmatics** the study of how language is used to accomplish goals
- **Discourse** the study of larger linguistic units

Importance for Text Mining

- *Phonology* only concerns spoken language
- *Discourse*, *Pragmatics*, and even *Semantics* is still rarely used

Why is NLP hard?

Difference to other areas in Computer Science

Computer scientist are used to dealing with precise, closed, artificial structures

- e.g., we build a “mini-world” for a database rather than attempting to model every aspect of the real world
- programming languages have a simple syntax (around 100 words) and a precise semantic

This approach does not work for natural language:

- tens of thousands of languages, with more than 100 000 words each
- complex syntax, many ambiguities, constantly changing and evolving

A corollary is that a TM system will never get it “100% right”

Ambiguity

Ambiguity appears on every analysis level

The classical examples:

- *He saw the man with the telescope.*
- *Time flies like an arrow. Fruit flies like a banana.*

And those are simple. . .

This does not get better with real-world sentences:

- *The board approved [its acquisition] [by Royal Trustco. Ltd.] [of Toronto] [for $27 a share] [at its monthly meeting].*

(cf. Manning & Schütze)

Ambiguity

Ambiguity appears on every analysis level

The classical examples:

- *He saw the man with the telescope.*
- *Time flies like an arrow. Fruit flies like a banana.*

And those are simple...

This does not get better with real-world sentences:

- *The board approved [its acquisition] [by Royal Trustco. Ltd.] [of Toronto] [for $27 a share] [at its monthly meeting].*

(cf. Manning & Schütze)

Current Trends in NLP

The classical way: until late 1980's

Rule-based approaches:

- are too rigid for natural language
- suffer from the *knowledge acquisition bottleneck*
- cannot keep up with changing/evolving language
  ex. "to google"

The statistical way: since early 1990's

"Statistical NLP" refers to all quantitative approaches, including Bayes' models, Hidden Markov Models (HMMs), Support Vector Machines (SVMs), Clustering, ...

- more robust & more flexible
- need a *Corpus* for (supervised or unsupervised) learning

But real-world systems typically combine both.

Tokenization

Preprocessing

Input files usually need some cleanup before processing can start:

- Remove "fluff" from web pages (ads, navigation bars, ...)
- Normalize text converted from PDF, Doc, or other binary formats
- Deal with errors in OCR'd documents
- Deal with tables, figures, captions, formulas, ...

Tokenization

Text is splitted into basic units called *Tokens*:

- word tokens
- number tokens
- space tokens
- ...

Consistent tokenization is important for all later processing steps

Tokenization

Preprocessing

Input files usually need some cleanup before processing can start:

- Remove “fluff” from web pages (ads, navigation bars, . . .)
- Normalize text converted from PDF, Doc, or other binary formats
- Deal with errors in OCR’d documents
- Deal with tables, figures, captions, formulas, . . .

Tokenization

Text is splitted into basic units called *Tokens*:

- word tokens
- number tokens
- space tokens
- . . .

Consistent tokenization is important for all later processing steps

Tokenization (II)

What is a word?

Unfortunately, even tokenization can be difficult:

- Is *John's sick* one token or two?
  If one → problems in parsing (where's the verb?)
  If two → what do we do with *John's house?*
- What to do with hyphens?
  E.g., *database* vs. *data-base* vs. *data base*
- what to do with "C++", "A/C", ":-)", "..."?

Even worse...

- Some languages don't use whitespace (e.g., Chinese)
  → need to run a *word segmentation* first
- Heavy compounding e.g. in German, decomposition necessary
  "Rinderbraten" (roast beef) → *Rind|erbraten?*
  Rind|erb|raten? Rinder|braten?

Tokenization (II)

What is a word?

Unfortunately, even tokenization can be difficult:

- Is *John's sick* one token or two?
  If one → problems in parsing (where's the verb?)
  If two → what do we do with *John's house?*
- What to do with hyphens?
  E.g., *database* vs. *data-base* vs. *data base*
- what to do with "C++", "A/C", ":-)", "..."?

Even worse...

- Some languages don't use whitespace (e.g., Chinese)
  → need to run a *word segmentation* first
- Heavy compounding e.g. in German, decomposition necessary
  "Rinderbraten" (roast beef) → *Rind|erbraten*?
  Rind|erb|raten? *Rinder|braten*?

Tokenization (III)

The good, the bad, and the . . .

Tokenization can become even more difficult in specific domains.

Software Documents

Documents include lots of source code snippets:

- `package java.util.*`
- *The range-view operation, subList(int fromIndex, int toIndex), returns a List view of the portion of this list whose indices range from fromIndex, inclusive, to toIndex , exclusive.*

Need to deal with URLs, methods, class names, etc.

Tokenization (III)

The good, the bad, and the . . .

Tokenization can become even more difficult in specific domains.

Software Documents

Documents include lots of source code snippets:

- package `java.util.*`
- *The range-view operation, subList(int fromIndex, int toIndex), returns a List view of the portion of this list whose indices range from fromIndex, inclusive, to toIndex , exclusive.*

Need to deal with URLs, methods, class names, etc.

Tokenization (IV)

Biological Documents

Highly complex expressions, chemical formulas, etc.:

- *1,4-β-xylanase II from Trichoderma reesei*
- *When N-formyl-L-methionyl-L-leucyl-L-phenylalanine (fMLP) was injected...*
- *Technetium-99m-CDO-MeB [Bis[1,2-cyclohexanedione-dioximato(1-)-O]-[1,2-cyclohexanedione dioximato(2-) -O]methyl-borato(2-)-N,N′,N″,N‴,N⁗,N′′′′′)-chlorotechnetium) belongs to a family of compounds...*

Sentence Splitting

Mark Sentence Boundaries

Detects sentence units. Easy case:

- often, sentences end with ".", "!", or "?"

Hard (or annoying) cases:

- difficult when a "." do not indicate an EOS:
  "MR. X", "3.14", "Y Corp.", . . .
- we can detect common abbreviations ("U.S."), but what if a sentence ends with one?
  ". . .announced today by the U.S. The. . .
- Sentences can be *nested* (e.g., within quotes)

Correct sentence boundary is important

for many downstream analysis tasks:

- POS-Taggers maximize probabilites of tags within a sentence
- Summarization systems rely on correct detection of sentence

Sentence Splitting

Mark Sentence Boundaries

Detects sentence units. Easy case:

- often, sentences end with ".", "!", or "?"

Hard (or annoying) cases:

- difficult when a "." do not indicate an EOS:
  "MR. X", "3.14", "Y Corp.", ...
- we can detect common abbreviations ("U.S."), but what if a sentence ends with one?
  "...announced today by the U.S. The...
- Sentences can be *nested* (e.g., within quotes)

Correct sentence boundary is important

for many downstream analysis tasks:

- POS-Taggers maximize probabilites of tags within a sentence
- Summarization systems rely on correct detection of sentence

Morphological Analysis

Morphological Variants

Words are changed through a morphological process called *inflection*:

- typically indicates changes in case, gender, number, tense, etc.
- example *car* $\longrightarrow$ cars, *give* $\longrightarrow$ *gives, gave, given*

Goal: "normalize" words

Stemming and Lemmatization

Two main approaches to normalization:

Stemming reduce words to a *base form*

Lemmatization reduce words to their *lemma*

Main difference: stemming just finds **any** base form, which doesn't even need to be a word in the language! Lemmatization find the actual *root* of a word, but requires morphological analysis.

Morphological Analysis

Morphological Variants

Words are changed through a morphological process called *inflection*:

- typically indicates changes in case, gender, number, tense, etc.
- example *car* $\longrightarrow$ cars, *give* $\longrightarrow$ *gives, gave, given*

Goal: "normalize" words

Stemming and Lemmatization

Two main approaches to normalization:

Stemming reduce words to a *base form*

Lemmatization reduce words to their *lemma*

Main difference: stemming just finds **any** base form, which doesn't even need to be a word in the language! Lemmatization find the actual *root* of a word, but requires morphological analysis.

Stemming vs. Lemmatization

Stemming

Commonly used in Information Retrieval:

- Can be achieved with rule-based algorithms, usually based on suffix-stripping
- Standard algorithm for English: the *Porter* stemmer
- Advantages: simple & fast
- Disadvantages:
  - Rules are language-dependent
  - Can create words that do not exist in the language, e.g., *computers* → *comput*
  - Often reduces different words to the same stem, e.g., *army, arm* → *arm* *stocks, stockings* → *stock*
- Stemming for German: German stemmer in the full-text search engine *Lucene*, *Snowball* stemmer with German rule file

Stemming vs. Lemmatization, Part II

Lemmatization

Lemmatization is the process of deriving the base form, or *lemma*, of a word from one of its inflected forms. This requires a morphological analysis, which in turn typically requires a *lexicon*.

- Advantages:
  - identifies the *lemma* (root form), which is an actual word
  - less errors than in stemming
- Disadvantages:
  - more complex than stemming, slower
  - requires additional language-dependent resources
- While stemming is good enough for Information Retrieval, Text Mining often requires lemmatization
  - Semantics is more important (we need to distinguish an *army* and an *arm*!)
  - Errors in low-level components can multiply when running downstream

Lemmatization Example

Lemmatization in German

Lemmatization for a morphologically complex language like German is complicated

- Cannot be solved through a rule-based algorithm

Kinder ⟶ *Kind*	*Vorlesungen* ⟶ *Vorlesung*	*Länder* ⟶ *Land*
Leiter ⟶ **Leit*	*Leben* ⟶ **Leb*	*Affären* ⟶ **Affare*

- An accurate lemmatization for German requires a lexicon
  - For each word, all inflected forms or morphological rules

The *Durm* German Lemmatizer

A self-learning context-aware lemmatization system for German that can create (and correct) a lexicon by processing German documents:

```
Menschen    Sg   Masc   Akk    Mensch  6   4/11/2005 15:8:16
                                4/11/2005 15:10:11   116  unlocked
```

Part-of-Speech (POS) Tagging

Where are we now?

So far, we splitted texts into *tokens* and *sentences* and performed some *normalization*.

- Still a long way to go to an *understanding* of natural language...

Typical approach in NLP: deal with the complexity of language by applying intermediate processing steps to acquire more and more structure. Next stop: *POS-Tagging*.

POS-Tagging

A statistical POS Tagger scans tokens and assigns POS Tags.

A black cat plays... → A/DT black/JJ cat/NN plays/VB...

- relies on different word order probabilities
- needs a manually tagged corpus for machine learning

Note: *this is not parsing!*

Part-of-Speech (POS) Tagging

Where are we now?

So far, we splitted texts into *tokens* and *sentences* and performed some *normalization*.

- Still a long way to go to an *understanding* of natural language...

Typical approach in NLP: deal with the complexity of language by applying intermediate processing steps to acquire more and more structure. Next stop: *POS-Tagging*.

POS-Tagging

A statistical POS Tagger scans tokens and assigns POS Tags.

A black cat plays... → A/DT black/JJ cat/NN plays/VB...

- relies on different word order probabilities
- needs a manually tagged corpus for machine learning

Note: *this is not parsing!*

Part-of-Speech (POS) Tagging (II)

Tagsets

A tagset defines the tags to assign to words. Main POS classes are:

Noun	refers to entities like people, places, things or ideas
Adjective	describes the properties of nouns or pronouns
Verb	describes actions, activities and states
Adverb	describes a verb, an adjective or another adverb
Pronoun	word that can take the place of a noun
Determiner	describes the particular reference of a noun
Preposition	expresses spatial or time relationships

Note: real tagsets have from 45 (Penn Treebank) to 146 tags (C7).

POS Tagging Algorithms

Fundamentals

POS-Tagging generally requires:

Training phase where a manually annotated corpus is processed by a machine learning algorithm; and a

Tagging algorithm that processes texts using learned parameters.

Performance is generally good (around 96%) when staying in the same domain.

Algorithms used in POS-Tagging

There is a multitude of approaches, commonly used are:

- Decision Trees
- Hidden Markov Models (HMMs)
- Support Vector Machines (SVM)
- Transformation-based Taggers (e.g., the Brill tagger)

POS Tagging Algorithms

Fundamentals

POS-Tagging generally requires:

Training phase where a manually annotated corpus is processed by a machine learning algorithm; and a

Tagging algorithm that processes texts using learned parameters.

Performance is generally good (around 96%) when staying in the same domain.

Algorithms used in POS-Tagging

There is a multitude of approaches, commonly used are:

- Decision Trees
- Hidden Markov Models (HMMs)
- Support Vector Machines (SVM)
- Transformation-based Taggers (e.g., the Brill tagger)

Syntax: Chunking and Parsing

Finding Syntactic Structures

We can now start a syntactic analysis of a sentence using:

Parsing producing a *parse tree* for a sentence using a parser, a grammar, and a lexicon

Chunking finding syntactic constituents like *Noun Phrases (NPs)* or *Verb Groups (VGs)* within a sentence

Chunking vs. Parsing

Producing a *full parse tree* often fails due to grammatical inaccuracies, novel words, bad tokenization, wrong sentence splits, errors in POS tagging, . . .

Hence, *chunking* and *partial parsing* are more commonly used.

Syntax: Chunking and Parsing

Finding Syntactic Structures

We can now start a syntactic analysis of a sentence using:

Parsing producing a *parse tree* for a sentence using a parser, a grammar, and a lexicon

Chunking finding syntactic constituents like *Noun Phrases (NPs)* or *Verb Groups (VGs)* within a sentence

Chunking vs. Parsing

Producing a *full parse tree* often fails due to grammatical inaccuracies, novel words, bad tokenization, wrong sentence splits, errors in POS tagging, . . .
Hence, *chunking* and *partial parsing* are more commonly used.

Noun Phrase Chunking

NP Chunker

Recognition of noun phrases through context-free grammar with Earley-type chart parser

Example

Grammar Excerpt

```
(NP     (DET MOD HEAD))
(MOD    (MOD-ingredients)
        (MOD-ingredients MOD)
        ())
(HEAD   (NN) ...)
```

Noun Phrase Chunking

NP Chunker

Recognition of noun phrases through context-free grammar with Earley-type chart parser

Grammar Excerpt

```
(NP      (DET MOD HEAD))
(MOD     (MOD-ingredients)
         (MOD-ingredients MOD)
         ())
(HEAD    (NN) ...)
```

Example

Noun Phrase Chunking

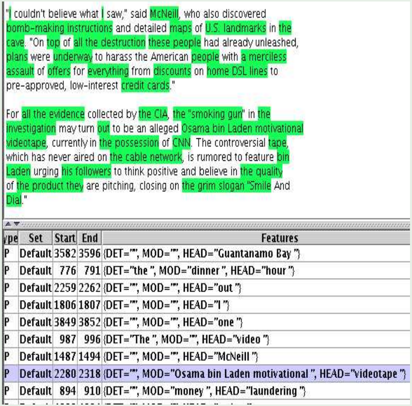
NP Chunker

Recognition of noun phrases through context-free grammar with Earley-type chart parser

Grammar Excerpt

```
(NP      (DET MOD HEAD))
(MOD     (MOD-ingredients)
         (MOD-ingredients MOD)
         ())
(HEAD    (NN) ...)
```

Example


"I couldn't believe what I saw," said McNeill, who also discovered bomb-making instructions and detailed maps of U.S. landmarks in the cave. "On top of all the destruction these people had already unleashed, plans were underway to harass the American people with a merciless assault of offers for everything from discounts on home DSL lines to pre-approved, low-interest credit cards."

For all the evidence collected by the CIA, the "smoking gun" in the investigation may turn out to be an alleged Osama bin Laden motivational videotape, currently in the possession of CNN. The controversial tape, which has never aired on the cable network, is rumored to feature bin Laden urging his followers to think positive and believe in the quality of the product they are pitching, closing on the grim slogan "Smile And Dial."

ype	Set	Start	End	Features
P	Default	3582	3596	{DET="", MOD="", HEAD="Guantanamo Bay "}
P	Default	776	791	{DET="the ", MOD="dinner ", HEAD="hour "}
P	Default	2259	2262	{DET="", MOD="", HEAD="out "}
P	Default	1806	1807	{DET="", MOD="", HEAD="I "}
P	Default	3849	3852	{DET="", MOD="", HEAD="one "}
P	Default	987	996	{DET="The ", MOD="", HEAD="video "}
P	Default	1487	1494	{DET="", MOD="", HEAD="McNeill "}
P	Default	2280	2318	{DET="", MOD="Osama bin Laden motivational ", HEAD="videotape "}
P	Default	894	910	{DET="", MOD="money ", HEAD="laundering "}

Chunking vs. Parsing, Round 2

What can we do with chunks?

(NP) chunks are very useful in finding named entities (NEs), e.g., *Persons, Companies, Locations, Patents, Organisms, . . .*.
But additional methods are needed for finding relations:

- *Who* invented X?
- *What* company created product Y that is doomed to fail?
- *Which* organism is this protein coming from?

Parse trees can help in determining these relationships

Parsing Challenges

Parsing is hard due to many kinds of ambiguities:

PP-Attachement which NP takes the PP? Compare:
He ate spaghetti with a fork.
He ate spaghetti with tomato sauce.

NP Bracketing *plastic cat food can cover*

Chunking vs. Parsing, Round 2

What can we do with chunks?

(NP) chunks are very useful in finding named entities (NEs), e.g., *Persons, Companies, Locations, Patents, Organisms*, ...
But additional methods are needed for finding relations:

- *Who* invented X?
- *What* company created product Y that is doomed to fail?
- *Which* organism is this protein coming from?

Parse trees can help in determining these relationships

Parsing Challenges

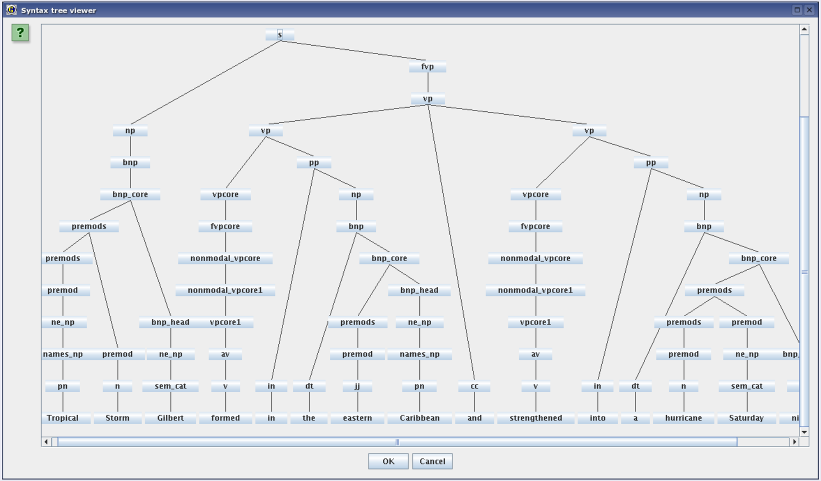
Parsing is hard due to many kinds of ambiguities:

PP-Attachement which NP takes the PP? Compare:
He ate spaghetti with a fork.
He ate spaghetti with tomato sauce.

NP Bracketing *plastic cat food can cover*

Parsing: Example

Example of a (partial) parser output using SUPPLE



Semantics

Moving on. . .

Now that we have syntactic information, we can start to address the *meaning* of words.

WordNets

A WordNet is a semantic network encoding the words of a single (or multiple) language(s) using:

- **Synsets** encoding the *meanings* for each word (e.g., *bank*)
- **Relations** synonymy, antonymy, hypernymy, hyponymy, holonymy, meronymy, homonymy, troponymy, . . .

The English WordNet currently encodes 147249 words (v2.1) and is freely available.

Example

Use WordNet to find out whether tea is something we can drink.

Semantics

Moving on...

Now that we have syntactic information, we can start to address the *meaning* of words.

WordNets

A WordNet is a semantic network encoding the words of a single (or multiple) language(s) using:

- **Synsets** encoding the *meanings* for each word (e.g., *bank*)
- **Relations** synonymy, antonymy, hypernymy, hyponymy, holonymy, meronymy, homonymy, troponymy, ...

The English WordNet currently encodes 147249 words (v2.1) and is freely available.

Example

Use WordNet to find out whether tea is something we can drink.

Semantics

Moving on...

Now that we have syntactic information, we can start to address the *meaning* of words.

