Introduction to Natural Language Processing

Steven Bird Ewan Klein Edward Loper

University of Melbourne, AUSTRALIA

University of Edinburgh, UK

University of Pennsylvania, USA

August 27, 2008

Knowledge and Communication in Language

- human knowledge, human communication, expressed in language
- language technologies: process human language automatically
- handheld devices: predictive text, handwriting recognition
- web search engines: access to information locked up in text
- two facets of the multilingual information society:
  - natural human-machine interfaces
  - access to stored information

Knowledge and Communication in Language

- human knowledge, human communication, expressed in language
- language technologies: process human language automatically
- handheld devices: predictive text, handwriting recognition
- web search engines: access to information locked up in text
- two facets of the multilingual information society:
    - natural human-machine interfaces
    - access to stored information

Knowledge and Communication in Language

- human knowledge, human communication, expressed in language
- language technologies: process human language automatically
- handheld devices: predictive text, handwriting recognition
- web search engines: access to information locked up in text
- two facets of the multilingual information society:
  - natural human-machine interfaces
  - access to stored information

Knowledge and Communication in Language

- human knowledge, human communication, expressed in language
- language technologies: process human language automatically
- handheld devices: predictive text, handwriting recognition
- web search engines: access to information locked up in text
- two facets of the multilingual information society:
  - natural human-machine interfaces
  - access to stored information

Knowledge and Communication in Language

- human knowledge, human communication, expressed in language
- language technologies: process human language automatically
- handheld devices: predictive text, handwriting recognition
- web search engines: access to information locked up in text
- two facets of the multilingual information society:
  - natural human-machine interfaces
  - access to stored information

Knowledge and Communication in Language

- human knowledge, human communication, expressed in language
- language technologies: process human language automatically
- handheld devices: predictive text, handwriting recognition
- web search engines: access to information locked up in text
- two facets of the multilingual information society:
  - natural human-machine interfaces
  - access to stored information

Knowledge and Communication in Language

- human knowledge, human communication, expressed in language
- language technologies: process human language automatically
- handheld devices: predictive text, handwriting recognition
- web search engines: access to information locked up in text
- two facets of the multilingual information society:
  - natural human-machine interfaces
  - access to stored information

Problem

- awash with language data
- inadequate tools (will this ever change?)
- overheads: Perl, Prolog, Java
- Natural Language Toolkit (NLTK) as a solution

Problem

- awash with language data
- inadequate tools (will this ever change?)
- overheads: Perl, Prolog, Java
- Natural Language Toolkit (NLTK) as a solution

Problem

- awash with language data
- inadequate tools (will this ever change?)
- overheads: Perl, Prolog, Java
- Natural Language Toolkit (NLTK) as a solution

Problem

- awash with language data
- inadequate tools (will this ever change?)
- overheads: Perl, Prolog, Java
- Natural Language Toolkit (NLTK) as a solution

NLTK: What you get...

- Book
- Documentation
- FAQ
- Installation instructions for Python, NLTK, data
- Distributions: *Windows, Mac OSX, Unix, data, documentation*
- CD-ROM: *Python, NLTK, documentation, third-party libraries for numerical processing and visualization, instructions*
- Mailing lists: `nltk-announce`, `nltk-devel`, `nltk-users`, `nltk-portuguese`

NLTK: What you get...

- Book
- Documentation
- FAQ
- Installation instructions for Python, NLTK, data
- Distributions: *Windows, Mac OSX, Unix, data, documentation*
- CD-ROM: *Python, NLTK, documentation, third-party libraries for numerical processing and visualization, instructions*
- Mailing lists: `nltk-announce`, `nltk-devel`, `nltk-users`, `nltk-portuguese`

NLTK: What you get...

- Book
- Documentation
- FAQ
- Installation instructions for Python, NLTK, data
- Distributions: *Windows, Mac OSX, Unix, data, documentation*
- CD-ROM: *Python, NLTK, documentation, third-party libraries for numerical processing and visualization, instructions*
- Mailing lists: `nltk-announce`, `nltk-devel`, `nltk-users`, `nltk-portuguese`

NLTK: What you get...

- Book
- Documentation
- FAQ
- Installation instructions for Python, NLTK, data
- Distributions: *Windows, Mac OSX, Unix, data, documentation*
- CD-ROM: *Python, NLTK, documentation, third-party libraries for numerical processing and visualization, instructions*
- Mailing lists: `nltk-announce`, `nltk-devel`, `nltk-users`, `nltk-portuguese`

NLTK: What you get...

