Chapter NLP:II

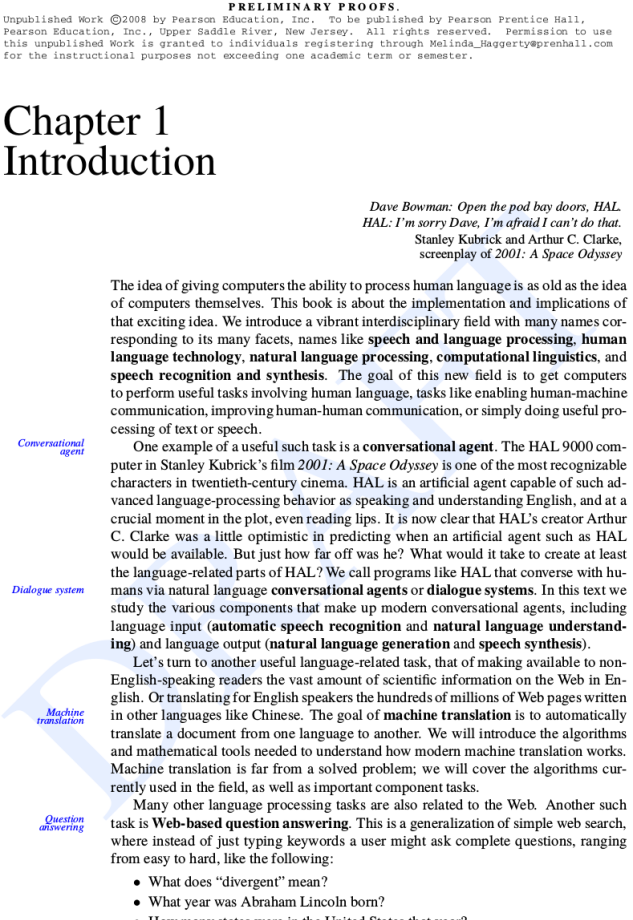
II. Text Models

- Text Structure
- Text Preprocessing I
- Text Preprocessing II
- Text Representation
- Text Similarity
- Sequence Modeling
- Language Modeling

Text Preprocessing

Overview

The goal of text preprocessing is its conversion into a canonical form.



PRELIMINARY PROOFS.

Chapter 1
Introduction

Dave Bowman: Open the pod bay doors, HAL.
HAL: I'm sorry Dave, I'm afraid I can't do that.
Stanley Kubrick and Arthur C. Clarke,
screenplay of *2001: A Space Odyssey*

The idea of giving computers the ability to process human language is as old as the idea of computers themselves. This book is about the implementation and implications of that exciting idea. We introduce a vibrant interdisciplinary field with many names corresponding to its many facets, names like **speech and language processing**, **human language technology**, **natural language processing**, **computational linguistics**, and **speech recognition and synthesis**. The goal of this new field is to get computers to perform useful tasks involving human language, tasks like enabling human-machine communication, improving human-human communication, or simply doing useful processing of text or speech.

Conversational agent

One example of a useful such task is a **conversational agent**. The HAL 9000 computer in Stanley Kubrick's film *2001: A Space Odyssey* is one of the most recognizable characters in twentieth-century cinema. HAL is an artificial agent capable of such advanced language-processing behavior as speaking and understanding English, and at a crucial moment in the plot, even reading lips. It is now clear that HAL's creator Arthur C. Clarke was a little optimistic in predicting when an artificial agent such as HAL would be available. But just how far off was he? What would it take to create at least the language-related parts of HAL? We call programs like HAL that converse with humans via natural language **conversational agents** or **dialogue systems**. In this text we study the various components that make up modern conversational agents, including language input (**automatic speech recognition** and **natural language understanding**) and language output (**natural language generation** and **speech synthesis**).

Dialogue system

Machine translation

Let's turn to another useful language-related task, that of making available to non-English-speaking readers the vast amount of scientific information on the Web in English. Or translating for English speakers the hundreds of millions of Web pages written in other languages like Chinese. The goal of **machine translation** is to automatically translate a document from one language to another. We will introduce the algorithms and mathematical tools needed to understand how modern machine translation works. Machine translation is far from a solved problem; we will cover the algorithms currently used in the field, as well as important component tasks.