WordNets

A WordNet is a semantic network encoding the words of a single (or multiple) language(s) using:

- **Synsets** encoding the *meanings* for each word (e.g., *bank*)
- **Relations** synonymy, antonymy, hypernymy, hyponymy, holonymy, meronymy, homonymy, troponymy, ...

The English WordNet currently encodes 147249 words (v2.1) and is freely available.

Example

Use WordNet to find out whether tea is something we can drink.

WordNet Example

Lookup for "tea"

WordNet Search - 2.1

Return to WordNet Home

Glossary - Help

SEARCH DISPLAY OPTIONS: (Select option to change) Change

Enter a word to search for: tea Search WordNet

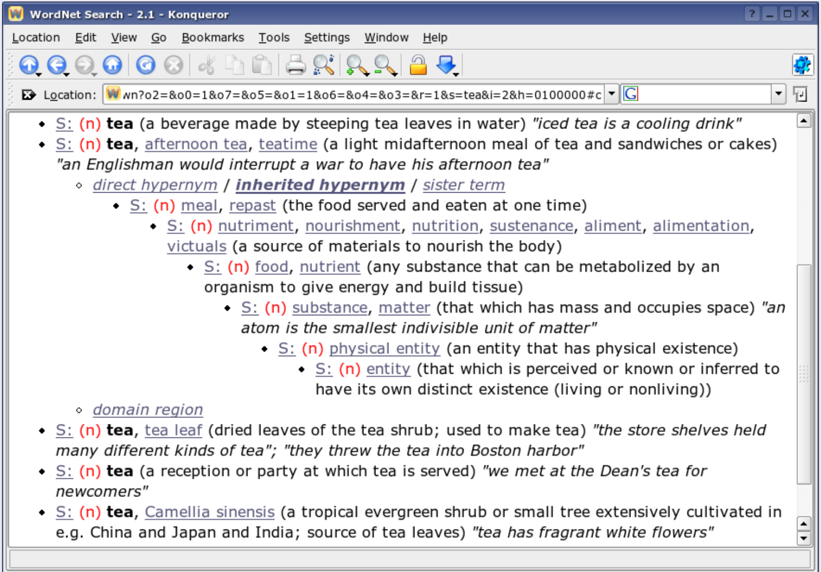
KEY: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Noun

- S: (n) **tea** (a beverage made by steeping tea leaves in water) *"iced tea is a cooling drink"*
- S: (n) **tea**, afternoon tea, teatime (a light midafternoon meal of tea and sandwiches or cakes) *"an Englishman would interrupt a war to have his afternoon tea"*
- S: (n) **tea**, tea leaf (dried leaves of the tea shrub; used to make tea) *"the store shelves held many different kinds of tea"; "they threw the tea into Boston harbor"*
- S: (n) **tea** (a reception or party at which tea is served) *"we met at the Dean's tea for newcomers"*
- S: (n) **tea**, Camellia sinensis (a tropical evergreen shrub or small tree extensively cultivated in

WordNet Example (II)

Hypernyms of "tea", Sense 2



WordNet Search - 2.1 - Konqueror

Location Edit View Go Bookmarks Tools Settings Window Help

Location: wn?o2=&o0=1&o7=&o5=&o1=1&o6=&o4=&o3=&r=1&s=tea&i=2&h=0100000#c

- S: (n) **tea** (a beverage made by steeping tea leaves in water) *"iced tea is a cooling drink"*
- S: (n) **tea**, afternoon tea, teatime (a light midafternoon meal of tea and sandwiches or cakes) *"an Englishman would interrupt a war to have his afternoon tea"*
  - *direct hypernym* / ***inherited hypernym*** / *sister term*
    - S: (n) meal, repast (the food served and eaten at one time)
      - S: (n) nutriment, nourishment, nutrition, sustenance, aliment, alimentation, victuals (a source of materials to nourish the body)
        - S: (n) food, nutrient (any substance that can be metabolized by an organism to give energy and build tissue)
          - S: (n) substance, matter (that which has mass and occupies space) *"an atom is the smallest indivisible unit of matter"*
            - S: (n) physical entity (an entity that has physical existence)
              - S: (n) entity (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))
  - *domain region*
- S: (n) **tea**, tea leaf (dried leaves of the tea shrub; used to make tea) *"the store shelves held many different kinds of tea"; "they threw the tea into Boston harbor"*
- S: (n) **tea** (a reception or party at which tea is served) *"we met at the Dean's tea for newcomers"*
- S: (n) **tea**, Camellia sinensis (a tropical evergreen shrub or small tree extensively cultivated in e.g. China and Japan and India; source of tea leaves) *"tea has fragrant white flowers"*

Logical Forms and Predicate-Argument Structures

Transforming Text into Logical Units

Suppose we found the correct sense for each word. We can now transform the text into a formal representation, e.g., first-oder predicate logic or description logics.

- knowledge is encoded independently from the textual description (e.g., "X bought A" and "A was acquired by X" both encode the same information)
- with this, formal reasoning becomes possible

Predicate-Argument Structures

Convert text into logical structures using predicates:

- $company(x_1) \wedge company(x_2) \wedge buy\text{-}act(x_1, x_2)$

PA structures can be derived from parse and additionally incorporate semantic information (e.g., using WordNet).

Logical Forms and Predicate-Argument Structures

Transforming Text into Logical Units

Suppose we found the correct sense for each word. We can now transform the text into a formal representation, e.g., first-oder predicate logic or description logics.

- knowledge is encoded independently from the textual description (e.g., "*X* bought *A*" and "*A* was acquired by *X*" both encode the same information)
- with this, formal reasoning becomes possible

Predicate-Argument Structures

Convert text into logical structures using predicates:

- $\mathit{company}(x_1) \wedge \mathit{company}(x_2) \wedge \mathit{buy\text{-}act}(x_1, x_2)$

PA structures can be derived from parse and additionally incorporate semantic information (e.g., using WordNet).

Pragmatics: Coreference Resolution

Problem

Entities in natural language texts are not identified with convenient unique IDs, but rather with constantly changing descriptions.
Example: *Mr. Bush, The president, he, George W., ...*

Solution

Automatic detection and collection of all textual descriptors that refer to the same entity within a *coreference chain*.

- can be used to find information about an entity, even when referenced by a different name
- important for many higher-level text analysis tasks

Coreference Resolution Algorithms

Pronomial coreferences can be detected quite reliably (also called *Anaphora Resolution*. Full (nominal) coreference resolution is hard.

Pragmatics: Coreference Resolution

Problem

Entities in natural language texts are not identified with convenient unique IDs, but rather with constantly changing descriptions.
Example: *Mr. Bush, The president, he, George W., . . .*

Solution

Automatic detection and collection of all textual descriptors that refer to the same entity within a *coreference chain*.

- can be used to find information about an entity, even when referenced by a different name
- important for many higher-level text analysis tasks

Coreference Resolution Algorithms

Pronomial coreferences can be detected quite reliably (also called *Anaphora Resolution*. Full (nominal) coreference resolution is hard.

Pragmatics: Coreference Resolution

Problem

Entities in natural language texts are not identified with convenient unique IDs, but rather with constantly changing descriptions.
Example: *Mr. Bush, The president, he, George W., . . .*

Solution

Automatic detection and collection of all textual descriptors that refer to the same entity within a *coreference chain*.

- can be used to find information about an entity, even when referenced by a different name
- important for many higher-level text analysis tasks

Coreference Resolution Algorithms

Pronomial coreferences can be detected quite reliably (also called *Anaphora Resolution*. Full (nominal) coreference resolution is hard.

Evaluation of NLP Systems

General Approach

The results of a system are compared to a manually created *gold standard* using various metrics.

Main Challenges

Manually annotating large amounts of texts for specific linguistic phenomena is **very** time-consuming (thus expensive):

- test set needs to be different from training set
- for some tasks, two or more annotations of the same data are needed (to measure *inter-annotator agreement*)

Annotated Corpora

For some tasks (e.g., POS tagging), annotated corpora are (freely) available.

Evaluation of NLP Systems

General Approach

The results of a system are compared to a manually created *gold standard* using various metrics.

Main Challenges

Manually annotating large amounts of texts for specific linguistic phenomena is **very** time-consuming (thus expensive):

- test set needs to be different from training set
- for some tasks, two or more annotations of the same data are needed (to measure *inter-annotator agreement*)

Annotated Corpora

For some tasks (e.g., POS tagging), annotated corpora are (freely) available.

Evaluation of NLP Systems

General Approach

The results of a system are compared to a manually created *gold standard* using various metrics.

Main Challenges

Manually annotating large amounts of texts for specific linguistic phenomena is **very** time-consuming (thus expensive):

- test set needs to be different from training set
- for some tasks, two or more annotations of the same data are needed (to measure *inter-annotator agreement*)

Annotated Corpora

For some tasks (e.g., POS tagging), annotated corpora are (freely) available.

Evaluation Measures

Accuracy and Error

Simplest measure are *accuracy* (percentage of correct results) and *error* (percentage of wrong results).

- not often used, as they are very insensitive to the interesting numbers
- reason is the usually large number of non-relevant and non-selected entities that is "hiding" all other numbers
- in other words, accuracy only reacts to real errors, and doesn't show how many correct results have been found as such

Precision and Recall

Precision

Like in Information Retrieval, *Precision* show the percentage of correct results within an answer:

$$\text{Precision} = \frac{\text{Correct} + \frac{1}{2}\text{Partial}}{\text{Correct} + \text{Spurious} + \frac{1}{2}\text{Partial}}$$

Recall

And *Recall* the percentage of the correct system results over all correct results:

$$\text{Recall} = \frac{\text{Correct} + \frac{1}{2}\text{Partial}}{\text{Correct} + \text{Missing} + \frac{1}{2}\text{Partial}}$$

Tradeoff

Note that you can always get 100% Precision by selecting nothing and 100% Recall by selecting everything. However, in NLP there is often no clear trade-off between the two.

Precision and Recall

Precision

Like in Information Retrieval, *Precision* show the percentage of correct results within an answer:

$$\text{Precision} = \frac{\text{Correct} + \frac{1}{2}\text{Partial}}{\text{Correct} + \text{Spurious} + \frac{1}{2}\text{Partial}}$$

Recall

And *Recall* the percentage of the correct system results over all correct results:

$$\text{Recall} = \frac{\text{Correct} + \frac{1}{2}\text{Partial}}{\text{Correct} + \text{Missing} + \frac{1}{2}\text{Partial}}$$

Tradeoff

Note that you can always get 100% Precision by selecting nothing and 100% Recall by selecting everything. However, in NLP there is often no clear trade-off between the two.

Precision and Recall

Precision

Like in Information Retrieval, *Precision* show the percentage of correct results within an answer:

$$\text{Precision} = \frac{\text{Correct} + \frac{1}{2}\text{Partial}}{\text{Correct} + \text{Spurious} + \frac{1}{2}\text{Partial}}$$

Recall

And *Recall* the percentage of the correct system results over all correct results:

$$\text{Recall} = \frac{\text{Correct} + \frac{1}{2}\text{Partial}}{\text{Correct} + \text{Missing} + \frac{1}{2}\text{Partial}}$$

Tradeoff

Note that you can always get 100% Precision by selecting nothing and 100% Recall by selecting everything. However, in NLP there is often no clear trade-off between the two.

F-Measure and IAA

Combining Precision and Recall

Often a combined measure of Precision and Recall is helpful. This can be done using the *F-Measure* (equal weight for $\beta = 1$):

$$\text{F-measure} = \frac{(\beta^2 + 1)P \cdot R}{(\beta^2 R) + P}$$

Measuring Inter-Annotator Agreement

There are many measures for computing IAA (Cohen's Kappa, prevalence, bias, ...), depending on the concrete task. On way to obtain the IAA is to compute P, R, and F values between two humans and averaging the results of $P(H_1)$ vs. $P(H_2)$ and $P(H_2)$ vs. $P(H_1)$.

In essence, FAA shows how *hard* a task is: if humans cannot agree on the correct result in more than 90% of all cases, don't expect your system to be better!

F-Measure and IAA

Combining Precision and Recall

Often a combined measure of Precision and Recall is helpful. This can be done using the *F-Measure* (equal weight for $\beta = 1$):

$$\text{F-measure} = \frac{(\beta^2 + 1) P \cdot R}{(\beta^2 R) + P}$$

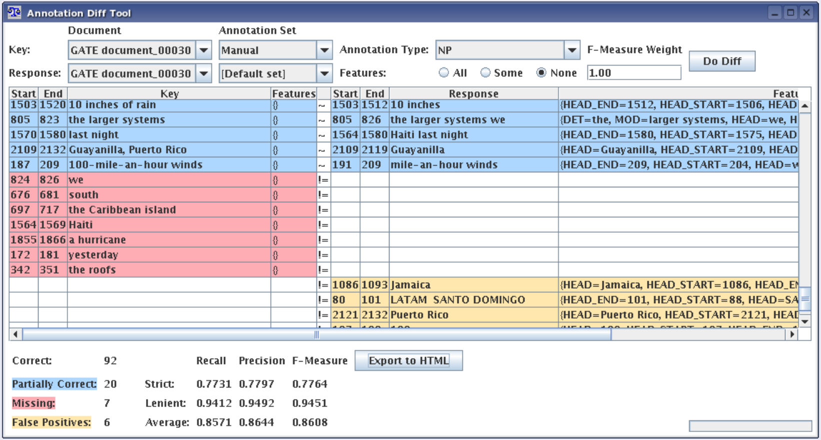
Measuring Inter-Annotator Agreement

There are many measures for computing IAA (Cohen's Kappa, prevalence, bias, ...), depending on the concrete task. On way to obtain the IAA is to compute P, R, and F values between two humans and averaging the results of $P(H_1)$ vs. $P(H_2)$ and $P(H_2)$ vs. $P(H_1)$.

In essence, FAA shows how *hard* a task is: if humans cannot agree on the correct result in more than 90% of all cases, don't expect your system to be better!

Evaluation Example

Evaluation of a Noun Phrase (NP) Chunker



Start	End	Key	Features		Start	End	Response	Featu
1503	1520	10 inches of rain	{}	~	1503	1512	10 inches	{HEAD_END=1512, HEAD_START=1506, HEAD
805	823	the larger systems	{}	~	805	826	the larger systems we	{DET=the, MOD=larger systems, HEAD=we, H
1570	1580	last night	{}	~	1564	1580	Haiti last night	{HEAD_END=1580, HEAD_START=1575, HEAD
2109	2132	Guayanilla, Puerto Rico	{}	~	2109	2119	Guayanilla	{HEAD=Guayanilla, HEAD_START=2109, HEAD
187	209	100-mile-an-hour winds	{}	~	191	209	mile-an-hour winds	{HEAD_END=209, HEAD_START=204, HEAD=w
824	826	we	{}	!=				
676	681	south	{}	!=				
697	717	the Caribbean island	{}	!=				
1564	1569	Haiti	{}	!=				
1855	1866	a hurricane	{}	!=				
172	181	yesterday	{}	!=				
342	351	the roofs	{}	!=				
				!=	1086	1093	Jamaica	{HEAD=Jamaica, HEAD_START=1086, HEAD_EN
				!=	80	101	LATAM SANTO DOMINGO	{HEAD_END=101, HEAD_START=88, HEAD=SA
				!=	2121	2132	Puerto Rico	{HEAD=Puerto Rico, HEAD_START=2121, HEAD

			Recall	Precision	F-Measure
Correct:	92				
Partially Correct:	20	Strict:	0.7731	0.7797	0.7764
Missing:	7	Lenient:	0.9412	0.9492	0.9451
False Positives:	6	Average:	0.8571	0.8644	0.8608

More Complex Metrics

OK, but. . .

. . .how do I define precision and recall for more complex tasks?

- Parsing Sentences (need to compare *parse trees*)
- Coreference Chains (need to compare *graphs*)
- Automatic Summaries (need to compare *whole texts*)

Parser Evaluation: The *PARSEVAL* Measure

A classical measure for parser evaluation is *PARSEVAL*. Compare a gold-standard parse tree to a system's one by segmenting it into its constituents (brackets). Then:

Precision is the number of brackets appearing the gold standard;

Recall measures how many of the gold standard's brackets are in the parse

Crossing Brackets measures how many brackets are crossing on average

More Complex Metrics

OK, but...

...how do I define precision and recall for more complex tasks?

- Parsing Sentences (need to compare *parse trees*)
- Coreference Chains (need to compare *graphs*)
- Automatic Summaries (need to compare *whole texts*)

Parser Evaluation: The *PARSEVAL* Measure

A classical measure for parser evaluation is *PARSEVAL*. Compare a gold-standard parse tree to a system's one by segmenting it into its constituents (brackets). Then:

Precision
: is the number of brackets appearing the gold standard;

Recall
: measures how many of the gold standard's brackets are in the parse

Crossing Brackets
: measures how many brackets are crossing on average

Evaluation: Summary

Some remarks

- Evaluation is often *very* expensive due to the large amount of time needed for manually annotating documents
- For some tasks (e.g., automatic summarization) the evaluation can be (almost) as difficult as the task itself
- Development of metrics for certain tasks, as well as the evaluation of evaluation metrics, is another branch of research
- Due to the high costs involved, and in order to ensure comparability of the results, the NLP community organises various *competitions* where system developers participate in solving prescribed tasks on the same data, using the same evaluation metrics. Examples are MUC, TREC, DUC, BioCreAtIvE, ...

Recommended Literature

NLP Foundations

- Daniel Jurafsky and James H. Martin, *Speech and Language Processing*, Prentice Hall, 2000
- Christopher D. Manning and Hinrich Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press, 1999.

Online

- Statistical natural language processing and corpus-based computational linguistics: An annotated list of resources `http://www-nlp.stanford.edu/links/statnlp.html`

Major Conferences

ACL, NAACL, EACL, COLING, HLT, EMNLP, LREC, ANLP, NLDB, ...

Part III

Technology

So you want to build a Text Mining system...

Requirements

A TM system requires a large amount of infrastructure work:

- Document handling, in various formats (plain text, HTML, XML, PDF, ...), from various sources (files, DBs, email, ...)
- Annotation handling (stand-off markup)
- Component implementations for standard tasks, like Tokenizers, Sentence Splitters, Part-of-Speech (POS) Taggers, Finite-State Transducers, Full Parsers, Classifiers, Noun Phrase Chunkers, Lemmatizers, Entity Taggers, Coreference Resolution Engines, Summarizers, ...

As well as *resources* for concrete tasks and languages:

- Lexicons, WordNets
- Grammar files and Language models
- etc.

Existing Resources

Fortunately, you don't have to start from scratch

Many (open source) tools and resources are available:

Tools: programs performing a single task, like classifiers, parsers, or NP chunkers

Frameworks: integrating architectures for combining and controlling all components and resources of an NLP system

Resources: for various languages, like lexicons, wordnets, or grammars

GATE and UIMA

Major Frameworks

Two important frameworks are:

- GATE *(General Architecture of Text Engineering)*, under development since 1995 at University of Sheffield, UK
- UIMA *(Unstructured Information Management Architecture)*, developed by IBM

Both frameworks are open source (GATE: LGPL, UIMA: CPL)
In the following, we will focus on GATE only.

General Architecture for Text Engineering (GATE)

GATE features

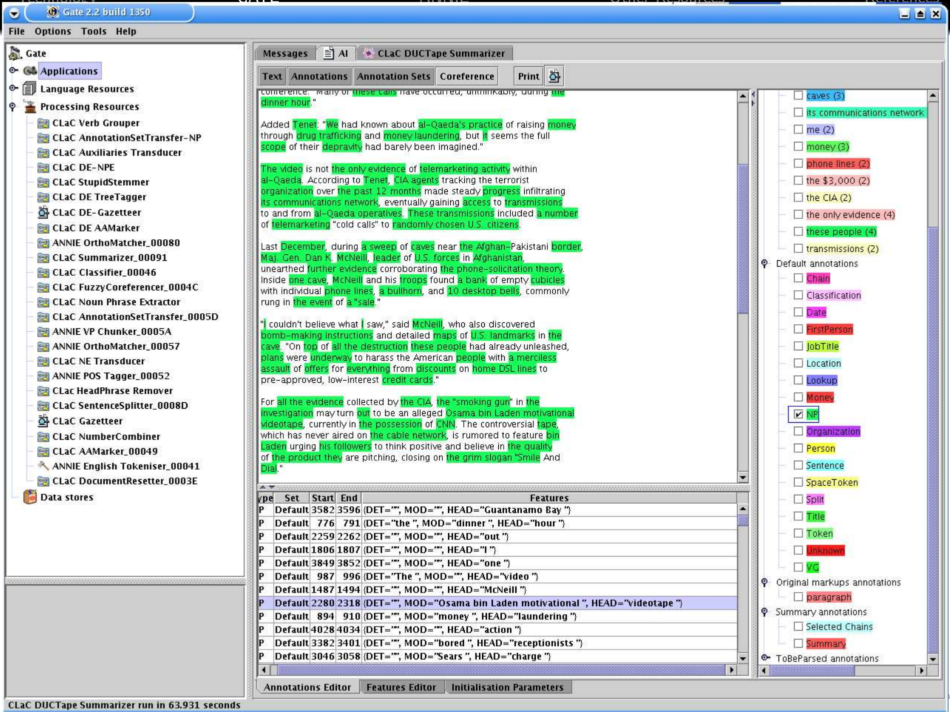
GATE *(General Architecture for Text Engineering)* is a component framework for the development of NLP applications.