- Book
- Documentation
- FAQ
- Installation instructions for Python, NLTK, data
- Distributions: *Windows, Mac OSX, Unix, data, documentation*
- CD-ROM: *Python, NLTK, documentation, third-party libraries for numerical processing and visualization, instructions*
- Mailing lists: `nltk-announce`, `nltk-devel`, `nltk-users`, `nltk-portuguese`

NLTK: What you get...

- Book
- Documentation
- FAQ
- Installation instructions for Python, NLTK, data
- Distributions: *Windows, Mac OSX, Unix, data, documentation*
- CD-ROM: *Python, NLTK, documentation, third-party libraries for numerical processing and visualization, instructions*
- Mailing lists: `nltk-announce`, `nltk-devel`, `nltk-users`, `nltk-portuguese`

NLTK: What you get...

- Book
- Documentation
- FAQ
- Installation instructions for Python, NLTK, data
- Distributions: *Windows, Mac OSX, Unix, data, documentation*
- CD-ROM: *Python, NLTK, documentation, third-party libraries for numerical processing and visualization, instructions*
- Mailing lists: `nltk-announce`, `nltk-devel`, `nltk-users`, `nltk-portuguese`

NLTK: Who it is for...

- people who want to learn how to:
  - write programs
  - to analyze written language
- does not presume programming abilities:
  - working examples
  - graded exercises
- experienced programmers:
  - quickly learn Python (if necessary)
  - Python features for NLP
  - NLP algorithms and data structures

NLTK: Who it is for...

- people who want to learn how to:
  - write programs
  - to analyze written language
- does not presume programming abilities:
  - working examples
  - graded exercises
- experienced programmers:
  - quickly learn Python (if necessary)
  - Python features for NLP
  - NLP algorithms and data structures

NLTK: Who it is for...

- people who want to learn how to:
  - write programs
  - to analyze written language
- does not presume programming abilities:
  - working examples
  - graded exercises
- experienced programmers:
  - quickly learn Python (if necessary)
  - Python features for NLP
  - NLP algorithms and data structures

NLTK: Who it is for...

- people who want to learn how to:
  - write programs
  - to analyze written language
- does not presume programming abilities:
  - working examples
  - graded exercises
- experienced programmers:
  - quickly learn Python (if necessary)
  - Python features for NLP
  - NLP algorithms and data structures

NLTK: Who it is for...

- people who want to learn how to:
  - write programs
  - to analyze written language
- does not presume programming abilities:
  - working examples
  - graded exercises
- experienced programmers:
  - quickly learn Python (if necessary)
  - Python features for NLP
  - NLP algorithms and data structures

NLTK: Who it is for...

- people who want to learn how to:
    - write programs
    - to analyze written language
- does not presume programming abilities:
    - working examples
    - graded exercises
- experienced programmers:
    - quickly learn Python (if necessary)
    - Python features for NLP
    - NLP algorithms and data structures

NLTK: Who it is for...

- people who want to learn how to:
  - write programs
  - to analyze written language
- does not presume programming abilities:
  - working examples
  - graded exercises
- experienced programmers:
  - quickly learn Python (if necessary)
  - Python features for NLP
  - NLP algorithms and data structures

NLTK: Who it is for...

- people who want to learn how to:
  - write programs
  - to analyze written language
- does not presume programming abilities:
  - working examples
  - graded exercises
- experienced programmers:
  - quickly learn Python (if necessary)
  - Python features for NLP
  - NLP algorithms and data structures

NLTK: Who it is for...

- people who want to learn how to:
  - write programs
  - to analyze written language
- does not presume programming abilities:
  - working examples
  - graded exercises
- experienced programmers:
  - quickly learn Python (if necessary)
  - Python features for NLP
  - NLP algorithms and data structures

NLTK: Who it is for...

- people who want to learn how to:
  - write programs
  - to analyze written language
- does not presume programming abilities:
  - working examples
  - graded exercises
- experienced programmers:
  - quickly learn Python (if necessary)
  - Python features for NLP
  - NLP algorithms and data structures

NLTK: What you will learn...

1. how to analyze language data
2. key concepts from linguistic description and analysis
3. how linguistic knowledge is used in NLP components
4. data structures and algorithms used in NLP and linguistic data management
5. standard corpora and their use in formal evaluation
6. organization of the field of NLP
7. skills in Python programming for NLP

NLTK: What you will learn...

1. how to analyze language data
2. key concepts from linguistic description and analysis
3. how linguistic knowledge is used in NLP components
4. data structures and algorithms used in NLP and linguistic data management
5. standard corpora and their use in formal evaluation
6. organization of the field of NLP
7. skills in Python programming for NLP

NLTK: What you will learn...