Question answering

Many other language processing tasks are also related to the Web. Another such task is **Web-based question answering**. This is a generalization of simple web search, where instead of just typing keywords a user might ask complete questions, ranging from easy to hard, like the following:

- What does "divergent" mean?
- What year was Abraham Lincoln born?
- How many states were in the United States that year?

screenshot-jurafsky08-speech-and-language-processing-pdftotext-output.txt

Open Save

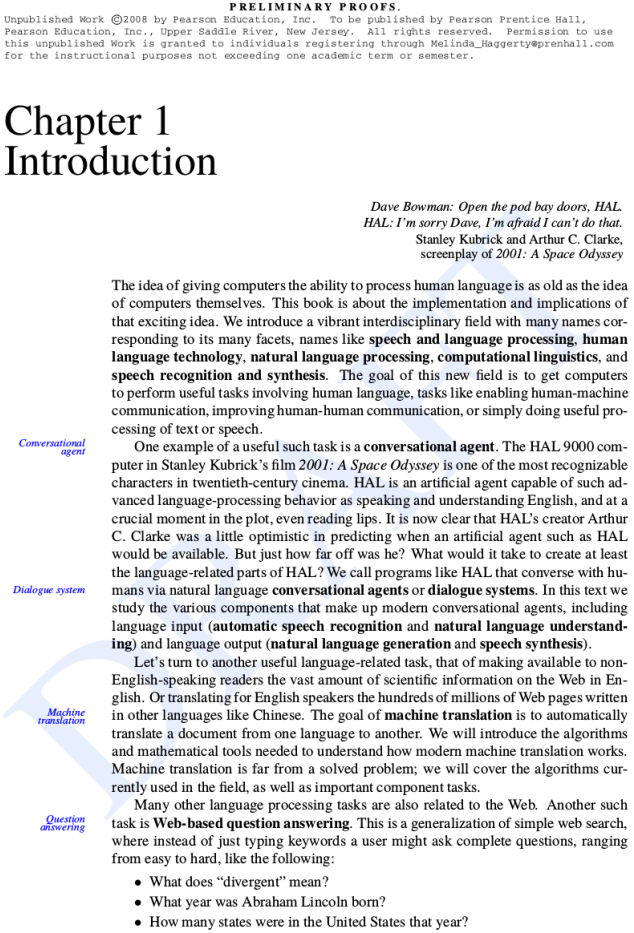
```
PRELIMINARY PROOFS.
c
Unpublished Work ©2008
by Pearson Education, Inc. To be published by Pearson Prentice Hall,
Pearson Education, Inc., Upper Saddle River, New Jersey. All rights reserved. Permission to use
this unpublished Work is granted to individuals registering through Melinda_Haggerty@prenhall.com
for the instructional purposes not exceeding one academic term or semester.

Chapter 1
Introduction
Dave Bowman: Open the pod bay doors, HAL.
HAL: I'm sorry Dave, I'm afraid I can't do that.
Stanley Kubrick and Arthur C. Clarke,
screenplay of 2001: A Space Odyssey

FT

D
RA

Conversational
agent

The idea of giving computers the ability to process human language is as old as the idea
of computers themselves. This book is about the implementation and implications of
that exciting idea. We introduce a vibrant interdisciplinary field with many names corresponding to its
  many facets, names like speech and language processing, human
language technology, natural language processing, computational linguistics, and
speech recognition and synthesis. The goal of this new field is to get computers
to perform useful tasks involving human language, tasks like enabling human-machine
communication, improving human-human communication, or simply doing useful processing of text or speech.
One example of a useful such task is a conversational agent. The HAL 9000 computer in Stanley Kubrick's
  film 2001: A Space Odyssey is one of the most recognizable
characters in twentieth-century cinema. HAL is an artificial agent capable of such advanced language-
  processing behavior as speaking and understanding English, and at a
crucial moment in the plot, even reading lips. It is now clear that HAL's creator Arthur
C. Clarke was a little optimistic in predicting when an artificial agent such as HAL
would be available. But just how far off was he? What would it take to create at least
the language-related parts of HAL? We call programs like HAL that converse with humans via natural
  language conversational agents or dialogue systems. In this text we
study the various components that make up modern conversational agents, including
language input (automatic speech recognition and natural language understanding) and language output
  (natural language generation and speech synthesis).
Let's turn to another useful language-related task, that of making available to nonEnglish-speaking
  readers the vast amount of scientific information on the Web in English. Or translating for English
  speakers the hundreds of millions of Web pages written
in other languages like Chinese. The goal of machine translation is to automatically
translate a document from one language to another. We will introduce the algorithms
and mathematical tools needed to understand how modern machine translation works.
Machine translation is far from a solved problem; we will cover the algorithms currently used in the
  field, as well as important component tasks.
Many other language processing tasks are also related to the Web. Another such
task is Web-based question answering. This is a generalization of simple web search,
where instead of just typing keywords a user might ask complete questions, ranging
from easy to hard, like the following:

Dialogue system

Machine
translation

Question
answering

• What does "divergent" mean?
• What year was Abraham Lincoln born?
• How many states were in the United States that year?