Rich Infrastructure: XML Parser, Corpus management, Unicode handling, Document Annotation Model, Finite State Transducer (JAPE Grammar), etc.

Standard Components: Tokeniser, Part-of-Speech (POS) Tagger, Sentence Splitter, etc.

Set of NLP tools: Information Retrieval (IR), Machine Learning, Database access, Ontology editor, Evaluation tool, etc.

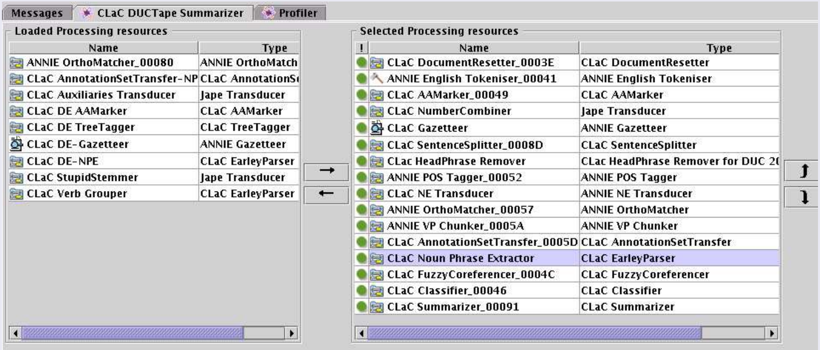
Clean Framework: Java Beans component model; Other tools can easily be integrated into GATE via *Wrappers*


Gate 2.2 build 1350
File Options Tools Help
Gate
Applications
Language Resources
Processing Resources
CLaC Verb Grouper
CLaC AnnotationSetTransfer-NP
CLaC Auxiliaries Transducer
CLaC DE-NPE
CLaC StupidStemmer
CLaC DE TreeTagger
CLaC DE-Gazetteer
CLaC DE AAMarker
ANNIE OrthoMatcher_00080
CLaC Summarizer_00091
CLaC Classifier_00046
CLaC FuzzyCoreferencer_0004C
CLaC Noun Phrase Extractor
CLaC AnnotationSetTransfer_0005D
ANNIE VP Chunker_0005A
ANNIE OrthoMatcher_00057
CLaC NE Transducer
ANNIE POS Tagger_00052
CLac HeadPhrase Remover
CLaC SentenceSplitter_0008D
CLaC Gazetteer
CLaC NumberCombiner
CLaC AAMarker_00049
ANNIE English Tokeniser_00041
CLaC DocumentResetter_0003E
Data stores
Messages
AI
CLaC DUCTape Summarizer
Text
Annotations
Annotation Sets
Coreference
Print
dinner hour."
Added Tenet: "We had known about al-Qaeda's practice of raising money through drug trafficking and money laundering, but it seems the full scope of their depravity had barely been imagined."
The video is not the only evidence of telemarketing activity within al-Qaeda. According to Tenet, CIA agents tracking the terrorist organization over the past 12 months made steady progress infiltrating its communications network, eventually gaining access to transmissions to and from al-Qaeda operatives. These transmissions included a number of telemarketing "cold calls" to randomly chosen U.S. citizens.
Last December, during a sweep of caves near the Afghan-Pakistani border, Maj. Gen. Dan K. McNeill, leader of U.S. forces in Afghanistan, unearthed further evidence corroborating the phone-solicitation theory. Inside one cave, McNeill and his troops found a bank of empty cubicles with individual phone lines, a bullhorn, and 10 desktop bells, commonly rung in the event of a "sale."
"I couldn't believe what I saw," said McNeill, who also discovered bomb-making instructions and detailed maps of U.S. landmarks in the cave. "On top of all the destruction these people had already unleashed, plans were underway to harass the American people with a merciless assault of offers for everything from discounts on home DSL lines to pre-approved, low-interest credit cards."
For all the evidence collected by the CIA, the "smoking gun" in the investigation may turn out to be an alleged Osama bin Laden motivational videotape, currently in the possession of CNN. The controversial tape, which has never aired on the cable network, is rumored to feature bin Laden urging his followers to think positive and believe in the quality of the product they are pitching, closing on the grim slogan "Smile And Dial."
ype Set Start End Features
P Default 3582 3596 {DET="", MOD="", HEAD="Guantanamo Bay "}
P Default 776 791 {DET="the ", MOD="dinner ", HEAD="hour "}
P Default 2259 2262 {DET="", MOD="", HEAD="out "}
P Default 1806 1807 {DET="", MOD="", HEAD="I "}
P Default 3849 3852 {DET="", MOD="", HEAD="one "}
P Default 987 996 {DET="The ", MOD="", HEAD="video "}
P Default 1487 1494 {DET="", MOD="", HEAD="McNeill "}
P Default 2280 2318 {DET="", MOD="Osama bin Laden motivational ", HEAD="videotape "}
P Default 894 910 {DET="", MOD="money ", HEAD="laundering "}
P Default 4028 4034 {DET="", MOD="", HEAD="action "}
P Default 3382 3401 {DET="", MOD="bored ", HEAD="receptionists "}
P Default 3046 3058 {DET="", MOD="Sears ", HEAD="charge "}
caves (3)
its communications network
me (2)
money (3)
phone lines (2)
the $3,000 (2)
the CIA (2)
the only evidence (4)
these people (4)
transmissions (2)
Default annotations
Chain
Classification
Date
FirstPerson
JobTitle
Location
Lookup
Money
NP
Organization
Person
Sentence
SpaceToken
Split
Title
Token
Unknown
VG
Original markups annotations
paragraph
Summary annotations
Selected Chains
Summary
ToBeParsed annotations
Annotations Editor
Features Editor
Initialisation Parameters
CLaC DUCTape Summarizer run in 63.931 seconds

GATE Concepts

A *Processing Pipeline* holds the required components

Component-based applications, assembled at run-time:



Results are exchanged between the components through document *annotations*.

Finite-State Language Processing with GATE

JAPE Transducers

JAPE *(Java Annotation Patterns Engine)* is a component to build finite-state transducers running over annotations from grammars.

- this is an application of *finite-state language processing*
- Transducers are basically (non-deterministic) finite-state machines, running over a graph data structure
- expressiveness of JAPE grammars corresponds to regular expressions
- basic format of a JAPE rule: `LHS:RHS`
  left-hand side matches annotations in documents, right-hand side adds annotations
- Java code can be included on the RHS, allowing computations that cannot be expressed in JAPE alone

Example for a JAPE grammar rule

Finding IP Addresses

```
// IP Address Rules
Rule: IPaddress1
(       {Token.kind == number}
        {Token.string == "."}
        {Token.kind == number}
        {Token.string == "."}
        {Token.kind == number}
        {Token.string == "."}
        {Token.kind == number}
):ipAddress -->
:ipAddress.Ip = {kind = "ipAddress", rule = "IPaddress1"}
```

Results

- matches e.g. `141.3.49.133`.
- for each detected address an annotation is added to the document at the matching start- and end-positions

A Nearly-New Information Extraction System (ANNIE)

Task: Find all *Persons* mentioned in a document

- A simple “search” function doesn’t help here
- What we need is *Information Extraction* (IE), particularly *Named Entity (NE) Detection* (entity-type `Person`)

ANNIE

GATE includes an example application, ANNIE, which can solve this task.

- developed for the news domain (newpapers, newswires), but can be adapted to other domains
- good starting point to practice NLP, IE, and TM

A Nearly-New Information Extraction System (ANNIE)

Task: Find all *Persons* mentioned in a document

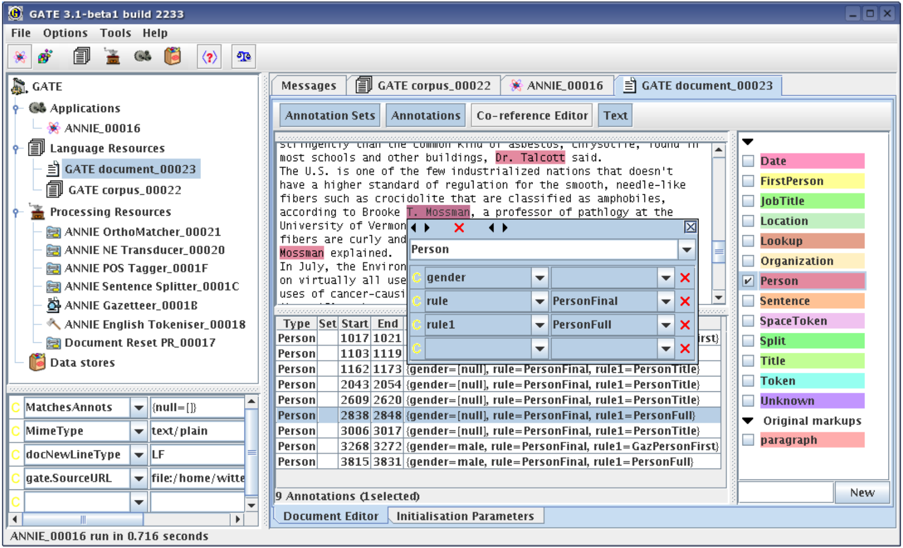
- A simple "search" function doesn't help here
- What we need is *Information Extraction* (IE), particularly *Named Entity (NE) Detection* (entity-type `Person`)

ANNIE

GATE includes an example application, ANNIE, which can solve this task.

- developed for the news domain (newpapers, newswires), but can be adapted to other domains
- good starting point to practice NLP, IE, and TM

Persons detected by ANNIE



Step 1: Tokenization

Tokenization Component

Tokenization is performed in two steps:

- a generic Unicode Tokeniser is fed with tokenisation rules for English
- afterwards, a grammer changes some of these tokens for later processing: e.g., “don’t” results in three tokens: “don”, “’”, and “t”. This is converted into two tokens, “do” and “n’t” for downstream components

For each detected token, a corresponding `Token` annotation is added to the document.

Step 1: Tokenization (Example)

Example Tokenisation Rules

```
#numbers#
// a number is any combination of digits
"DECIMAL_DIGIT_NUMBER"+ >Token;kind=number;

#whitespace#
(SPACE_SEPARATOR) >SpaceToken;kind=space;
(CONTROL) >SpaceToken;kind=control;
```

Example Output

Type	Set	Start	End	Features
Token		158	163	{kind=word, length=5, orth=lowercase, string=years}
SpaceToken		163	164	{kind=space, length=1, string= }
Token		164	167	{kind=word, length=3, orth=lowercase, string=ago}
Token		167	168	{kind=punctuation, length=1, string=,}
SpaceToken		168	169	{kind=space, length=1, string= }
Token		169	180	{kind=word, length=11, orth=lowercase, string=researchers}
SpaceToken		180	181	{kind=space, length=1, string= }

1417 Annotations (0 selected)

Step 2: Gazetteering

Gazetteer Component

The *Gazetteer* uses structured plain text lists to annotate words with a `major_type` and `minor_type`

- each lists represents a concept or type, e.g., female first names, mountains, countries, male titles, streets, festivals, dates, planets, organizations, cities, ...
- ambiguities are not resolved at this step—e.g., a string can be annotated both as *female first name* and *city*
- GATE provides several different Gazetteer implementation: Simple Gazetteer, HashGazetteer, FlexibleGazetteer, OntoGazetteer, ...
- Gazetteer lists can be (a) created by hand, (b) derived from databases, (c) “learned” through patterns, e.g., from web sites

Step 2: Gazetteering (Example)

Gazetteer Definition

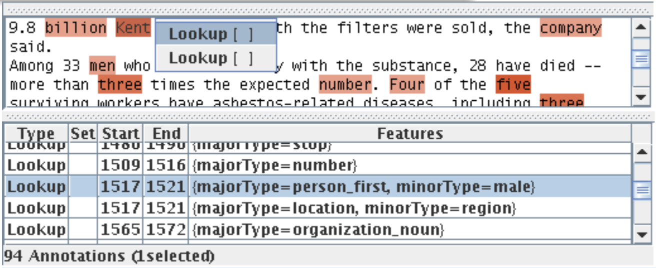
Connecting lists with major/minor types:

```
organization.lst:organization
organization_nouns.lst:organization_noun
person_ambig.lst:person_first:ambig
person_ending.lst:person_ending
person_female.lst:person_first:female
person_female_cap.lst:person_first:female
person_female_lower.lst:person_first:female
person_full.lst:person_full
```

Example List

Person_female.lst:

```
Acantha
Acenith
Achala
Achava
Achsah
Ada
Adah
Adalgisa
```



Step 3: Sentence Splitting

Task: Split Stream of Tokens into Sentences

Sentences are important units in texts

- Correct detection important for downstream components, e.g., the POS-Tagger

Precise splitting can be annoyingly hard:

- a "." (dot) often does **not** indicate an EOS
- Abbreviations *"The U.S. government"*, but: "... *announced by the U.S."*
- Ambiguous boundaries "!", ";", ":", nested sentences (e.g., inside quotations) etc.
- Formatting detection (headlines, footnotes, tables, ...)

ANNIE Sentence Splitter

Uses grammar rules and abbreviation lists to detect sentence boundaries.

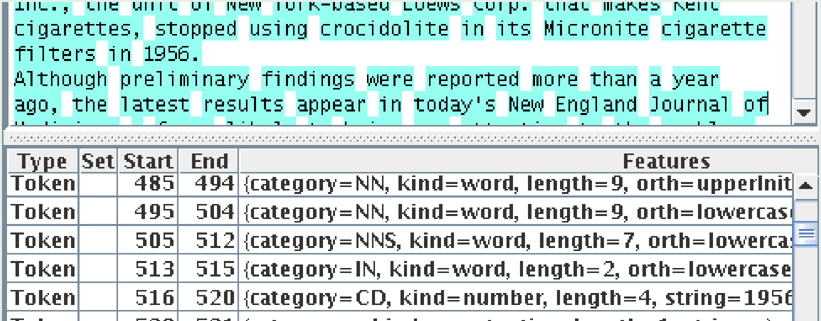
Step 4: Part-of-Speech (POS) Tagging

Producing POS Annotations

POS-Tagging assigns a part-of-speech-tag (POS tag) to each Token.

- GATE includes the Hepple tagger for English, which is a modified version of the Brill tagger

Example output



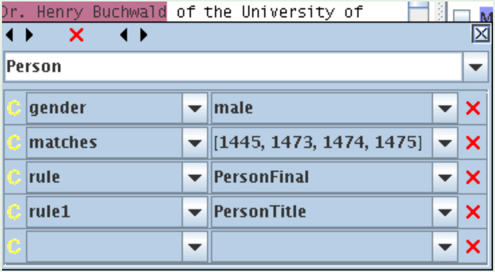
Type	Set	Start	End	Features
Token		485	494	{category=NN, kind=word, length=9, orth=upperInit
Token		495	504	{category=NN, kind=word, length=9, orth=lowercas
Token		505	512	{category=NNS, kind=word, length=7, orth=lowerca
Token		513	515	{category=IN, kind=word, length=2, orth=lowercase
Token		516	520	{category=CD, kind=number, length=4, string=1956

Step 5: Named Entity (NE) Detection

Transducer-based NE Detection

Using all the information obtained in the previous steps (Tokens, Gazetteer lookups, POS tags), ANNIE now runs a sequence of JAPE-Transducers to detect Named Entities (NE)s.

Example for a detected *Person*



We can now look at the grammar rules that found this person.

Entity Detection: Finding Persons

Strategy

A JAPE grammar rule combines information obtained from POS-tags with Gazetteer lookup information

- although the last name in the example is not in any list, it can be found based on its POS tag and an additional first name/last name rule (not shown)
- many additional rules for other Person patterns, as well as Organizations, Dates, Addresses, ...

Persons with Titles

```
Rule:   PersonTitle
Priority: 35
(
 {Token.category == DT}|
 {Token.category == PRP}|
 {Token.category == RB}
)?
(
 (TITLE)+
 ((FIRSTNAME | FIRSTNAMEAMBIG
    | INITIALS2)
 )?
  (PREFIX)*
  (UPPER)
 (PERSONENDING)?
)
:person --> ...
```

Step 6: Coreference Resolution

Finding Coreferences

Remember the problem of coreference resolution:

- need to find all instances of an entity in a text,
- even when referred to by different textual descriptors

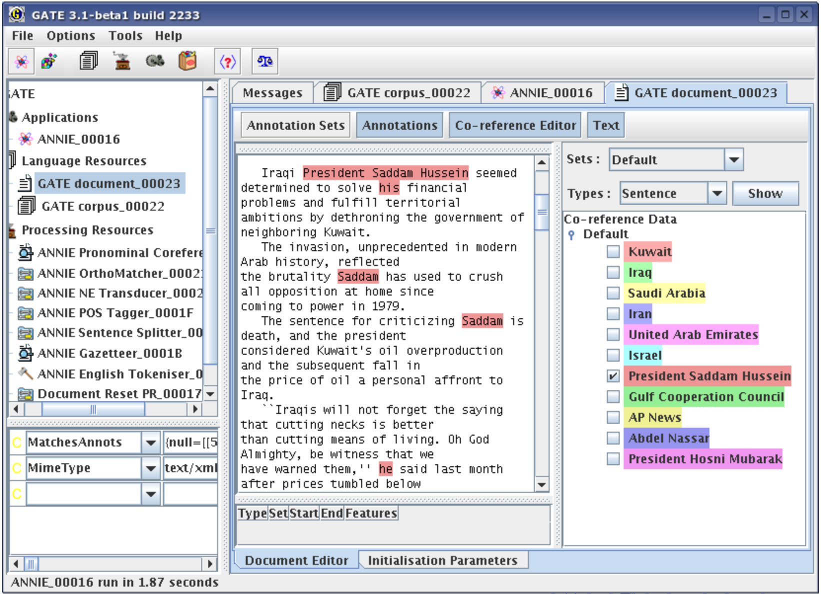
Coreference resolution in ANNIE

GATE provides two components for performing a restricted subset of coreference resolution:

Pronomial Coreferences finds anaphors (e.g., "he" referring to a previously mentioned person) and also some cataphors (e.g., "Before *he* bought the car, *John*...")

Nominal Coreferences a number of JAPE rules match entities based on orthographic features, e.g., a person "John Smith" will be matched with "Mr. Smith"

Coreference Resolution Example



GATE Plugins

More GATE Plugins

GATE comes with a number of other language plugins, which are either implemented directly for GATE, or use *wrappers* to access external resources:

Verb Grouper: a JAPE grammar to analyse verb groups (VGs)

SUPPLE Parser: a Prolog-based parser for (partial) parsing that can create logical forms

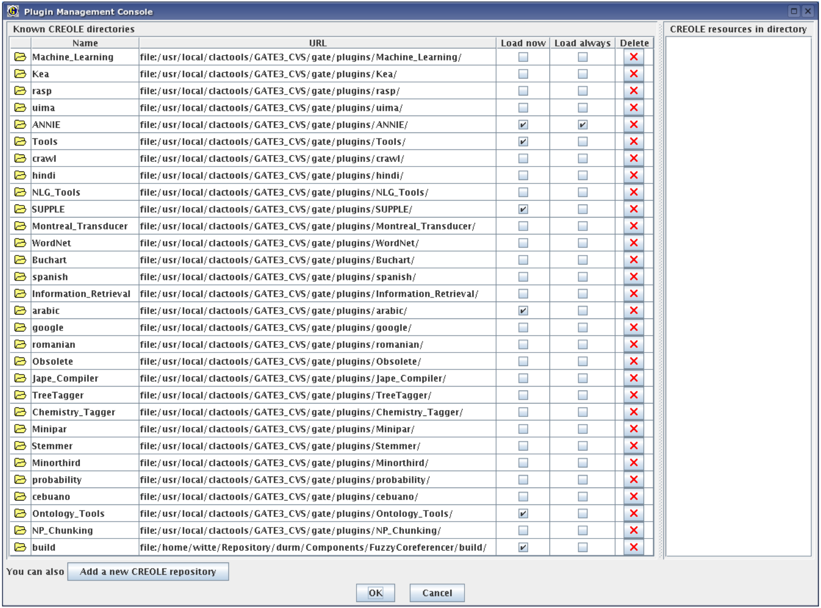
Chemistry Tagger: component to find chemistry items (formulas, elements etc.)