1. how to analyze language data
2. key concepts from linguistic description and analysis
3. how linguistic knowledge is used in NLP components
4. data structures and algorithms used in NLP and linguistic data management
5. standard corpora and their use in formal evaluation
6. organization of the field of NLP
7. skills in Python programming for NLP

NLTK: What you will learn...

1. how to analyze language data
2. key concepts from linguistic description and analysis
3. how linguistic knowledge is used in NLP components
4. data structures and algorithms used in NLP and linguistic data management
5. standard corpora and their use in formal evaluation
6. organization of the field of NLP
7. skills in Python programming for NLP

NLTK: What you will learn...

1. how to analyze language data
2. key concepts from linguistic description and analysis
3. how linguistic knowledge is used in NLP components
4. data structures and algorithms used in NLP and linguistic data management
5. standard corpora and their use in formal evaluation
6. organization of the field of NLP
7. skills in Python programming for NLP

NLTK: What you will learn...

1. how to analyze language data
2. key concepts from linguistic description and analysis
3. how linguistic knowledge is used in NLP components
4. data structures and algorithms used in NLP and linguistic data management
5. standard corpora and their use in formal evaluation
6. organization of the field of NLP
7. skills in Python programming for NLP

NLTK: What you will learn...

1. how to analyze language data
2. key concepts from linguistic description and analysis
3. how linguistic knowledge is used in NLP components
4. data structures and algorithms used in NLP and linguistic data management
5. standard corpora and their use in formal evaluation
6. organization of the field of NLP
7. skills in Python programming for NLP

NLTK: Your likely goals...

Goals	**Background**	
	Arts and Humanities	*Science and Engineering*
Language Analysis	Programming to manage language data, explore linguistic models, and test empirical claims	Language as a source of interesting problems in data modeling, data mining, and knowledge discovery
Language Technology	Learning to program, with applications to familiar problems, to work in language technology or other technical field	Knowledge of linguistic algorithms and data structures for high quality, maintainable language processing software

Philosophy

- practical
- programming
- principled
- pragmatic
- pleasurable
- portal

Philosophy

- practical
- programming
- principled
- pragmatic
- pleasurable
- portal

Philosophy

- practical
- programming
- principled
- pragmatic
- pleasurable
- portal

Philosophy

- practical
- programming
- principled
- pragmatic
- pleasurable
- portal

Philosophy

- practical
- programming
- principled
- pragmatic
- pleasurable
- portal

Philosophy

- practical
- programming
- principled
- pragmatic
- pleasurable
- portal

Structure

- Three parts:
  1. **Basics:** text processing, tokenization, tagging, lexicons, language engineering, text classification
  2. **Parsing:** phrase structure, trees, grammars, chunking, parsing
  3. **Advanced Topics:** selected topics in greater depth: feature-based grammar, unification, semantics, linguistic data management
- each part: chapter on programming; three chapters on NLP
- each chapter: motivation, sections, graded exercises, summary, further reading

Structure

- Three parts:
  1. **Basics:** text processing, tokenization, tagging, lexicons, language engineering, text classification
  2. **Parsing:** phrase structure, trees, grammars, chunking, parsing
  3. **Advanced Topics:** selected topics in greater depth: feature-based grammar, unification, semantics, linguistic data management
- each part: chapter on programming; three chapters on NLP
- each chapter: motivation, sections, graded exercises, summary, further reading

Structure

- Three parts:
  1. **Basics:** text processing, tokenization, tagging, lexicons, language engineering, text classification
  2. **Parsing:** phrase structure, trees, grammars, chunking, parsing
  3. **Advanced Topics:** selected topics in greater depth: feature-based grammar, unification, semantics, linguistic data management
- each part: chapter on programming; three chapters on NLP
- each chapter: motivation, sections, graded exercises, summary, further reading

Structure

- Three parts:
  1. **Basics:** text processing, tokenization, tagging, lexicons, language engineering, text classification
  2. **Parsing:** phrase structure, trees, grammars, chunking, parsing
  3. **Advanced Topics:** selected topics in greater depth: feature-based grammar, unification, semantics, linguistic data management
- each part: chapter on programming; three chapters on NLP
- each chapter: motivation, sections, graded exercises, summary, further reading

Structure

- Three parts:
  1. **Basics:** text processing, tokenization, tagging, lexicons, language engineering, text classification
  2. **Parsing:** phrase structure, trees, grammars, chunking, parsing
  3. **Advanced Topics:** selected topics in greater depth: feature-based grammar, unification, semantics, linguistic data management
- each part: chapter on programming; three chapters on NLP
- each chapter: motivation, sections, graded exercises, summary, further reading