```

Plain Text Tab Width: 2 Ln 1, Col 35 INS

Text Preprocessing

Overview

The goal of text preprocessing is its conversion into a canonical form.



PRELIMINARY PROOFS.

Chapter 1
Introduction

Dave Bowman: Open the pod bay doors, HAL.
HAL: I'm sorry Dave, I'm afraid I can't do that.
Stanley Kubrick and Arthur C. Clarke,
screenplay of *2001: A Space Odyssey*

The idea of giving computers the ability to process human language is as old as the idea of computers themselves. This book is about the implementation and implications of that exciting idea. We introduce a vibrant interdisciplinary field with many names corresponding to its many facets, names like **speech and language processing**, **human language technology**, **natural language processing**, **computational linguistics**, and **speech recognition and synthesis**. The goal of this new field is to get computers to perform useful tasks involving human language, tasks like enabling human-machine communication, improving human-human communication, or simply doing useful processing of text or speech.

Conversational agent

One example of a useful such task is a **conversational agent**. The HAL 9000 computer in Stanley Kubrick's film *2001: A Space Odyssey* is one of the most recognizable characters in twentieth-century cinema. HAL is an artificial agent capable of such advanced language-processing behavior as speaking and understanding English, and at a crucial moment in the plot, even reading lips. It is now clear that HAL's creator Arthur C. Clarke was a little optimistic in predicting when an artificial agent such as HAL would be available. But just how far off was he? What would it take to create at least the language-related parts of HAL? We call programs like HAL that converse with humans via natural language **conversational agents** or **dialogue systems**. In this text we study the various components that make up modern conversational agents, including language input (**automatic speech recognition** and **natural language understanding**) and language output (**natural language generation** and **speech synthesis**).