Web Crawler: wrapper for the Websphinx crawler to construct a corpus from the Web

Kea Wrapper: for the *Kea* keyphrase detector

Ontology tools: for using (Jena) ontologies in pipelines, e.g., with the OntoGazetteer and Ontology-aware JAPE transducer

GATE Plugins



Plugin Management Console

Known CREOLE directories

Name	URL	Load now	Load always	Delete
Machine_Learning	file:/usr/local/clactools/GATE3_CVS/gate/plugins/Machine_Learning/	☐	☐	✕
Kea	file:/usr/local/clactools/GATE3_CVS/gate/plugins/Kea/	☐	☐	✕
rasp	file:/usr/local/clactools/GATE3_CVS/gate/plugins/rasp/	☐	☐	✕
uima	file:/usr/local/clactools/GATE3_CVS/gate/plugins/uima/	☐	☐	✕
ANNIE	file:/usr/local/clactools/GATE3_CVS/gate/plugins/ANNIE/	☑	☑	✕
Tools	file:/usr/local/clactools/GATE3_CVS/gate/plugins/Tools/	☑	☐	✕
crawl	file:/usr/local/clactools/GATE3_CVS/gate/plugins/crawl/	☐	☐	✕
hindi	file:/usr/local/clactools/GATE3_CVS/gate/plugins/hindi/	☐	☐	✕
NLG_Tools	file:/usr/local/clactools/GATE3_CVS/gate/plugins/NLG_Tools/	☐	☐	✕
SUPPLE	file:/usr/local/clactools/GATE3_CVS/gate/plugins/SUPPLE/	☑	☐	✕
Montreal_Transducer	file:/usr/local/clactools/GATE3_CVS/gate/plugins/Montreal_Transducer/	☐	☐	✕
WordNet	file:/usr/local/clactools/GATE3_CVS/gate/plugins/WordNet/	☐	☐	✕
Buchart	file:/usr/local/clactools/GATE3_CVS/gate/plugins/Buchart/	☐	☐	✕
spanish	file:/usr/local/clactools/GATE3_CVS/gate/plugins/spanish/	☐	☐	✕
Information_Retrieval	file:/usr/local/clactools/GATE3_CVS/gate/plugins/Information_Retrieval/	☐	☐	✕
arabic	file:/usr/local/clactools/GATE3_CVS/gate/plugins/arabic/	☑	☐	✕
google	file:/usr/local/clactools/GATE3_CVS/gate/plugins/google/	☐	☐	✕
romanian	file:/usr/local/clactools/GATE3_CVS/gate/plugins/romanian/	☐	☐	✕
Obsolete	file:/usr/local/clactools/GATE3_CVS/gate/plugins/Obsolete/	☐	☐	✕
Jape_Compiler	file:/usr/local/clactools/GATE3_CVS/gate/plugins/Jape_Compiler/	☐	☐	✕
TreeTagger	file:/usr/local/clactools/GATE3_CVS/gate/plugins/TreeTagger/	☐	☐	✕
Chemistry_Tagger	file:/usr/local/clactools/GATE3_CVS/gate/plugins/Chemistry_Tagger/	☐	☐	✕
Minipar	file:/usr/local/clactools/GATE3_CVS/gate/plugins/Minipar/	☐	☐	✕
Stemmer	file:/usr/local/clactools/GATE3_CVS/gate/plugins/Stemmer/	☐	☐	✕
Minorthird	file:/usr/local/clactools/GATE3_CVS/gate/plugins/Minorthird/	☐	☐	✕
probability	file:/usr/local/clactools/GATE3_CVS/gate/plugins/probability/	☐	☐	✕
cebuano	file:/usr/local/clactools/GATE3_CVS/gate/plugins/cebuano/	☐	☐	✕
Ontology_Tools	file:/usr/local/clactools/GATE3_CVS/gate/plugins/Ontology_Tools/	☑	☐	✕
NP_Chunking	file:/usr/local/clactools/GATE3_CVS/gate/plugins/NP_Chunking/	☐	☐	✕
build	file:/home/witte/Repository/durm/Components/FuzzyCoreferencer/build/	☑	☐	✕

CREOLE resources in directory

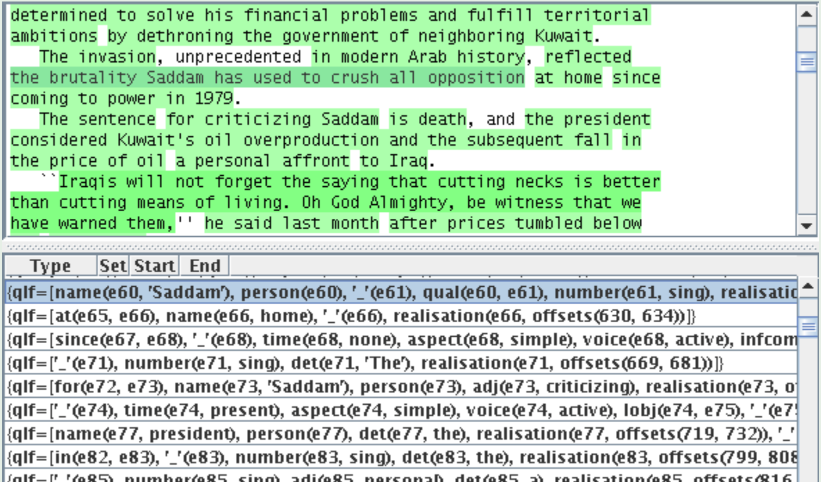
You can also [Add a new CREOLE repository]

[OK] [Cancel]

SUPPLE Parser

Bottom-up Parser for English

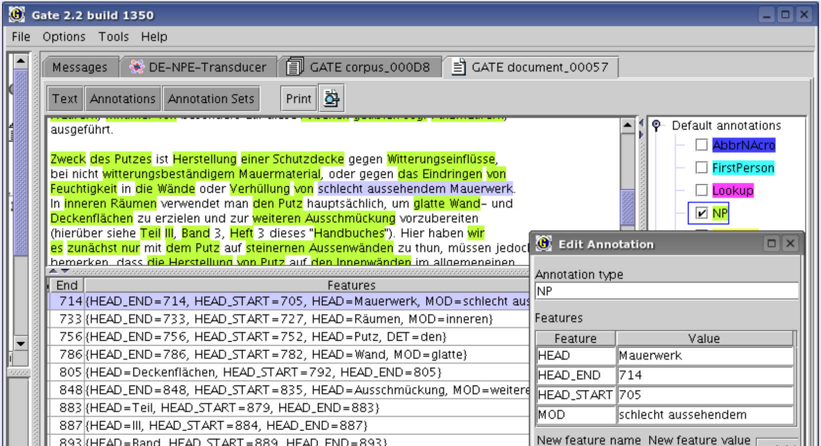
Constructs (partial) syntax trees and logical forms for English sentences. Implemented in Prolog.



Multi-lingual Noun Phrase Chunker

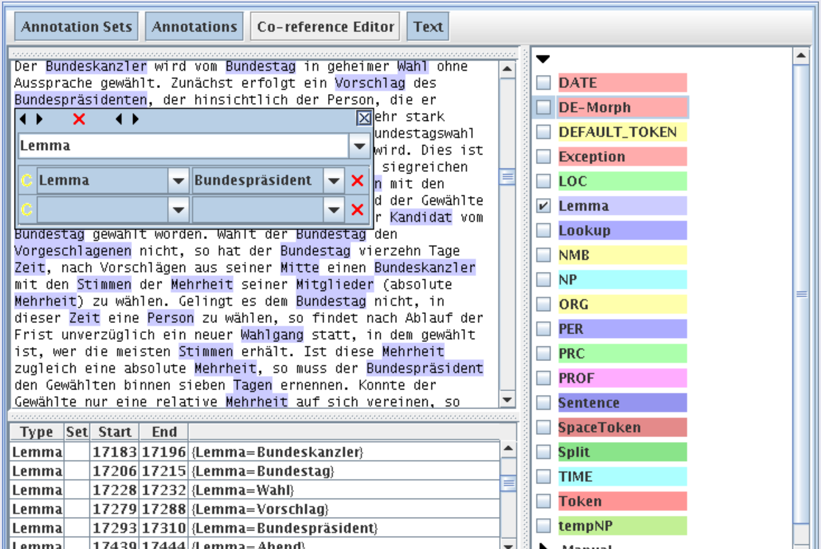
MuNPEx

MuNPEx is an open-source multi-lingual noun phrase (NP) chunker implemented in JAPE. Currently supported are English, German, French, and Spanish (in beta).



The *Durm* German Lemmatizer

An Open Source Lemmatizer for German



References

Frameworks

The GATE *(General Architecture for Text Engineering*) System:

- http://gate.ac.uk
- http://sourceforge.net/projects/gate
- User's Guide: http://gate.ac.uk/sale/tao/

IBM's UIMA *(Unstructured Information Management Architecture)*:

- http://www.research.ibm.com/UIMA/
- http://sourceforge.net/projects/uima-framework/

Other Resources

- WordNet: http://wordnet.princeton.edu/
- MuNPEx: http://www.ipd.uka.de/~durm/tm/munpex/

Part IV

Applications

Text Mining Applications

Bringing it all together...

We now look at some actual Text Mining applications:

Automatic Summarization: of single and multiple documents

Opinion Mining: extracting *opinions* by consumers regarding companies and their products

Question-Answering: answering factual questions

Text Mining in Biology: the BioRAT and MutationMiner systems

For Summarization and Biology, we'll look into some systems in detail.

An everyday task

Given:

Lots of information; WWW with millions of pages

Question:

What countries are or have been involved in land or water boundary disputes with each other over oil resources or exploration? How have disputes been resolved, or towards what kind of resolution are the countries moving? What other factors affect the disputes?

Task:

Write a summary answering the question in about 250 words!

An everyday task

Given:

Lots of information; WWW with millions of pages

Question:

What countries are or have been involved in land or water boundary disputes with each other over oil resources or exploration? How have disputes been resolved, or towards what kind of resolution are the countries moving? What other factors affect the disputes?

Task:

Write a summary answering the question in about 250 words!

Automatic Summarization

Definition

A summary text is a condensed derivative of a source text, reducing content by selection and/or generalisation on what is important.

Note

Distinguish between:

- *abstracting*-based summaries, and
- *extracting*-based summaries.

Automatically created summaries are (almost) exclusively text extracts.

The Challenge

to identify the informative segments at the expense of the rest

Automatic Summarization

Definition

A summary text is a condensed derivative of a source text, reducing content by selection and/or generalisation on what is important.

Note

Distinguish between:

- *abstracting*-based summaries, and
- *extracting*-based summaries.

Automatically created summaries are (almost) exclusively text extracts.

The Challenge

to identify the informative segments at the expense of the rest

Automatic Summarization

Definition

A summary text is a condensed derivative of a source text, reducing content by selection and/or generalisation on what is important.

Note

Distinguish between:

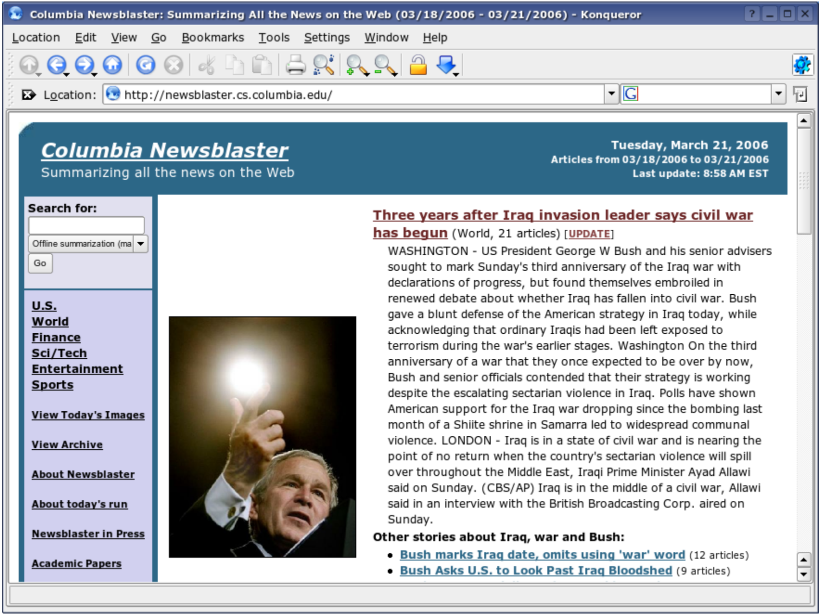
- *abstracting*-based summaries, and
- *extracting*-based summaries.

Automatically created summaries are (almost) exclusively text extracts.

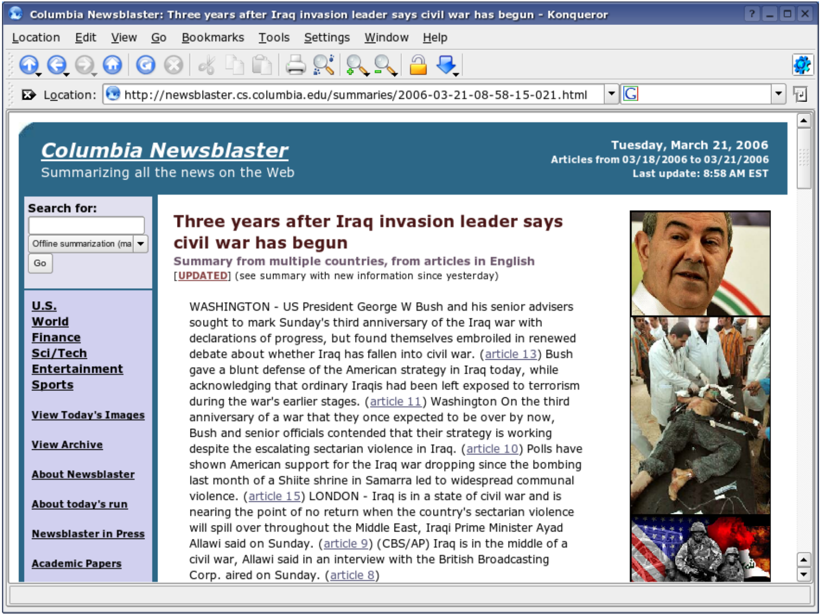
The Challenge

to identify the informative segments at the expense of the rest

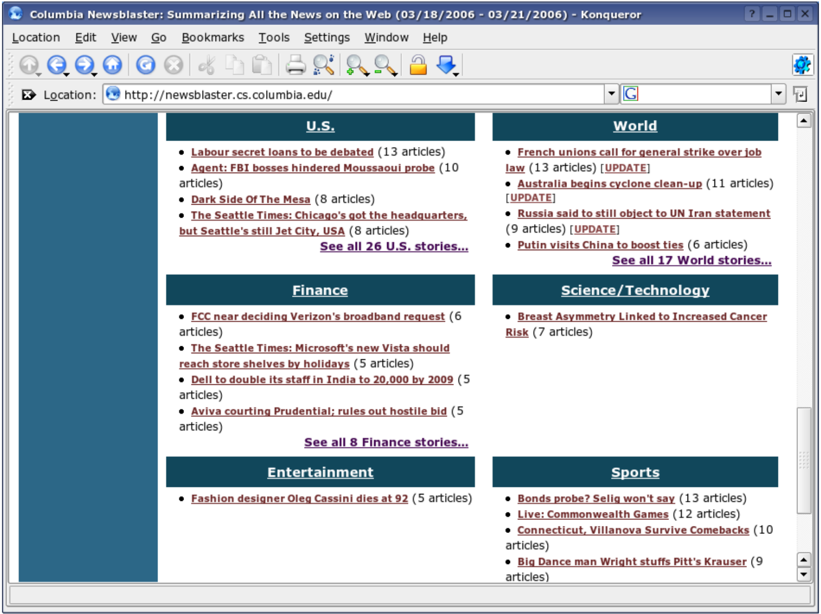
The NewsBlaster System (Columbia U.)



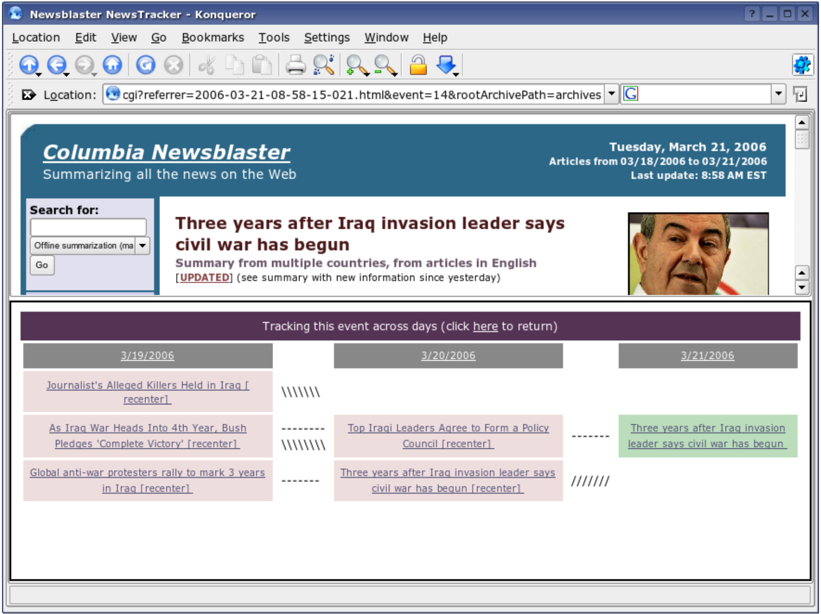
A Multi-Document Summary generated by NewsBlaster



NewsBlaster: Article Classification



NewsBlaster: Tracking Events over Time



Research in Automatic Summarization

The Challenge

- Various summarization systems produce different kinds of summaries, from different data, for different purposes, using different evaluations
- Impossible to measure (scientific) progress

Document Understanding Conference (DUC)

The solution: hold a *competition*

- Started in 2001
- Organized by U.S. National Institute of Standardization and Technology (NIST)
- Forum to compare summarization systems
- For all systems the same tasks, data, and evaluation methods

Research in Automatic Summarization

The Challenge

- Various summarization systems produce different kinds of summaries, from different data, for different purposes, using different evaluations
- Impossible to measure (scientific) progress

Document Understanding Conference (DUC)

The solution: hold a *competition*

- Started in 2001
- Organized by U.S. National Institute of Standardization and Technology (NIST)
- Forum to compare summarization systems
- For all systems the same tasks, data, and evaluation methods

Document Understanding Conference (DUC)

Data

- newspaper and newswire articles (AP, NYT, XIE, ...)
- topical clusters of various length (2004: 10, 2005: 25–50, 2006: 25

Tasks

In 2004:

- short summaries of single articles (10 words)
- summaries of single articles (100 words)
- multi-document summaries of a 10-document cluster
- cross-language summaries (machine translated Arabic)
- summaries focused by a question *"Who is X?"*

In 2005–2006:

- Focused multi-document summaries for a given *context*

Summarization System ERSS (CLaC/IPD)

Main processing steps

Preprocessing Tokenizer, Sentence Splitter, POS Tagger, ...

MuNPEx noun phrase chunker (JAPE-based)

FCR fuzzy coreference resolution algorithm

Classy naive Bayesian classifier for multi-dimensional text categorization

Summarizer summarization framework with individual strategies

Implementation based on the GATE architecture.