Structure

- Three parts:
  1. **Basics:** text processing, tokenization, tagging, lexicons, language engineering, text classification
  2. **Parsing:** phrase structure, trees, grammars, chunking, parsing
  3. **Advanced Topics:** selected topics in greater depth: feature-based grammar, unification, semantics, linguistic data management
- each part: chapter on programming; three chapters on NLP
- each chapter: motivation, sections, graded exercises, summary, further reading

Python: Key Features

- simple yet powerful, shallow learning curve
- object-oriented: encapsulation, re-use
- scripting language, facilitates interactive exploration
- excellent functionality for processing linguistic data
- extensive standard library, incl graphics, web, numerical processing
- downloaded for free from `http://www.python.org/`

Python: Key Features

- simple yet powerful, shallow learning curve
- object-oriented: encapsulation, re-use
- scripting language, facilitates interactive exploration
- excellent functionality for processing linguistic data
- extensive standard library, incl graphics, web, numerical processing
- downloaded for free from `http://www.python.org/`

Python: Key Features

- simple yet powerful, shallow learning curve
- object-oriented: encapsulation, re-use
- scripting language, facilitates interactive exploration
- excellent functionality for processing linguistic data
- extensive standard library, incl graphics, web, numerical processing
- downloaded for free from `http://www.python.org/`

Python: Key Features

- simple yet powerful, shallow learning curve
- object-oriented: encapsulation, re-use
- scripting language, facilitates interactive exploration
- excellent functionality for processing linguistic data
- extensive standard library, incl graphics, web, numerical processing
- downloaded for free from `http://www.python.org/`

Python: Key Features

- simple yet powerful, shallow learning curve
- object-oriented: encapsulation, re-use
- scripting language, facilitates interactive exploration
- excellent functionality for processing linguistic data
- extensive standard library, incl graphics, web, numerical processing
- downloaded for free from `http://www.python.org/`

Python: Key Features

- simple yet powerful, shallow learning curve
- object-oriented: encapsulation, re-use
- scripting language, facilitates interactive exploration
- excellent functionality for processing linguistic data
- extensive standard library, incl graphics, web, numerical processing
- downloaded for free from `http://www.python.org/`

Python Example

```
import sys
for line in sys.stdin.readlines():
    for word in line.split():
        if word.endswith(’ing’):
            print word
```

1. whitespace: nesting lines of code; scope
2. object-oriented: attributes, methods (e.g. `line`)
3. readable

Comparison with Perl

```
while (<>) {
    foreach my $word (split) {
        if ($word =~ /ing$/) {
            print "$word\n";
        }
    }
}
```

1. syntax is obscure: *what are:* `<> $ my split` ?
2. "it is quite easy in Perl to write programs that simply look like raving gibberish, even to experienced Perl programmers" (Hammond *Perl Programming for Linguists* 2003:47)
3. large programs difficult to maintain, reuse

What NLTK adds to Python

NLTK defines a basic infrastructure that can be used to build NLP programs in Python. It provides:

- Basic classes for representing data relevant to natural language processing
- Standard interfaces for performing tasks, such as tokenization, tagging, and parsing
- Standard implementations for each task, which can be combined to solve complex problems
- Demonstrations (parsers, chunkers, chatbots)
- Extensive documentation, including tutorials and reference documentation

What NLTK adds to Python

NLTK defines a basic infrastructure that can be used to build NLP programs in Python. It provides:

- Basic classes for representing data relevant to natural language processing
- Standard interfaces for performing tasks, such as tokenization, tagging, and parsing
- Standard implementations for each task, which can be combined to solve complex problems
- Demonstrations (parsers, chunkers, chatbots)
- Extensive documentation, including tutorials and reference documentation

What NLTK adds to Python

NLTK defines a basic infrastructure that can be used to build NLP programs in Python. It provides:

- Basic classes for representing data relevant to natural language processing
- Standard interfaces for performing tasks, such as tokenization, tagging, and parsing
- Standard implementations for each task, which can be combined to solve complex problems
- Demonstrations (parsers, chunkers, chatbots)
- Extensive documentation, including tutorials and reference documentation

What NLTK adds to Python

NLTK defines a basic infrastructure that can be used to build NLP programs in Python. It provides:

- Basic classes for representing data relevant to natural language processing
- Standard interfaces for performing tasks, such as tokenization, tagging, and parsing
- Standard implementations for each task, which can be combined to solve complex problems
- Demonstrations (parsers, chunkers, chatbots)
- Extensive documentation, including tutorials and reference documentation

What NLTK adds to Python

NLTK defines a basic infrastructure that can be used to build NLP programs in Python. It provides:

- Basic classes for representing data relevant to natural language processing
- Standard interfaces for performing tasks, such as tokenization, tagging, and parsing
- Standard implementations for each task, which can be combined to solve complex problems
- Demonstrations (parsers, chunkers, chatbots)
- Extensive documentation, including tutorials and reference documentation

NLTK Design: Requirements

1. **simplicity:** intuitive framework with substantial building blocks
2. **consistency:** uniform data structures, interfaces — predictability
3. **extensibility:** accommodates new components (replicate vs extend exiting functionality)
4. **modularity:** interaction between components
5. **well-documented:** substantial documentation

NLTK Design: Requirements

1. **simplicity:** intuitive framework with substantial building blocks
2. **consistency:** uniform data structures, interfaces — predictability
3. **extensibility:** accommodates new components (replicate vs extend exiting functionality)
4. **modularity:** interaction between components
5. **well-documented:** substantial documentation

NLTK Design: Requirements

1. **simplicity:** intuitive framework with substantial building blocks
2. **consistency:** uniform data structures, interfaces — predictability
3. **extensibility:** accommodates new components (replicate vs extend exiting functionality)
4. **modularity:** interaction between components
5. **well-documented:** substantial documentation

NLTK Design: Requirements

1. **simplicity:** intuitive framework with substantial building blocks
2. **consistency:** uniform data structures, interfaces — predictability
3. **extensibility:** accommodates new components (replicate vs extend exiting functionality)
4. **modularity:** interaction between components
5. **well-documented:** substantial documentation

NLTK Design: Requirements

1. **simplicity:** intuitive framework with substantial building blocks
2. **consistency:** uniform data structures, interfaces — predictability
3. **extensibility:** accommodates new components (replicate vs extend exiting functionality)
4. **modularity:** interaction between components
5. **well-documented:** substantial documentation

NLTK Design: Non-requirements

1. **encyclopedic:** has many gaps; opportunity for students to extend it
2. **efficiency:** not highly optimised for runtime performance
3. **programming tricks:** avoid in preference for clear implementations (replicate vs extend exiting functionality)

NLTK Design: Non-requirements

1. **encyclopedic:** has many gaps; opportunity for students to extend it
2. **efficiency:** not highly optimised for runtime performance
3. **programming tricks:** avoid in preference for clear implementations (replicate vs extend exiting functionality)

NLTK Design: Non-requirements

1. **encyclopedic:** has many gaps; opportunity for students to extend it
2. **efficiency:** not highly optimised for runtime performance
3. **programming tricks:** avoid in preference for clear implementations (replicate vs extend exiting functionality)

Corpora Distributed with NLTK

- Australian ABC News, 2 genres, 660k words, sentence-segmented
- Brown Corpus, 15 genres, 1.15M words, tagged
- CMU Pronouncing Dictionary, 127k entries
- CoNLL 2000 Chunking Data, 270k words, tagged and chunked
- CoNLL 2002 Named Entity, 700k words, pos- and named-entity-tagged (Dutch, Spanish)
- Floresta Treebank, 9k sentences (Portuguese)
- Genesis Corpus, 6 texts, 200k words, 6 languages
- Gutenberg (sel), 14 texts, 1.7M words
- Indian POS-Tagged Corpus, 60k words pos-tagged (Bangla, Hindi, Marathi, Telugu)
- NIST 1999 Info Extr (sel), 63k words, newswire and named-entity SGML markup
- Names Corpus, 8k male and female names
- PP Attachment Corpus, 28k prepositional phrases, tagged as noun or verb modifiers
- Presidential Addresses, 485k words, formatted text
- Roget's Thesaurus, 200k words, formatted text
- SEMCOR, 880k words, part-of-speech and sense tagged
- SENSEVAL 2, 600k words, part-of-speech and sense tagged
- Shakespeare XML Corpus (sel), 8 books
- Stopwords Corpus, 2,400 stopwords for 11 languages
- Switchboard Corpus (sel), 36 phonecalls, transcribed, parsed
- Univ Decl Human Rights, 480k words, 300+ languages
- US Pres Addr Corpus, 480k words
- Penn Treebank (sel), 40k words, tagged and parsed
- TIMIT Corpus (sel), audio files and transcripts for 16 speakers
- Wordlist Corpus, 960k words and 20k affixes for 8 languages
- WordNet, 145k synonym sets