Summarization System ERSS (CLaC/IPD)

Main processing steps

Preprocessing Tokenizer, Sentence Splitter, POS Tagger, . . .

MuNPEx noun phrase chunker (JAPE-based)

FCR fuzzy coreference resolution algorithm

Classy naive Bayesian classifier for multi-dimensional text categorization

Summarizer summarization framework with individual strategies

Implementation based on the GATE architecture.

Summarization System ERSS (CLaC/IPD)

Main processing steps

Preprocessing Tokenizer, Sentence Splitter, POS Tagger, ...

MuNPEx noun phrase chunker (JAPE-based)

FCR fuzzy coreference resolution algorithm

Classy naive Bayesian classifier for multi-dimensional text categorization

Summarizer summarization framework with individual strategies

Implementation based on the GATE architecture.

Summarization System ERSS (CLaC/IPD)

Main processing steps

Preprocessing Tokenizer, Sentence Splitter, POS Tagger, ...

MuNPEx noun phrase chunker (JAPE-based)

FCR fuzzy coreference resolution algorithm

Classy naive Bayesian classifier for multi-dimensional text categorization

Summarizer summarization framework with individual strategies

Implementation based on the GATE architecture.

Summarization System ERSS (CLaC/IPD)

Main processing steps

Preprocessing Tokenizer, Sentence Splitter, POS Tagger, ...

MuNPEx noun phrase chunker (JAPE-based)

FCR fuzzy coreference resolution algorithm

Classy naive Bayesian classifier for multi-dimensional text categorization

Summarizer summarization framework with individual strategies

Implementation based on the GATE architecture.

Summarization System ERSS (CLaC/IPD)

Main processing steps

Preprocessing Tokenizer, Sentence Splitter, POS Tagger, ...

MuNPEx noun phrase chunker (JAPE-based)

FCR fuzzy coreference resolution algorithm

Classy naive Bayesian classifier for multi-dimensional text categorization

Summarizer summarization framework with individual strategies

Implementation based on the GATE architecture.

ERSS: Preprocessing Steps

Basic Preprocessing

Tokenization, Sentence Splitting, POS Tagging, ...

Number Interpreter

Locates number expressions and assignes numerical values, e.g., *"two"* $\rightarrow$ *2.*

Abbreviation & Acronym Detector

Scans tokens for acronyms *("GM", "IBM", ...)* and abbreviations *(e.g., "e.g.", "Fig.", ...)* and adds the full text.

Gazetteer

Scans input tokens and adds *type* information based on a number of word lists: *city, company, currency, festival, mountain, person_female, planet, region, street, timezone, title, water, ...*

ERSS: Preprocessing Steps

Basic Preprocessing

Tokenization, Sentence Splitting, POS Tagging, . . .

Number Interpreter

Locates number expressions and assignes numerical values, e.g., *"two"* $\rightarrow$ *2*.

Abbreviation & Acronym Detector

Scans tokens for acronyms *("GM", "IBM", . . .)* and abbreviations *(e.g., "e.g.", "Fig.", . . .)* and adds the full text.

Gazetteer

Scans input tokens and adds *type* information based on a number of word lists: *city, company, currency, festival, mountain, person_female, planet, region, street, timezone, title, water, . . .*

ERSS: Preprocessing Steps

Basic Preprocessing

Tokenization, Sentence Splitting, POS Tagging, . . .

Number Interpreter

Locates number expressions and assignes numerical values, e.g., *"two"* $\rightarrow$ *2*.

Abbreviation & Acronym Detector

Scans tokens for acronyms *("GM", "IBM", . . .)* and abbreviations *(e.g., "e.g.", "Fig.", . . .)* and adds the full text.

Gazetteer

Scans input tokens and adds *type* information based on a number of word lists: *city, company, currency, festival, mountain, person_female, planet, region, street, timezone, title, water, . . .*

ERSS: Preprocessing Steps

Basic Preprocessing

Tokenization, Sentence Splitting, POS Tagging, . . .

Number Interpreter

Locates number expressions and assignes numerical values, e.g., *"two"* $\rightarrow$ *2*.

Abbreviation & Acronym Detector

Scans tokens for acronyms *("GM", "IBM", . . .)* and abbreviations *(e.g., "e.g.", "Fig.", . . .)* and adds the full text.

Gazetteer

Scans input tokens and adds *type* information based on a number of word lists: *city, company, currency, festival, mountain, person_female, planet, region, street, timezone, title, water, . . .*

Preprocessing Steps (II)

Named Entity (NE) Recognition

Scans a sequence of (annotated) tokens with JAPE grammars and adds NE information: *Date*, *Person*, *Organization*, ...

Example: Tokens "10", "o", "'", "clock" → *Date::TimeOClock*

JAPE Grammars

- Regular-expression based grammars
- used to generate *finite state Transducers* (non-deterministic finite state machines)

Example Grammar

```
Rule: TimeOClock  // ten o'clock
({Lookup.minorType == hour}
 {Token.string == "o"}
 {Token.string == "'"}
 {Token.string == "clock"}
):time
-->
 :time.TempTime = {kind = "positive",
                   rule = "TimeOClock"}
```

Preprocessing Steps (II)

Named Entity (NE) Recognition

Scans a sequence of (annotated) tokens with JAPE grammars and adds NE information: *Date*, *Person*, *Organization*, ...

Example: Tokens “10”, “o”, “’”, “clock” → *Date::TimeOClock*

JAPE Grammars

- Regular-expression based grammars
- used to generate *finite state Transducers* (non-deterministic finite state machines)

Example Grammar

```
Rule: TimeOClock  // ten o'clock
({Lookup.minorType == hour}
 {Token.string == "o"}
 {Token.string == "'"}
 {Token.string == "clock"}
):time
-->
 :time.TempTime = {kind = "positive",
                   rule = "TimeOClock"}
```

Preprocessing Steps (II)

Named Entity (NE) Recognition

Scans a sequence of (annotated) tokens with JAPE grammars and adds NE information: *Date*, *Person*, *Organization*, . . .

Example: Tokens "10", "o", "'", "clock" → *Date::TimeOClock*

JAPE Grammars

- Regular-expression based grammars
- used to generate *finite state Transducers* (non-deterministic finite state machines)

Example Grammar

```
Rule: TimeOClock  // ten o'clock
({Lookup.minorType == hour}
 {Token.string == "o"}
 {Token.string == "'"}
 {Token.string == "clock"}
):time
-->
 :time.TempTime = {kind = "positive",
                   rule = "TimeOClock"}
```

Fuzzy Coreference Resolution

Coreference Resolution

Input to a coreference resolution algorithm is a set of noun phrases (NPs). Example: *Mr. Bush* $\overset{?}{\longleftrightarrow}$ *the president* $\overset{?}{\longleftrightarrow}$ *he*

Fuzzy Representation of Coreference

Core idea: coreference between noun phrases is almost never "100% certain"

- fuzzy model: represent certainty of coreference *explicitly* with a membership degree
- formally: represent fuzzy chain $\mathcal{C}$ with a fuzzy set $\mu_{\mathcal{C}}$, mapping the domain of all NPs in a text to the [0,1]-interval
- then, each noun phrase np_i has a corresponding membership degree $\mu_{\mathcal{C}}(\mathrm{np}_i)$, indicating how certain this NP is a member of chain $\mathcal{C}$

Fuzzy Coreference Resolution

Coreference Resolution

Input to a coreference resolution algorithm is a set of noun phrases (NPs). Example: *Mr. Bush* $\overset{?}{\longleftrightarrow}$ *the president* $\overset{?}{\longleftrightarrow}$ *he*

Fuzzy Representation of Coreference

Core idea: coreference between noun phrases is almost never "100% certain"

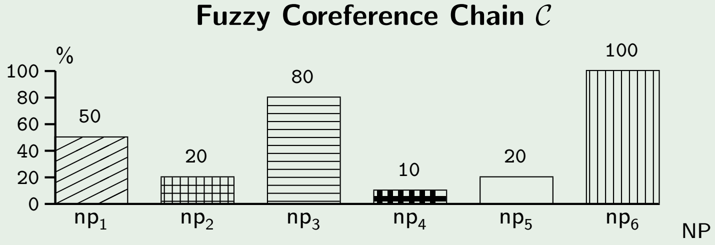
- fuzzy model: represent certainty of coreference *explicitly* with a membership degree
- formally: represent fuzzy chain $\mathcal{C}$ with a fuzzy set $\mu_{\mathcal{C}}$, mapping the domain of all NPs in a text to the [0,1]-interval
- then, each noun phrase np_i has a corresponding membership degree $\mu_{\mathcal{C}}(\mathrm{np}_i)$, indicating how certain this NP is a member of chain $\mathcal{C}$

Fuzzy Coreference Resolution

Fuzzy Coreference Chain

Fuzzy set $\mu_{\mathcal{C}} : NP \rightarrow [0, 1]$

Example



Fuzzy Coreference Chains

Properties of fuzzy chains

- each chain holds *all* noun phrases in a text
- i.e., each NP is a member of every chain (but with very different certainties)
- we don't have to reject inconsistencies right away — they can be reconciled later through suitable fuzzy operators
- also, there is no arbitrary boundary for discriminating between *"corefering"* and *"not corefering"*
- thus, in this step we don't lose information we might need later

Fuzzy Clustering

How can we *build* fuzzy chains?

- Use knowledge-poor heuristics to check for coreference between NP pairs
- Examples: Substring, Synonym/Hypernym, Pronoun, CommonHead, Acronym...
- Fuzzy heuristic: return a *degree* of coreference $\in [0, 1]$

Creating Chains by Clustering

Idea: initally, each NP represents one chain (where it is its *medoid*). Then:

- apply a single-link hierarchical clustering strategy,
- using the fuzzy degree as an (inverse) distance measure

This results in NP clusters, which can be converted into coreference chains.

Fuzzy Clustering

How can we *build* fuzzy chains?

- Use knowledge-poor heuristics to check for coreference between NP pairs
- Examples: Substring, Synonym/Hypernym, Pronoun, CommonHead, Acronym...
- Fuzzy heuristic: return a *degree* of coreference $\in [0, 1]$

Creating Chains by Clustering

Idea: initally, each NP represents one chain (where it is its *medoid*). Then:

- apply a single-link hierarchical clustering strategy,
- using the fuzzy degree as an (inverse) distance measure

This results in NP clusters, which can be converted into coreference chains.

Designing Fuzzy Heuristics

Fuzzy Heuristics

How can we compute a coreference degree $\mu^{\mathcal{H}_i}_{(\mathrm{np}_j,\mathrm{np}_k)}$?

Fuzzy Substring Heuristic: (character n-gram match) return coreference degree of 1.0 if two NP string are identical, 0.0 if they share no substring. Otherwise, select longest matching substring and set coreference degree to its percentage of first NP.

Fuzzy Synonym/Hypernym Heuristic: Synonyms (determined through *WordNet*) receive a coreference degree of 1.0. If two NPs are hypernyms, set the coreference degree depending on distance in the hierarchy (i.e., longer paths result in lower certainty degrees).

Designing Fuzzy Heuristics

Fuzzy Heuristics

How can we compute a coreference degree $\mu^{\mathcal{H}_i}_{(\mathsf{np}_j,\mathsf{np}_k)}$?

Fuzzy Substring Heuristic: (character n-gram match) return coreference degree of 1.0 if two NP string are identical, 0.0 if they share no substring. Otherwise, select longest matching substring and set coreference degree to its percentage of first NP.

Fuzzy Synonym/Hypernym Heuristic: Synonyms (determined through *WordNet*) receive a coreference degree of 1.0. If two NPs are hypernyms, set the coreference degree depending on distance in the hierarchy (i.e., longer paths result in lower certainty degrees).

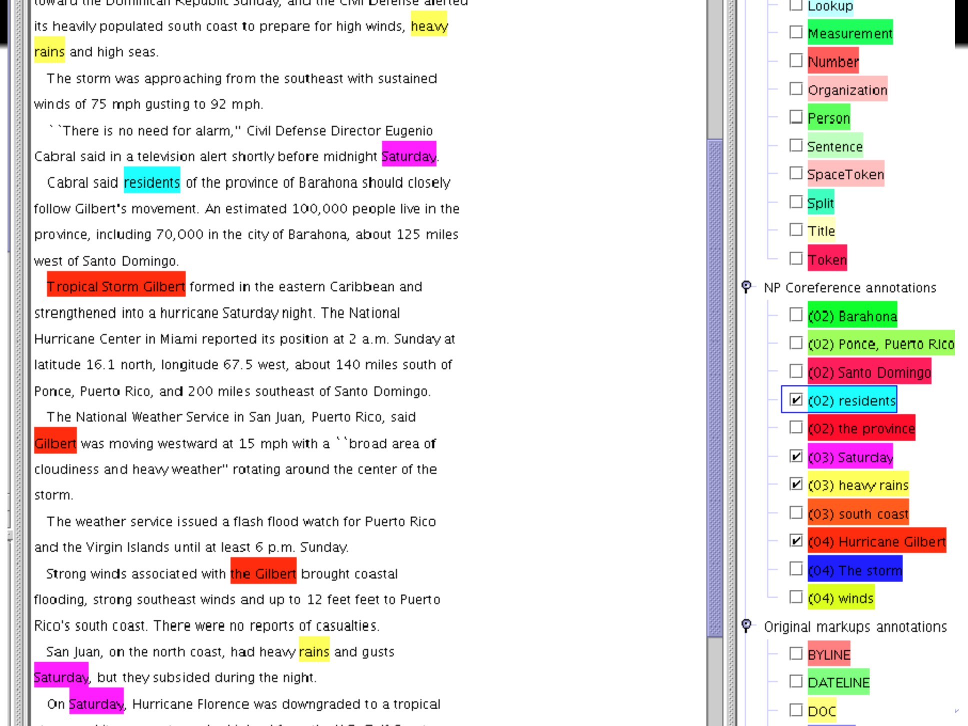
Designing Fuzzy Heuristics

Fuzzy Heuristics

How can we compute a coreference degree $\mu^{\mathcal{H}_i}_{(\mathrm{np}_j,\mathrm{np}_k)}$?

Fuzzy Substring Heuristic: (character n-gram match) return coreference degree of 1.0 if two NP string are identical, 0.0 if they share no substring. Otherwise, select longest matching substring and set coreference degree to its percentage of first NP.

Fuzzy Synonym/Hypernym Heuristic: Synonyms (determined through *WordNet*) receive a coreference degree of 1.0. If two NPs are hypernyms, set the coreference degree depending on distance in the hierarchy (i.e., longer paths result in lower certainty degrees).


its heavily populated south coast to prepare for high winds, heavy rains and high seas.
The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph.
``There is no need for alarm,'' Civil Defense Director Eugenio Cabral said in a television alert shortly before midnight Saturday.
Cabral said residents of the province of Barahona should closely follow Gilbert's movement. An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo.
Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.
The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a ``broad area of cloudiness and heavy weather'' rotating around the center of the storm.
The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.
Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds and up to 12 feet feet to Puerto Rico's south coast. There were no reports of casualties.
San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night.
On Saturday, Hurricane Florence was downgraded to a tropical
Lookup
Measurement
Number
Organization
Person
Sentence
SpaceToken
Split
Title
Token
NP Coreference annotations
(02) Barahona
(02) Ponce, Puerto Rico
(02) Santo Domingo
(02) residents
(02) the province
(03) Saturday
(03) heavy rains
(03) south coast
(04) Hurricane Gilbert
(04) The storm
(04) winds
Original markups annotations
BYLINE
DATELINE
DOC

Summarizer

ERSS (Experimental Resolution System Summarizer)

A Summary should contain the most important entities within a text. Assumption: these are also mentioned more often, hence result in longer coreference chains.

Summarization Algorithm (Single Documents)

1. Rank coreference chains by size (and other features)
2. For each chain: select highest-ranking NP/Sentence
3. extract NP (short summary)
   or complete sentence (long summary)
4. continue with next-longest chain until length limit has been reached

Summarizer

ERSS (Experimental Resolution System Summarizer)

A Summary should contain the most important entities within a text. Assumption: these are also mentioned more often, hence result in longer coreference chains.

Summarization Algorithm (Single Documents)

1. Rank coreference chains by size (and other features)
2. For each chain: select highest-ranking NP/Sentence
3. extract NP (short summary)
   or complete sentence (long summary)
4. continue with next-longest chain until length limit has been reached

ERSS: Keyword-style Summary Examples

Automatically created 10-word-summaries

Can you guess the text's topic?

Space News: [the shuttle Discovery's Hubble repair mission, the observatory's central computer]

People & Politics: [Lewinsky, President Bill Clinton, her testimony, the White House scandal]

Business & Economics: [PAL, the company's stock, a management-proposed recovery plan, the laid-off workers]

(from DUC 2003)

ERSS: Single-Document Summary Example

Automatically created 100-word summary (from DUC 2004)

President Yoweri Museveni insists they will remain there until Ugandan security is guaranteed, despite Congolese President Laurent Kabila's protests that Uganda is backing Congolese rebels attempting to topple him. After a day of fighting, Congolese rebels said Sunday they had entered Kindu, the strategic town and airbase in eastern Congo used by the government to halt their advances. The rebels accuse Kabila of betraying the eight-month rebellion that brought him to power in May 1997 through mismanagement and creating divisions among Congo's 400 tribes. A day after shooting down a jetliner carrying 40 people, rebels clashed with government troops near a strategic airstrip in eastern Congo on Sunday.

Summarizer (II): more complicated summaries

Multi-Document Summaries

Many tasks in DUC require summaries of *multiple* documents:

- cross-document summary
- focused summary
- context-based summary (DUC 2005, 2006)

Solution

Additionally build *cross-document coreference chains* and summarize using a *fuzzy cluster graph algorithm.*

- For focused and context-based summaries, only use those chains that connect the question(s) with the documents (even if they have a lower rank)

Summarizer (II): more complicated summaries

Multi-Document Summaries

Many tasks in DUC require summaries of *multiple* documents:

- cross-document summary
- focused summary
- context-based summary (DUC 2005, 2006)

Solution

Additionally build *cross-document coreference chains* and summarize using a *fuzzy cluster graph algorithm*.

- For focused and context-based summaries, only use those chains that connect the question(s) with the documents (even if they have a lower rank)

Example for a Focused Summary generated by ERSS

"Who is Stephen Hawking?"

Hawking, 56, is the Lucasian Professor of Mathematics at Cambridge, a post once held by Sir Isaac Newton. Hawking, 56, suffers from Lou Gehrig's Disease, which affects his motor skills, and speaks by touching a computer screen that translates his words through an electronic synthesizers. Stephen Hawking, the Cambridge University physicist, is renowned for his brains. Hawking, a professor of physics an mathematics at Cambridge University in England, has gained immense celebrity, written a best-selling book, fathered three children, and done a huge amount for the public image of disability. Hawking, Mr. Big Bang Theory, has devoted his life to solving the mystery of how the universe started and where it's headed.

Example for a context-based summary (Excerpt)

Question

What countries are or have been involved in land or water boundary disputes with each other over oil resources or exploration? How have disputes been resolved, or towards what kind of resolution are the countries moving? What other factors affect the disputes?

System summary (first ∼70 words of 250 total)

The ministers of Asean - grouping Brunei, Indonesia, Malaysia, the Philippines, Singapore and Thailand - raised the Spratlys issue at a meeting yesterday with Qian Qichen, their Chinese counterpart. The meeting takes place against a backdrop of the continuing territorial disputes involving three Asean members - China, Vietnam and Taiwan - over the Spratley Islands in the South China Sea, a quarrel which could deteriorate shortly with the expected start of oil exploration in the area. . .

Example for a context-based summary (Excerpt)

Question

What countries are or have been involved in land or water boundary disputes with each other over oil resources or exploration? How have disputes been resolved, or towards what kind of resolution are the countries moving? What other factors affect the disputes?

System summary (first ∼70 words of 250 total)

The ministers of Asean - grouping Brunei, Indonesia, Malaysia, the Philippines, Singapore and Thailand - raised the Spratlys issue at a meeting yesterday with Qian Qichen, their Chinese counterpart. The meeting takes place against a backdrop of the continuing territorial disputes involving three Asean members - China, Vietnam and Taiwan - over the Spratley Islands in the South China Sea, a quarrel which could deteriorate shortly with the expected start of oil exploration in the area. . .

How can we evaluate summaries?

Problem

A summary is not right or wrong. Hard to find criterias.

Intrinsic

- Compare with model summaries
- Compare with source text
- Look solely at summary

Extrinsic

- Regarding external task
- Example: Use summary to cook a meal

Manual

- Subjective view
- High costs (40 systems X 50 clusters X 2 assessors = 4000 summaries)

Automatic

- High availability (during development)
- Repeatable and fast

How can we evaluate summaries?

Problem

A summary is not right or wrong. Hard to find criterias.

Intrinsic

- Compare with model summaries
- Compare with source text
- Look solely at summary

Extrinsic

- Regarding external task
- Example: Use summary to cook a meal

Manual

- Subjective view
- High costs (40 systems X 50 clusters X 2 assessors = 4000 summaries)

Automatic

- High availability (during development)
- Repeatable and fast

How can we evaluate summaries?

Problem

A summary is not right or wrong. Hard to find criterias.

Intrinsic

- Compare with model summaries
- Compare with source text
- Look solely at summary

Extrinsic

- Regarding external task
- Example: Use summary to cook a meal

Manual

- Subjective view
- High costs (40 systems X 50 clusters X 2 assessors = 4000 summaries)

Automatic

- High availability (during development)
- Repeatable and fast

Manual Measures

Summary Evaluation Environment: Linguistic quality

- Grammaticality
- Non-redundancy
- Referential clarity
- Focus
- Structure & Coherence

Responsiveness (2005)

- Pseudo-extrinsic
- How well was the question answered
- Form & Content
- In relation to the other systems' summaries

Manual Measures

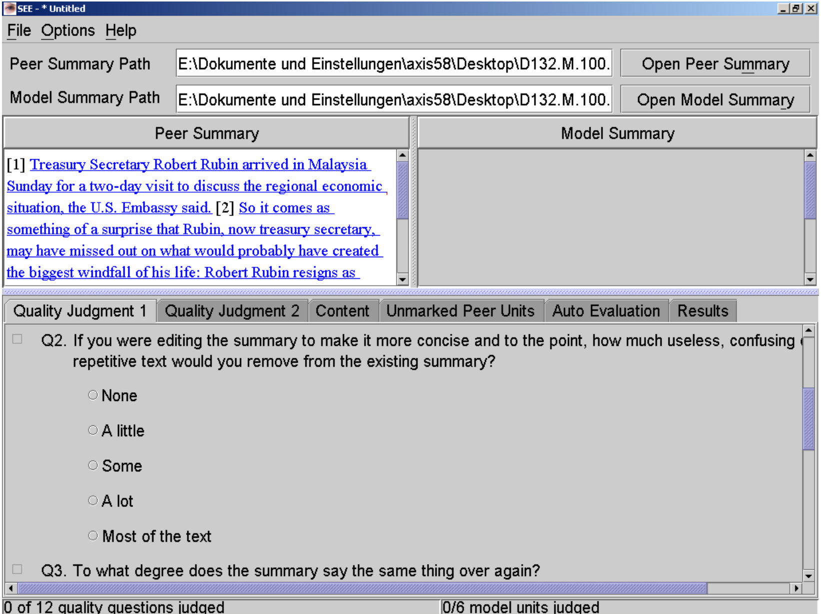
Summary Evaluation Environment: Linguistic quality

- Grammaticality
- Non-redundancy
- Referential clarity
- Focus
- Structure & Coherence

Responsiveness (2005)

- Pseudo-extrinsic
- How well was the question answered
- Form & Content
- In relation to the other systems' summaries

Manual Measures: SEE – Quality evaluation



Automatic Measures: ROUGE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

measures n-gram overlap between a peer and a set of reference summaries.

Definition

$$ROUGE_n = \frac{\sum_{C \in ModelUnits} \sum_{\text{n-gram} \in C} Count_{match}(\text{n-gram})}{\sum_{C \in ModelUnits} \sum_{\text{n-gram} \in C} Count(\text{n-gram})}$$

$ROUGE_{SU4} = ROUGE_2$ with skip of max. 4 words between two 2-grams

$ROUGE_2 / ROUGE_{SU4}$

S1 police killed the gunman

S2 police stopped the gunman

Evaluation: ERSS Results

DUC 2004

26 systems from 25 different groups, both industry and academic.
Evaluation performed by NIST (see `http://duc.nist.gov`).

ROUGE Results

Task 2: Cross-Document Common Topic Summaries

- *Best:* 0.38, *Worst:* 0.24, *Average:* 0.34, **ERSS:** 0.36
- ERSS statistically indistinguishable from top system within a 0.05 confidence level

Task 5: Focused Summaries

- *Best:* 0.35, *Worst:* 0.26, *Average:* 0.31, **ERSS:** 0.33
- same as above

Similar results for all other tasks.

Automatic Measures: Pyramids & Basic Elements

Driving force

- Scores of systems are not distinguishable.
- Only exact matches count. Abstractions are ignored.

Pyramids

- Comparing content units (not n-grams) of peer and models.
- Chunks occuring in more models get higher points.
- Needs manual annotation of peers **and** models.

Basic Elements

- Peer and Model summaries are parsed, extracting general relations between words of a sentence.
- Compute overlap of extracted **Head-Modifier-Relation**-Tripels between peer and models.

$\Rightarrow$ Peers don't have to be annotated by hand!

Automatic Measures: Pyramids & Basic Elements

Driving force

- Scores of systems are not distinguishable.
- Only exact matches count. Abstractions are ignored.

Pyramids

- Comparing content units (not n-grams) of peer and models.
- Chunks occuring in more models get higher points.
- Needs manual annotation of peers **and** models.

Basic Elements

- Peer and Model summaries are parsed, extracting general relations between words of a sentence.
- Compute overlap of extracted **Head-Modifier-Relation**-Tripels between peer and models.

$\Rightarrow$ Peers don't have to be annotated by hand!

Automatic Measures: Pyramids & Basic Elements

Driving force

- Scores of systems are not distinguishable.
- Only exact matches count. Abstractions are ignored.

Pyramids

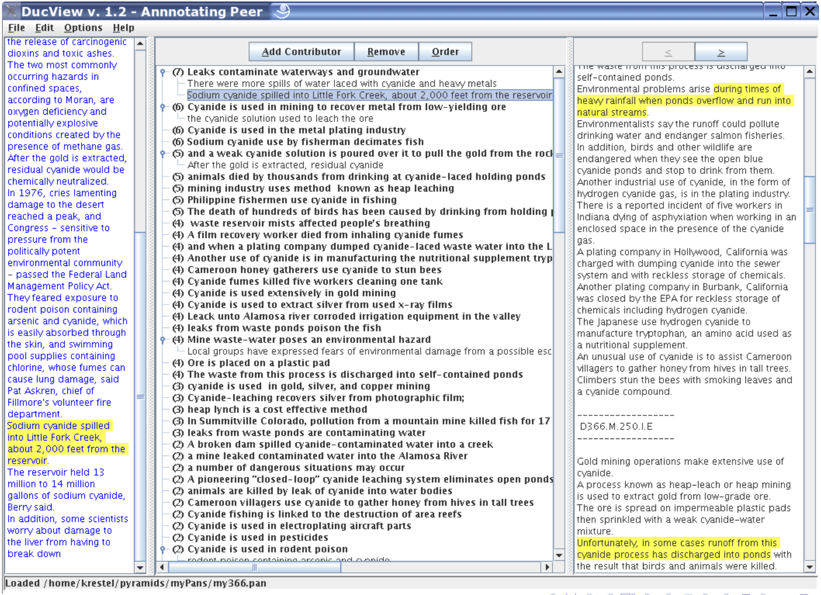
- Comparing content units (not n-grams) of peer and models.
- Chunks occuring in more models get higher points.
- Needs manual annotation of peers **and** models.

Basic Elements

- Peer and Model summaries are parsed, extracting general relations between words of a sentence.
- Compute overlap of extracted **Head-Modifier-Relation**-Tripels between peer and models.

$\Rightarrow$ Peers don't have to be annotated by hand!

Automatic Measures: Pyramids – GUI



Automatic Measures: Basic Elements

"Law enforcement officers from nine African countries are meeting in Nairobi this week to create a regional task force to fight international crime syndicates dealing in ivory, rhino horn, diamonds, arms, and drugs."

officers—enforcement—nn		syndicates—intern.—mod
officers—countries—from		meeting—officers—subj
nairobi—create—rel	create—week—subj	force—regional—mod
diamonds—arms—conj	countries—nine—nn	force—fight—rel
fight—force—subj	create—force—obj	syndicates—crime—nn
fight—syndicates—obj	horn—rhino—nn	ivory—horn—conj
horn—diamonds—conj	force—task—nn	arms—and—punc
arms—drugs—conj	meeting—nairobi—in	countries—african—nn

Basic Elements (head—modifier—relation) of the sentence shown on top

Summarization: Summary

Some Conclusions. . .

- Systems score very close to each other, partly due to the automatic ROUGE measure
- Automatic summaries still have a long way to go regarding style, coherence, and capabilities for abstraction
- Evaluation (almost) as difficult as the actual task

The Future?

Still, context-based summarization is promising:

- Do you really want to spent hours with Google? Scenario:
  - When writing a report/paper/memo on a certain topic,
  - a system will permanently scan your context,
  - retrieve documents pertaining to your topic,
  - and propose (hopefully relevant) information by itself
- Prediction: This will eventually find its way into Email clients, Word processors, Web browsers, etc.

[cf. Witte 2004 (IIWeb), Witte et al. 2005 (Semantic Desktop)]

Summarization: Summary

Some Conclusions...

- Systems score very close to each other, partly due to the automatic ROUGE measure
- Automatic summaries still have a long way to go regarding style, coherence, and capabilities for abstraction
- Evaluation (almost) as difficult as the actual task

The Future?

Still, context-based summarization is promising:

- Do you really want to spent hours with Google? Scenario:
  - When writing a report/paper/memo on a certain topic,
  - a system will permanently scan your context,
  - retrieve documents pertaining to your topic,
  - and propose (hopefully relevant) information by itself
- Prediction: This will eventually find its way into Email clients, Word processors, Web browsers, etc.

[cf. Witte 2004 (IIWeb), Witte et al. 2005 (Semantic Desktop)]

Opinion Mining

Motivation

Nowadays, there are countless websites containing huge amounts of product reviews written by consumers:

- E.g., `Amazon.com`, `Epinions.com`

But, like always, now there's too much information:

- You do not really want to spend more time on reading the reviews for a book than the book itself
- For a company, it is difficult to track all opinions regarding its product published on websites

Solution: use Text Mining to process and summarize opinions.

Opinion Mining: General Approach

Processing Steps

Detect Product Features: discussed in the review

Detect Opinions: regarding these features

Determine Polarity: of these opinions (positive? negative?)

Rank opinions: based on their strength (compare "so-so" vs. "desaster")

[cf. Popescu & Etzioni, HLT/EMNLP 2005]

Solution?

- Use *NE Detection* and *NP Chunking* to identify features
- Find opinions either within the NPs *"a very high resolution"*, or within adjacent constituents using parsing
- Match opinions (using stemming or lemmatization) against a lexicon containing polarity information
- Sort and rank opinions based on the number of reviews and strength

Opinion Mining: General Approach

Processing Steps

Detect Product Features: discussed in the review
Detect Opinions: regarding these features
Determine Polarity: of these opinions (positive? negative?)
Rank opinions: based on their strength (compare "so-so" vs. "desaster")

[cf. Popescu & Etzioni, HLT/EMNLP 2005]

Solution?

- Use *NE Detection* and *NP Chunking* to identify features
- Find opinions either within the NPs *"a very high resolution"*, or within adjacent constituents using parsing
- Match opinions (using stemming or lemmatization) against a lexicon containing polarity information
- Sort and rank opinions based on the number of reviews and strength

Question-Answering (QA)

Answering Factual Questions

A task somewhat related to automatic summarization is answering (factual) questions posed in natural languages.

Examples

From TREC-9 (2000):

- *Who invented the paper clip?*
- *Where is the Danube?*
- *How many years ago did the ship Titanic sink?*

The TREC Competition

The *Text REtrieval Conference* (TREC), also organized by NIST, includes a QA track.

QA Systems

Typical Approach in QA

Most QA systems roughly follow a three-step process:

- **Retrieval Step:** find documents from a set that might be relevant for the question
- **Answer Detection Step:** process retrieved documents to find possible answers
- **Reply Formulation Step:** create an answer in the required format (single NP, full sentence etc.)

How to find the answer?

Again, a multitude of approaches:

- **Syntactic:** find matching patterns or parse (sub-)trees (with some transformations) in both Q and A
- **Semantic:** transform both Q and A into a logical form and use inference to check consistency
- **Google:** plug the question into Google and select the answer with a syntactic strategy…

QA Systems

Typical Approach in QA

Most QA systems roughly follow a three-step process:

Retrieval Step: find documents from a set that might be relevant for the question

Answer Detection Step: process retrieved documents to find possible answers

Reply Formulation Step: create an answer in the required format (single NP, full sentence etc.)

How to find the answer?

Again, a multitude of approaches:

Syntactic: find matching patterns or parse (sub-)trees (with some transformations) in both Q and A

Semantic: transform both Q and A into a logical form and use inference to check consistency

Google: plug the question into Google and select the answer with a syntactic strategy. . .

Google does some QA. . .

Ask Google: When was Julius Caesar born?

Text Mining in the Biological Domain

Biological Research

Like in other disciplines, researchers and practitioners in biology

- need up-to-date information
- but have too much literature to cope with

Particular to Biology

- biological databases containing results of experiments
- manually curated databases
- central repositories for literature (PubMed/Medline/Entrez)

General Idea of our Work

Support researchers in biology, by information extraction (automatic curation suporrt) and combining NLP results with databases and end user's tools

Text Mining in the Biological Domain

Biological Research

Like in other disciplines, researchers and practitioners in biology

- need up-to-date information
- but have too much literature to cope with

Particular to Biology

- biological databases containing results of experiments
- manually curated databases
- central repositories for literature (PubMed/Medline/Entrez)

General Idea of our Work

Support researchers in biology, by information extraction (automatic curation suporrt) and combining NLP results with databases and end user's tools

Text Mining in the Biological Domain

Biological Research

Like in other disciplines, researchers and practitioners in biology

- need up-to-date information
- but have too much literature to cope with

Particular to Biology

- biological databases containing results of experiments
- manually curated databases
- central repositories for literature (PubMed/Medline/Entrez)

General Idea of our Work

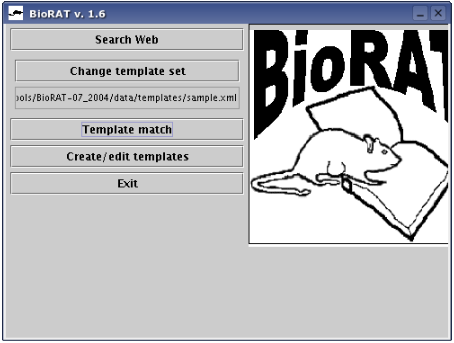
Support researchers in biology, by information extraction (automatic curation suporrt) and combining NLP results with databases and end user's tools

The BioRAT System

BioRAT

BioRAT is *a search engine and information extraction tool for biological research*

- developed at University College London (UCL) in cooperation with GlaxoSmithKline



BioRAT provides

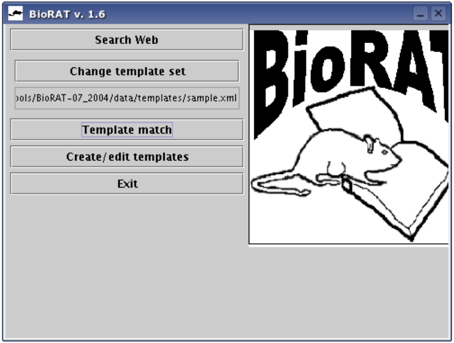
- a web spidering/information retrieval engine
- an information extraction system based on GATE
- a “template” design tool for IE patterns

The BioRAT System

BioRAT

BioRAT is *a search engine and information extraction tool for biological research*

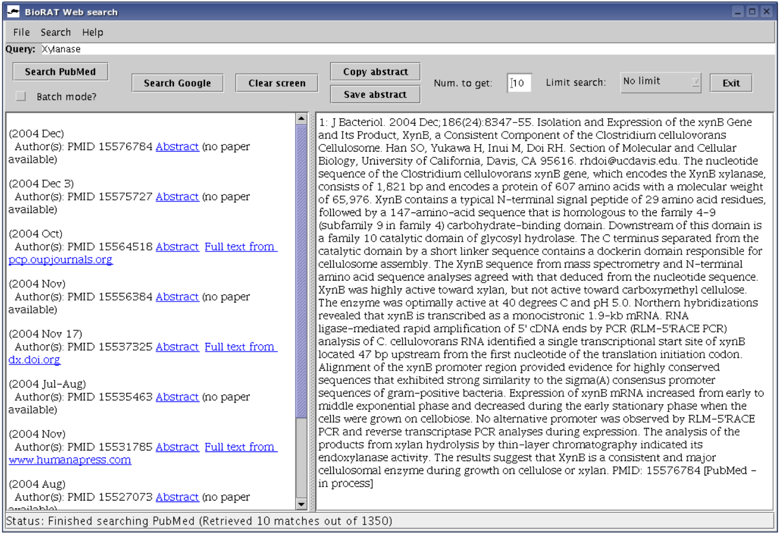
- developed at University College London (UCL) in cooperation with GlaxoSmithKline



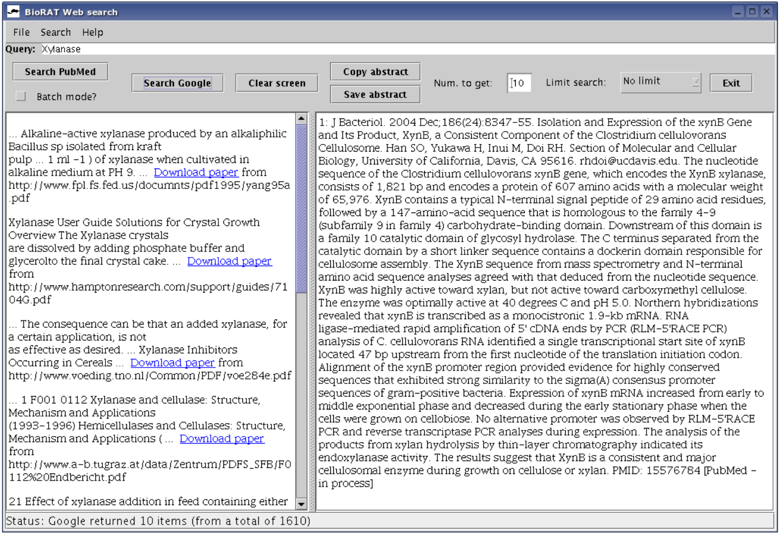
BioRAT provides

- a web spidering/information retrieval engine
- an information extraction system based on GATE
- a “template” design tool for IE patterns

BioRAT: Information Retrieval



BioRAT: Information Retrieval



BioRAT: Information Extraction

Template-based Extraction (actually regular expressions)

- Preprocessing provides *Tokens* and *POS tags*
- Gazetteering step uses lists derived from SwissProt and MeSH to annotate *entities* (genes, proteins, drugs, procedures, . . .)
- Templates (JAPE grammars) define patterns for extraction

Templates

- *Sample:* find pattern <noun> <prep> <drug/chemical>
- *DIP:* find protein-protein interactions

Example Grammar

```
Rule: sample1
Priority: 1000
(
({Token.category == NN}):block0
({Token.category == IN}):block1
({Lookup.majorType
   == "chemicals_and_drugs"}):block2
) --> (add result...)
```

BioRAT: Information Extraction

Template-based Extraction (actually regular expressions)

- Preprocessing provides *Tokens* and *POS tags*
- Gazetteering step uses lists derived from SwissProt and MeSH to annotate *entities* (genes, proteins, drugs, procedures, ...)
- Templates (JAPE grammars) define patterns for extraction

Templates

- *Sample:* find pattern <noun> <prep> <drug/chemical>
- *DIP:* find protein-protein interactions

Example Grammar

```
Rule: sample1
Priority: 1000
(
({Token.category == NN}):block0
({Token.category == IN}):block1
({Lookup.majorType
   == "chemicals_and_drugs"}):block2
) --> (add result...)
```

BioRAT: Information Extraction

Template-based Extraction (actually regular expressions)

- Preprocessing provides *Tokens* and *POS tags*
- Gazetteering step uses lists derived from SwissProt and MeSH to annotate *entities* (genes, proteins, drugs, procedures, ...)
- Templates (JAPE grammars) define patterns for extraction

Templates

- *Sample:* find pattern <noun> <prep> <drug/chemical>
- *DIP:* find protein-protein interactions

Example Grammar

```
Rule: sample1
Priority: 1000
(
({Token.category == NN}):block0
({Token.category == IN}):block1
({Lookup.majorType
   == "chemicals_and_drugs"}):block2
) --> (add result...)
```

BioRAT: Extraction Results

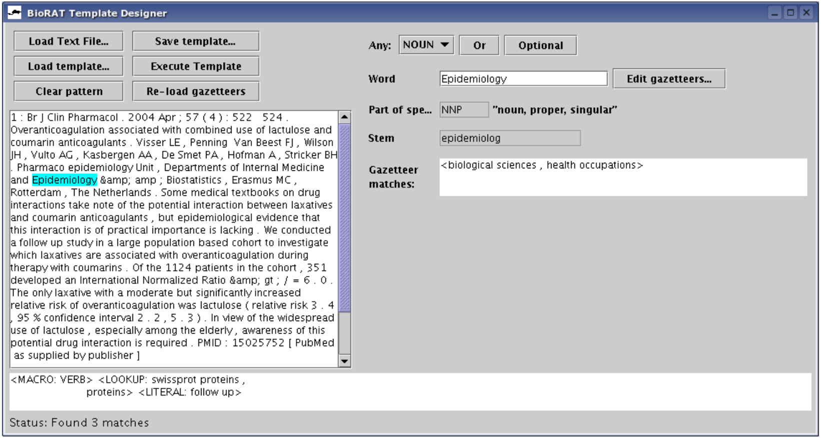
BioRAT Template results display

Text file	Rule	Position	Length	block0	block1	block2	Context
...:0:0:0p1/BioCase/Xylanase3.txt	sample1	6.61%	3	addition	of	disulfide	In practice, thermal stability of xylanases has been improved by addition of disulfide bridges (1115), engineering polar side chains into a protein surface (16), increasing aromatic interactions (17), and other amino acid substitutions (11 21).
...:0:0:0p1/BioCase/Xylanase3.txt	sample1	9.61%	3	characterization	of	proteins	The advent of electrospray ionization (ESI) has opened up mass spectrometry (MS) as one of the most powerful analytical techniques for structural and functional characterization of proteins (28).
...:0:0:0p1/BioCase/Xylanase3.txt	sample1	10.57%	3	configuration	for	protein	ESI in combination with a Fourier transform ion cyclotron resonance (FT ICR) mass analyzer (30, 31) represents the most powerful instrumental configuration for protein analyses.
...:0:0:0p1/BioCase/Xylanase3.txt	sample1	28.98%	3	presence	of	acetonitrile	In contrast, clear changes appeared in the presence of acetonitrile.
...:0:0:0p1/BioCase/Xylanase3.txt	sample1	38.9%	3	number	of	disulfide	Hence, to verify the number of disulfide bridges (i.e.
...:0:0:0p1/BioCase/Xylanase3.txt	sample1	41.52%	3	number	of	disulfide	disulfide reduced), and 2red (two disulfides reduced; in DB1 only) were used to assign the correct number of disulfide bridges, i.e.
...:0:0:0p1/BioCase/Xylanase3.txt	sample1	46.73%	3	ratio	in	solution	D(t)) Htotal a(Hfaste kfastt + Hslowekslowt) in which Htotal is a total number of exchangeable hydrogens in TRX II (341 based on the sequence), a is a D/H ratio in solution (0.95), Hfast and Hslow are the numbers for fast and slow exchanging hydrogens, and kfast and kslow are the corresponding pseudo first order rate constants.
...:0:0:0p1/BioCase/Xylanase3.txt	sample1	60.64%	3	number	of	disulfides	Standard deviation calculations between the theoretical and the experimental isotopic distributions confirmed the expected number of disulfides in DS2, DS5, and DB1 mutants (i.e.

Close

Status:

BioRAT: Template Design Tool



BioRAT: Some Observations

BioRAT Performance

- Authors report 39% recall and 48% precision on the DIP task
- Comparable to the SUISEKI system (Blaschke et al.), which is statistics-based

System Design

More interestingly,

- BioRAT is rather “low” on NLP knowledge,
- yet surprisingly useful for Biologists

Interesting pattern:

- NLP is “just another” system component
- Users (Biologists) are empowered: no need for computational linguists to add/modify/remove grammar rules

BioRAT: Some Observations

BioRAT Performance

- Authors report 39% recall and 48% precision on the DIP task
- Comparable to the SUISEKI system (Blaschke et al.), which is statistics-based

System Design

More interestingly,

- BioRAT is rather "low" on NLP knowledge,
- yet surprisingly useful for Biologists

Interesting pattern:

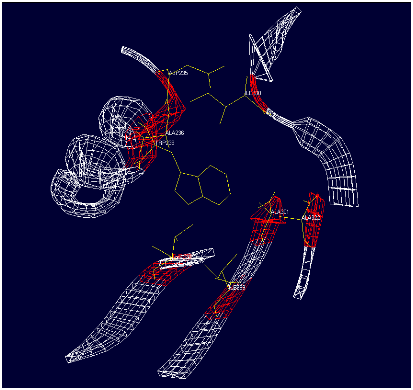
- NLP is "just another" system component
- Users (Biologists) are empowered: no need for computational linguists to add/modify/remove grammar rules

MutationMiner: Motivation

Challenge

Support Bio-Engineers designing proteins:

- need up-to-date, relevant information from research literature
- need for automated updates
- need for integration with structural biology tools



MutationMiner: Background

Existing Resources

Protein Mutant Database

Center for Information Biology and DNA Data Bank of Japan

National Institute of Genetics

- 1999: authors quote 3-year backlog of unprocessed publications
- Funding for manual curation limited / declining
- Manual data submission is slow and incomplete
- Sequence and structure databases expanding
- New techniques: Directed Evolution
- New alignment algorithms: e.g. Fugue, Muscle

Protein Mutant Database

Example PMD Entry (manually curated)

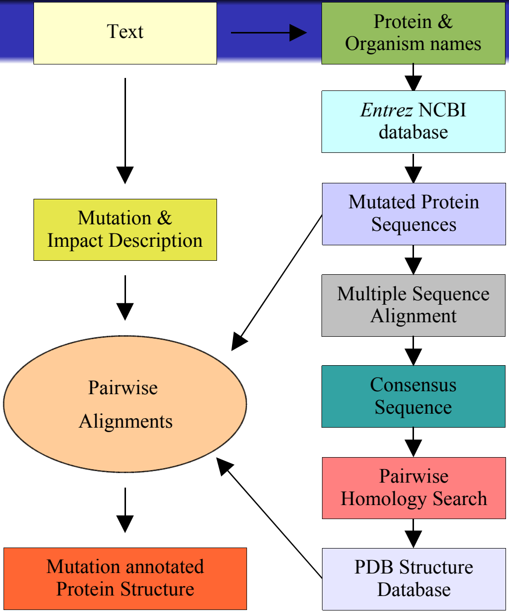
```
ENTRY   A931290 - Artificial 1921240
AUTHORS     Lee Y.-E., Lowe S.E., Henrissat B. & Zeikus J.G.
JOURNAL     J.Bacteriol. (1993) 175(18), 5890-5898 [LINK-TO-MEDLINE]
TITLE  Characterization of the active site and thermostability
    regions of endoxylanase from Thermoanaerobacterium saccharolyticum
CROSS-REFERENCE A48490 [LINK TO PIR "A48490"] No PDB-LINK for "A48490"
PROTEIN         Endoxylanase (endo-1,4-beta-xylanase) #EC3.2.1.8
SOURCE  Thermoanaerobacterium saccharolyticum
N-TERMINAL      MMKNN
EXPRESSION-SYSTEM       Escherichia coli
CHANGE  Asp 537 Asn     FUNCTION        Endoxylanase activity [0]
CHANGE  Glu 541 Gln     FUNCTION        Endoxylanase activity [=]
CHANGE  His 572 Asn     FUNCTION        Endoxylanase activity [=]
CHANGE  Glu 600 Gln     FUNCTION        Endoxylanase activity [0]
CHANGE  Asp 602 Asn     FUNCTION        Endoxylanase activity [0]
```

MutationMiner: Goal

Aim

Develop a system to

- extract annotations regarding mutations from full-text papers; and
- legitimately link them to protein structure visualizations



MutationMiner NLP: Input

Input documents are typically in HTML, XML, or PDF formats:

9556 *Biochemistry* **2004**, *43,* 9556−9566

Characterization of Mutant Xylanases Using Fourier Transform Ion Cyclotron Resonance Mass Spectrometry: Stabilizing Contributions of Disulfide Bridges and N-Terminal Extensions[†]

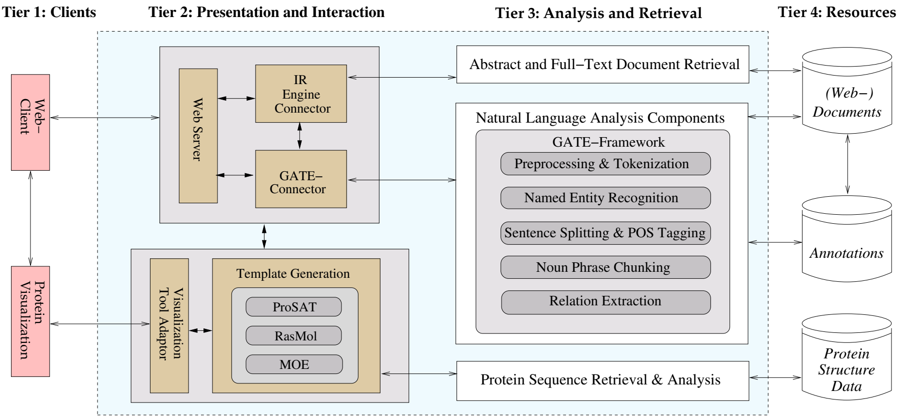
Janne Jänis,[‡] Ossi Turunen,[§] Matti Leisola,[§] Peter J. Derrick,[||] Juha Rouvinen,[‡] and Pirjo Vainiotalo*[,‡]

Department of Chemistry, University of Joensuu, P.O. Box 111, FI-80101 Joensuu, Finland, Laboratory of Bioprocess Engineering, Helsinki University of Technology, P.O. Box 6100, FI-02015 HUT, Finland, and Department of Chemistry, Institute of Mass Spectrometry, University of Warwick, Coventry CV4 7AL, United Kingdom

Received February 27, 2004; Revised Manuscript Received May 17, 2004

ABSTRACT: Structural properties and thermal stability of *Trichoderma reesei* endo-1,4-β-xylanase II (TRX II) and its three recombinant mutants were characterized using electrospray ionization Fourier transform ion cyclotron resonance (ESI FT-ICR) mass spectrometry and hydrogen/deuterium (H/D) exchange reactions. TRX II has been previously stabilized by a disulfide bridge C110−C154 and other site-directed mutations (TRX II mutants DS2 and DS5). Very recently, a highly thermostable mutant was introduced by combining mutations of DS5 with an N-terminal disulfide bridge C2−C28 (mutant DB1). Accurate

MutationMiner Architecture



MutationMiner: NLP Subsystem

NLP Steps

Tokenization split input into tokens

Gazetteering using lists derived from Swissprot and MeSH

Named Entity recognition find proteins, mutations, organisms

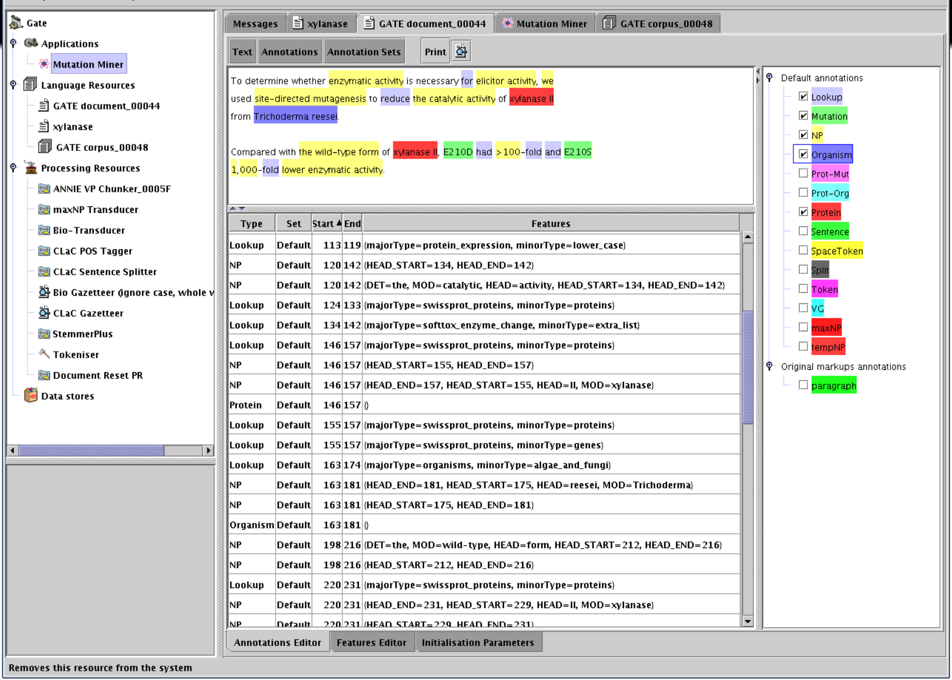
Sentence splitting sentence boundary detection

POS tagging add part-of-speech tags

NP Chunking e.g. *the/DET catalytic/MOD activity/HEAD*

Relation detection find protein-organism and protein-mutation relations

"Wild-type and mutated xylanase II proteins (termed E210D and E210S) were expressed in S. cerevisiae grown in liquid culture."



Gate
Applications
Mutation Miner
Language Resources
GATE document_00044
xylanase
GATE corpus_00048
Processing Resources
ANNIE VP Chunker_0005F
maxNP Transducer
Bio-Transducer
CLaC POS Tagger
CLaC Sentence Splitter
Bio Gazetteer (ignore case, whole v
CLaC Gazetteer
StemmerPlus
Tokeniser
Document Reset PR
Data stores
Messages
xylanase
GATE document_00044
Mutation Miner
GATE corpus_00048
Text
Annotations
Annotation Sets
Print
To determine whether enzymatic activity is necessary for elicitor activity, we used site-directed mutagenesis to reduce the catalytic activity of xylanase II from Trichoderma reesei.
Compared with the wild-type form of xylanase II, E210D had >100-fold and E210S 1,000-fold lower enzymatic activity.
Type | Set | Start | End | Features
Lookup | Default | 113 | 119 | {majorType=protein_expression, minorType=lower_case}
NP | Default | 120 | 142 | {HEAD_START=134, HEAD_END=142}
NP | Default | 120 | 142 | {DET=the, MOD=catalytic, HEAD=activity, HEAD_START=134, HEAD_END=142}
Lookup | Default | 124 | 133 | {majorType=swissprot_proteins, minorType=proteins}
Lookup | Default | 134 | 142 | {majorType=softtox_enzyme_change, minorType=extra_list}
Lookup | Default | 146 | 157 | {majorType=swissprot_proteins, minorType=proteins}
NP | Default | 146 | 157 | {HEAD_START=155, HEAD_END=157}
NP | Default | 146 | 157 | {HEAD_END=157, HEAD_START=155, HEAD=II, MOD=xylanase}
Protein | Default | 146 | 157 | {}
Lookup | Default | 155 | 157 | {majorType=swissprot_proteins, minorType=proteins}
Lookup | Default | 155 | 157 | {majorType=swissprot_proteins, minorType=genes}
Lookup | Default | 163 | 174 | {majorType=organisms, minorType=algae_and_fungi}
NP | Default | 163 | 181 | {HEAD_END=181, HEAD_START=175, HEAD=reesei, MOD=Trichoderma}
NP | Default | 163 | 181 | {HEAD_START=175, HEAD_END=181}
Organism | Default | 163 | 181 | {}
NP | Default | 198 | 216 | {DET=the, MOD=wild-type, HEAD=form, HEAD_START=212, HEAD_END=216}
NP | Default | 198 | 216 | {HEAD_START=212, HEAD_END=216}
Lookup | Default | 220 | 231 | {majorType=swissprot_proteins, minorType=proteins}
NP | Default | 220 | 231 | {HEAD_END=231, HEAD_START=229, HEAD=II, MOD=xylanase}
NP | Default | 220 | 231 | {HEAD_START=229, HEAD_END=231}
Default annotations
Lookup
Mutation
NP
Organism
Prot-Mut
Prot-Org
Protein
Sentence
SpaceToken
Split
Token
VG
maxNP
tempNP
Original markups annotations
paragraph
Annotations Editor
Features Editor
Initialisation Parameters
Removes this resource from the system

MutationMiner: Further Processing

Results

- Results are information about *Proteins*, *Organisms*, and *Mutations*, along with *context information*

Next Step

- These results could already be used to (semi-)automatically curate PMD entries
- But remember the original goal: integrate results into end user's tools
- Needs data that can be further processed by bioinformatics tools

Thus, we need to find the corresponding real-world entities in biological databases: *amino acid sequences*

MutationMiner: Further Processing

Results

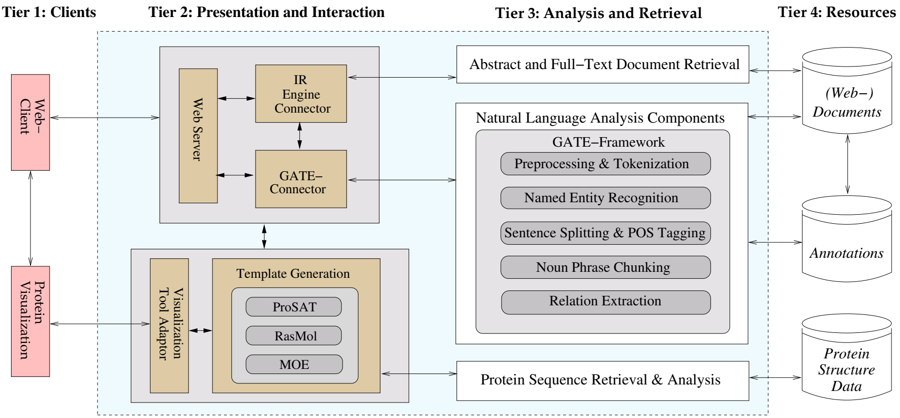
- Results are information about *Proteins*, *Organisms*, and *Mutations*, along with *context information*

Next Step

- These results could already be used to (semi-)automatically curate PMD entries
- But remember the original goal: integrate results into end user's tools
- Needs data that can be further processed by bioinformatics tools

Thus, we need to find the corresponding real-world entities in biological databases: *amino acid sequences*

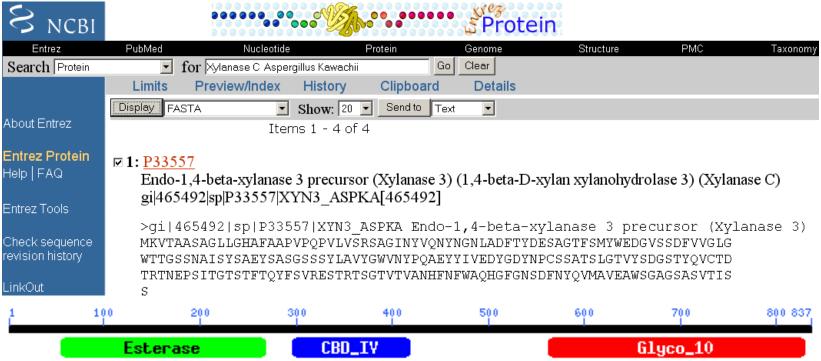
MutationMiner Architecture



MutationMiner: Sequence Retrieval

Sequence Retrieval

- Retrieval of FASTA formatted sequences for protein accessions obtained by NLP analysis of texts
- Obtained through querying *Entrez* NCBI database (E-fetch)



MutationMiner: Sequence Analysis

CLUSTAL W (1.82) multiple sequence alignment

```
            10        20        30        40        50
1  YRP-TGTYK-GTVKSDGGTYDIYTTTRYNAPSIDGD-RTTFTQYWSVRQS  gi|139865|sp|P09850|XYNA_BACCI
1  YRP-TGTYK-GTVKSDGGTYDIYTTTRYNAPSIDGD-RTTFTQYWSVRQS  gi|640242|pdb|1BCX|Xylanase
1  YRP-TGTYK-GTVTSDGGTYDVYQTTRVNAPSVEG--TKTFNQYWSVRQS  gi|17942986|pdb|1HIX|BChain
1  YRP-TGAYK-GSFYADGGTYDIYETTRVNQPSIIG--IATFKQYWSVRQT  gi|1351447|sp|P00694|XYNA_BACP
1  YNPSTGATKLGEVTSDGSVYDIYRTQRVNQPSIIG--TATFYQYWSVRRN  gi|549461|sp|P36217|XYN2TRIRE
1  YNPCSSATSLGTVYSDGSTYQVCTDTRTNEPSITG--TSTFTQYFSVRES  gi|465492|sp|P33557|XYN3_ASPKA
1  RGVPLDCVGFQSHLIVG---QVPGDFRQNLQRFADLGVDVRITELDIRMR  gi|121856|sp|P07986|GUX_CELFI
1  RGVPIDCVGFQSHFNSGS--PYNSNFRTTLQNFAALGVDVAITELDIQG-  gi|6226911|sp|P26514|XYNA_STRL
1  RGVPIDGVGFQCHFINGMSPEYLASIDQNIKRYAEIGVIVSFTEIDIRIP  gi|139886|sp|P10478|XYNZ_CLOTM
```

- sequence analyzed and sliced in regions using CDD *(conserved domain database)* search tools
- iterative removal of outlying sequences through statistical scoring using *Alistat*
- generation of a consensus sequence using a HMM (HMMER)
- locate NLP-extracted mutations on sequence

Sequence Analysis Results

Amino Acid Sequence Analysis

- We now have a set of filtered sequences, describing proteins and their mutations
- Still not a very intuitive presentation of results

Suitable visualization needed!

3D-Structure Visualization

- Idea: map mutations of proteins directly to a 3D-visualization of their structural representation
- However, for this we need to find a 3D-model (homolog)

Solution: access Protein Data Bank (PDB) using BLAST for a suitable 3D-model and map NLP results onto this structure

Sequence Analysis Results

Amino Acid Sequence Analysis

- We now have a set of filtered sequences, describing proteins and their mutations
- Still not a very intuitive presentation of results

Suitable visualization needed!

3D-Structure Visualization

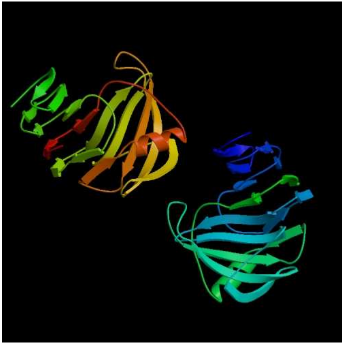
- Idea: map mutations of proteins directly to a 3D-visualization of their structural representation
- However, for this we need to find a 3D-model (homolog)

Solution: access Protein Data Bank (PDB) using BLAST for a suitable 3D-model and map NLP results onto this structure

MutationMiner: PDB Structure Retrieval

Title	Crystallographic Analyses Of Family 11 Endo–1,4-Xylanase Xyl1
Classification	Hydrolase
Compound	Mol_Id: 1; Molecule: Endo-1,4–Xylanase; Chain: A, B; Ec: 3.2.1.8;
Exp. Method	X-ray Diffraction

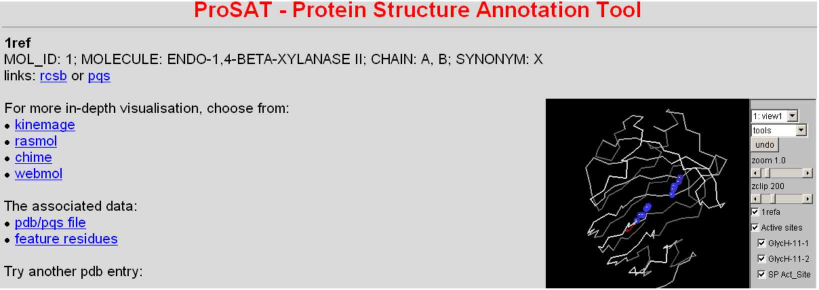
```
JRNL TITL 2 ENDO-[BETA]-1,4-XYLANASE XYL1 FROM STREPTOMYCES SP. S38
JRNL REF  ACTA CRYSTALLOGR.,SECT.D V. 57 1813 2001
JRNL REFN ASTM ABCRE6 DK ISSN 0907-4449
...
DBREF  1HIX A    1   190  TREMBL Q59962   Q59962
DBREF  1HIX B    1   190  TREMBL Q59962   Q59962
...
ATOM 1 N   ILE A 4 48.459 19.245 17.075 1.00 24.52 N
ATOM 2 CA  ILE A 4 47.132 19.306 17.680 1.00 50.98 C
ATOM 3 C   ILE A 4 47.116 18.686 19.079 1.00 49.94 C
ATOM 4 O   ILE A 4 48.009 17.936 19.465 1.00 70.83 O
ATOM 5 CB  ILE A 4 46.042 18.612 16.837 1.00 50.51 C
ATOM 6 CG1 ILE A 4 46.419 17.217 16.338 1.00 51.09 C
ATOM 7 CG2 ILE A 4 45.613 19.514 15.687 1.00 54.39 C
ATOM 8 CD1 ILE A 4 46.397 17.045 14.836 1.00 46.72 C
ATOM 9 N   THR A 5 46.077 19.024 19.828 1.00 40.65 N
...
MASTER   321  0  0  2  28  0  0  9 3077  2  0 30
END
```

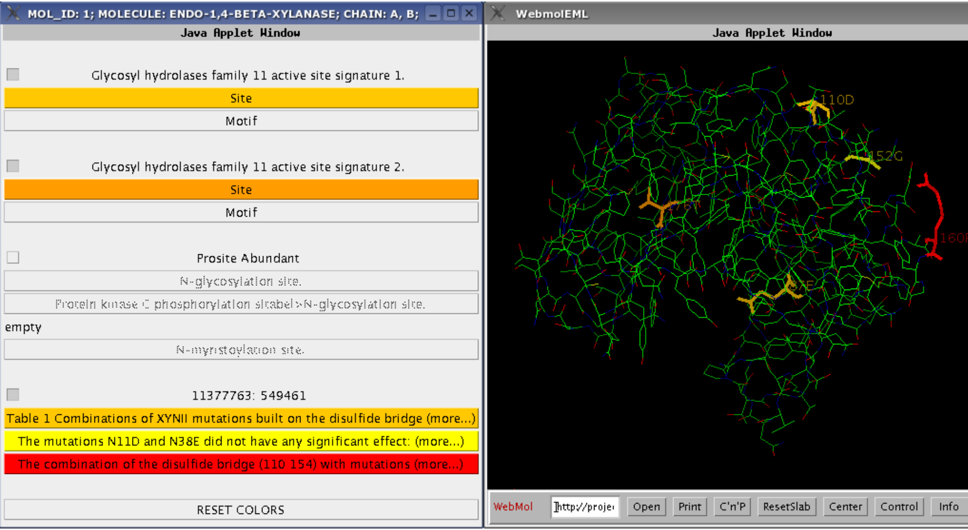


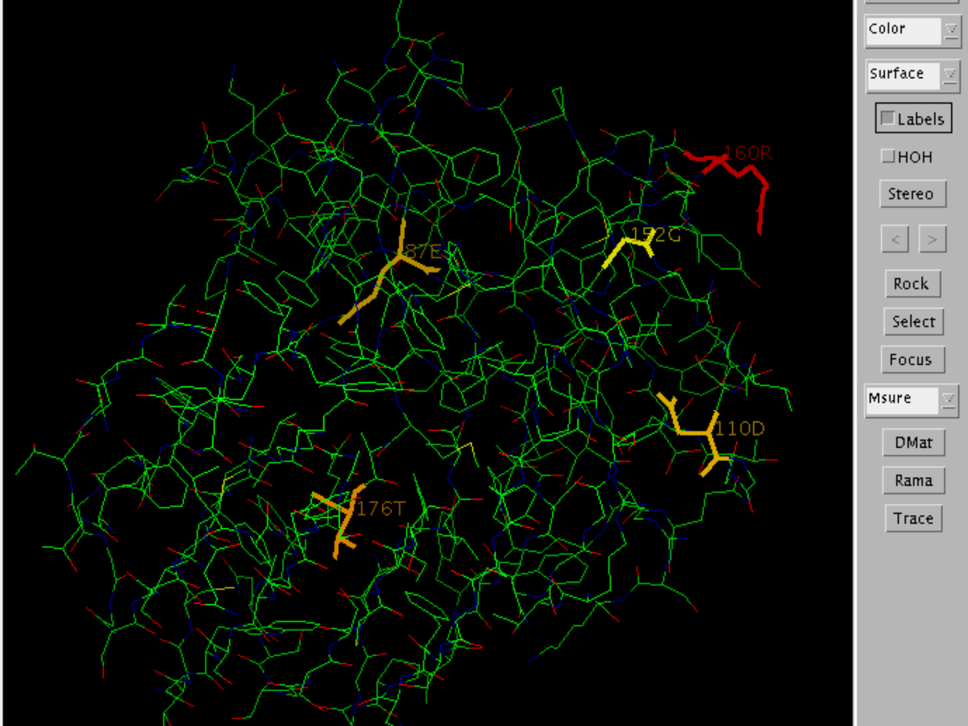
MutationMiner: Visualization

Visualization Tools

- ProSAT is a tool to map SwissProt sequence features and Prosite patterns on to a 3D structure of a protein.
- We are now able to upload the 3D structure together with our textual annotations for rendering using a Webmol interface




MOL_ID: 1; MOLECULE: ENDO-1,4-BETA-XYLANASE; CHAIN: A, B;
Java Applet Window
Glycosyl hydrolases family 11 active site signature 1.
Site
Motif
Glycosyl hydrolases family 11 active site signature 2.
Site
Motif
Prosite Abundant
N-glycosylation site.
empty
N-myristoylation site.
11377763: 549461
Table 1 Combinations of XYNII mutations built on the disulfide bridge (more...)
The mutations N11D and N38E did not have any significant effect: (more...)
The combination of the disulfide bridge (110 154) with mutations (more...)
RESET COLORS
WebmolEML
Java Applet Window
110D
WebMol
Open
Print
C'n'P
ResetSlab
Center
Control
Info


160R
152G
87E
110D
176T
Color
Surface
Labels
HOH
Stereo
<
>
Rock
Select
Focus
Msure
DMat
Rama
Trace

Implementation and Evaluation

Implementation

NLP subsystem implemented using the GATE architecture

Testing Corpus

First evaluation performed on research literature concerning the *Xylanase* protein family (20 papers)

NLP subsystem partial evaluation results

	Abstract only		Full paper	
	Protein/Organism	Mutations	Protein/Organism	Mutations
Precision	0.88	1.00	0.91	0.84
Recall	0.71	0.85	0.46	0.97
F-Measure	0.79	0.92	0.61	0.90

Implementation and Evaluation

Implementation

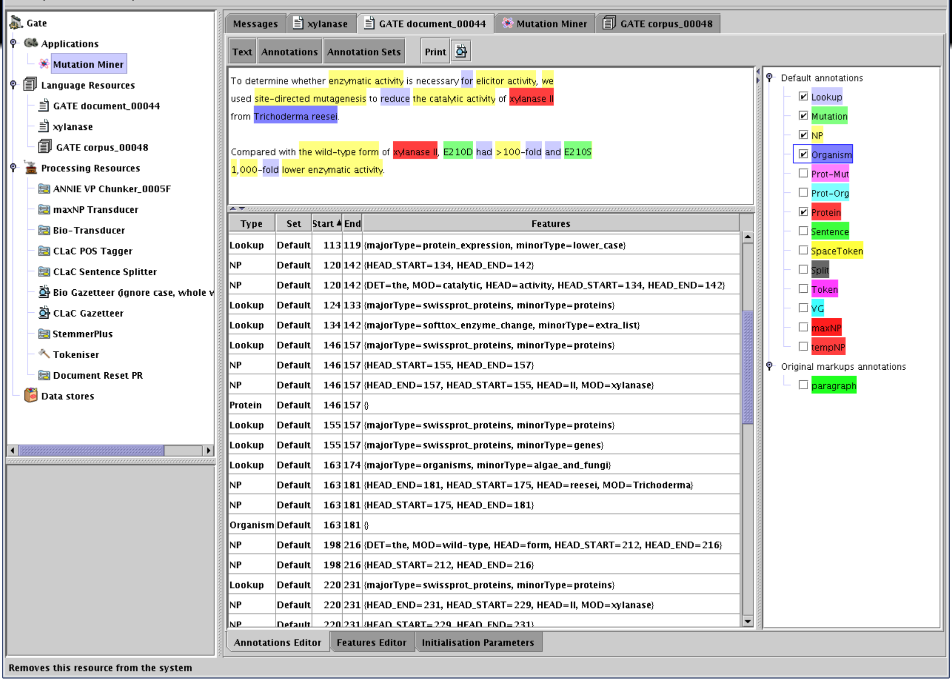
NLP subsystem implemented using the GATE architecture

Testing Corpus

First evaluation performed on research literature concerning the *Xylanase* protein family (20 papers)

NLP subsystem partial evaluation results

	Abstract only		Full paper	
	Protein/Organism	Mutations	Protein/Organism	Mutations
Precision	0.88	1.00	0.91	0.84
Recall	0.71	0.85	0.46	0.97
F-Measure	0.79	0.92	0.61	0.90


Gate
Applications
Mutation Miner
Language Resources
GATE document_00044
xylanase
GATE corpus_00048
Processing Resources
ANNIE VP Chunker_0005F
maxNP Transducer
Bio-Transducer
CLaC POS Tagger
CLaC Sentence Splitter
Bio Gazetteer (ignore case, whole v
CLaC Gazetteer
StemmerPlus
Tokeniser
Document Reset PR
Data stores
Messages
xylanase
GATE document_00044
Mutation Miner
GATE corpus_00048
Text
Annotations
Annotation Sets
Print
To determine whether enzymatic activity is necessary for elicitor activity, we used site-directed mutagenesis to reduce the catalytic activity of xylanase II from Trichoderma reesei.
Compared with the wild-type form of xylanase II, E210D had >100-fold and E210S 1,000-fold lower enzymatic activity.
Type | Set | Start | End | Features
Lookup | Default | 113 | 119 | {majorType=protein_expression, minorType=lower_case}
NP | Default | 120 | 142 | {HEAD_START=134, HEAD_END=142}
NP | Default | 120 | 142 | {DET=the, MOD=catalytic, HEAD=activity, HEAD_START=134, HEAD_END=142}
Lookup | Default | 124 | 133 | {majorType=swissprot_proteins, minorType=proteins}
Lookup | Default | 134 | 142 | {majorType=softtox_enzyme_change, minorType=extra_list}
Lookup | Default | 146 | 157 | {majorType=swissprot_proteins, minorType=proteins}
NP | Default | 146 | 157 | {HEAD_START=155, HEAD_END=157}
NP | Default | 146 | 157 | {HEAD_END=157, HEAD_START=155, HEAD=II, MOD=xylanase}
Protein | Default | 146 | 157 | {}
Lookup | Default | 155 | 157 | {majorType=swissprot_proteins, minorType=proteins}
Lookup | Default | 155 | 157 | {majorType=swissprot_proteins, minorType=genes}
Lookup | Default | 163 | 174 | {majorType=organisms, minorType=algae_and_fungi}
NP | Default | 163 | 181 | {HEAD_END=181, HEAD_START=175, HEAD=reesei, MOD=Trichoderma}
NP | Default | 163 | 181 | {HEAD_START=175, HEAD_END=181}
Organism | Default | 163 | 181 | {}
NP | Default | 198 | 216 | {DET=the, MOD=wild-type, HEAD=form, HEAD_START=212, HEAD_END=216}
NP | Default | 198 | 216 | {HEAD_START=212, HEAD_END=216}
Lookup | Default | 220 | 231 | {majorType=swissprot_proteins, minorType=proteins}
NP | Default | 220 | 231 | {HEAD_END=231, HEAD_START=229, HEAD=II, MOD=xylanase}
NP | Default | 220 | 231 | {HEAD_START=229, HEAD_END=231}
Default annotations
Lookup
Mutation
NP
Organism
Prot-Mut
Prot-Org
Protein
Sentence
SpaceToken
Split
Token
VG
maxNP
tempNP
Original markups annotations
paragraph
Annotations Editor
Features Editor
Initialisation Parameters
Removes this resource from the system

MutationMiner: Conclusions and Ongoing Work

Conclusions

- Integration of bio-NLP, bio-DBs, and bioinformatics tools is very promising and has high practical relevance

Current Work

- Analysis of *Dehalogenase*, *Biphenyl Dioxygenase*, and *Subtilisin*
- Application to human health related scenarios, like *BrcA* protein, which is involved in breast cancer

Future Work

- Extrinsic evaluation with domain specialists, both protein researchers and industry practitioners

MutationMiner: Conclusions and Ongoing Work

Conclusions

- Integration of bio-NLP, bio-DBs, and bioinformatics tools is very promising and has high practical relevance

Current Work

- Analysis of *Dehalogenase*, *Biphenyl Dioxygenase*, and *Subtilisin*
- Application to human health related scenarios, like *BrcA* protein, which is involved in breast cancer

Future Work

- Extrinsic evaluation with domain specialists, both protein researchers and industry practitioners

MutationMiner: Conclusions and Ongoing Work

Conclusions

- Integration of bio-NLP, bio-DBs, and bioinformatics tools is very promising and has high practical relevance

Current Work

- Analysis of *Dehalogenase*, *Biphenyl Dioxygenase*, and *Subtilisin*
- Application to human health related scenarios, like *BrcA* protein, which is involved in breast cancer

Future Work

- Extrinsic evaluation with domain specialists, both protein researchers and industry practitioners

References

General Text Mining Books

- Weiss, Indurkhya, Zhang, Damerau, *Text Mining: Predictive Methods for Analyzing Unstructured Information*, Springer, 2005.
- Sophia Ananiadou and John McNaught (Eds.), *Text Mining for Biology and Biomedicine*, Artech House, 2006.
- Inderjeet Mani, *Automatic Summarization*, John Benjamins B.V., 2001

Papers

- Cunningham, *Information Extraction, Automatic*, Encyclopedia of Language and Linguistics, 2005. `http://gate.ac.uk/sale/ell2/ie/`

References (II)

Conferences

- Automatic Summarization: Document Understanding Conference (DUC), `http://duc.nist.gov`
- Text REtrieval Conference: `http://trec.nist.gov/`
- Exploring Attitude and Affect in Text: Theories and Applications, AAAI Spring Symposium, 2004, AAAI Press, Technical Report SS-04-07

Related Literature

- Zhong & Liu (Eds.), *Intelligent Technologies for Information Analysis*, Springer, 2004
- Peter Morville, *Ambient Findability*, O'Reilly, 2006

And of course...

- `http://rene-witte.net`

Part V

Conclusions

21 Conclusions

Some conclusions

The present

- A complete semantic understanding of natural language is currently (and in the mid-term future) impossible
- However, the available language technologies can already provide more information than simple keyword-indexing
- Implementations are not mainstream yet, but this is changing as more open-source systems become available

The future

- Prediction: within 5 years, Text Mining will start to enter mainstream, similarly to the *Google* effect
- Requires more interdisciplinary work: computational linguists, information system engineers, domain experts

Some conclusions

The present

- A complete semantic understanding of natural language is currently (and in the mid-term future) impossible
- However, the available language technologies can already provide more information than simple keyword-indexing
- Implementations are not mainstream yet, but this is changing as more open-source systems become available

The future

- Prediction: within 5 years, Text Mining will start to enter mainstream, similarly to the *Google* effect
- Requires more interdisciplinary work: computational linguists, information system engineers, domain experts