Dialogue system

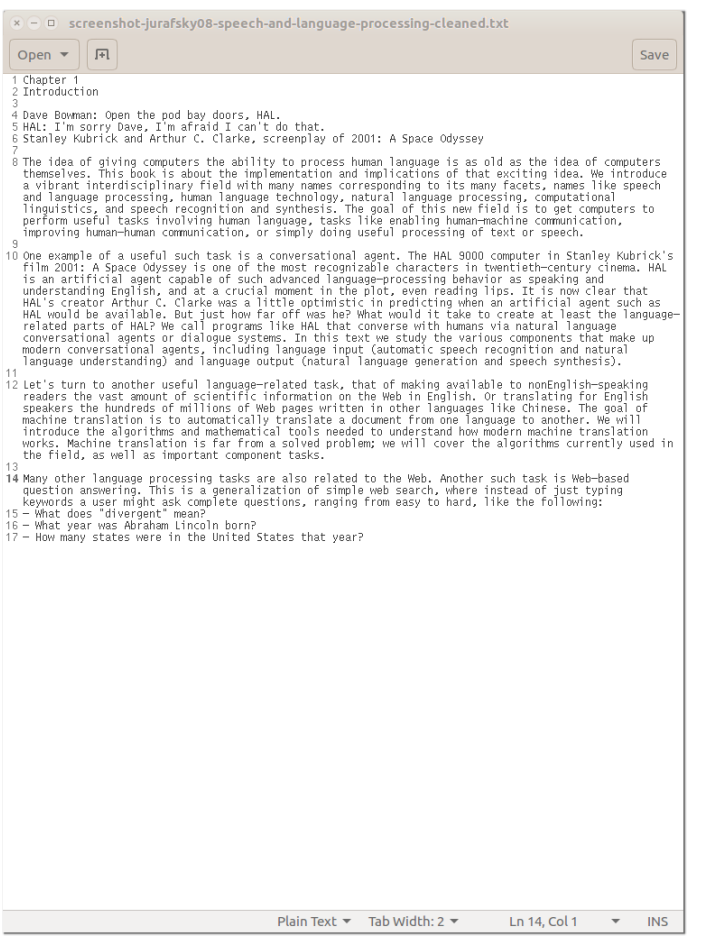
Machine translation

Let's turn to another useful language-related task, that of making available to non-English-speaking readers the vast amount of scientific information on the Web in English. Or translating for English speakers the hundreds of millions of Web pages written in other languages like Chinese. The goal of **machine translation** is to automatically translate a document from one language to another. We will introduce the algorithms and mathematical tools needed to understand how modern machine translation works. Machine translation is far from a solved problem; we will cover the algorithms currently used in the field, as well as important component tasks.

Question answering

Many other language processing tasks are also related to the Web. Another such task is **Web-based question answering**. This is a generalization of simple web search, where instead of just typing keywords a user might ask complete questions, ranging from easy to hard, like the following:

- What does "divergent" mean?
- What year was Abraham Lincoln born?
- How many states were in the United States that year?



screenshot-jurafsky08-speech-and-language-processing-cleaned.txt

Open Save

```
1 Chapter 1
2 Introduction
3
4 Dave Bowman: Open the pod bay doors, HAL.
5 HAL: I'm sorry Dave, I'm afraid I can't do that.
6 Stanley Kubrick and Arthur C. Clarke, screenplay of 2001: A Space Odyssey
7
8 The idea of giving computers the ability to process human language is as old as the idea of computers
  themselves. This book is about the implementation and implications of that exciting idea. We introduce
  a vibrant interdisciplinary field with many names corresponding to its many facets, names like speech
  and language processing, human language technology, natural language processing, computational
  linguistics, and speech recognition and synthesis. The goal of this new field is to get computers to
  perform useful tasks involving human language, tasks like enabling human-machine communication,
  improving human-human communication, or simply doing useful processing of text or speech.
9
10 One example of a useful such task is a conversational agent. The HAL 9000 computer in Stanley Kubrick's
   film 2001: A Space Odyssey is one of the most recognizable characters in twentieth-century cinema. HAL
   is an artificial agent capable of such advanced language-processing behavior as speaking and
   understanding English, and at a crucial moment in the plot, even reading lips. It is now clear that
   HAL's creator Arthur C. Clarke was a little optimistic in predicting when an artificial agent such as
   HAL would be available. But just how far off was he? What would it take to create at least the language-
   related parts of HAL? We call programs like HAL that converse with humans via natural language
   conversational agents or dialogue systems. In this text we study the various components that make up
   modern conversational agents, including language input (automatic speech recognition and natural
   language understanding) and language output (natural language generation and speech synthesis).
11
12 Let's turn to another useful language-related task, that of making available to nonEnglish-speaking
   readers the vast amount of scientific information on the Web in English. Or translating for English
   speakers the hundreds of millions of Web pages written in other languages like Chinese. The goal of
   machine translation is to automatically translate a document from one language to another. We will
   introduce the algorithms and mathematical tools needed to understand how modern machine translation
   works. Machine translation is far from a solved problem; we will cover the algorithms currently used in
   the field, as well as important component tasks.
13
14 Many other language processing tasks are also related to the Web. Another such task is Web-based
   question answering. This is a generalization of simple web search, where instead of just typing
   keywords a user might ask complete questions, ranging from easy to hard, like the following:
15 - What does "divergent" mean?
16 - What year was Abraham Lincoln born?
17 - How many states were in the United States that year?
```

Plain Text Tab Width: 2 Ln 14, Col 1 INS

Text Preprocessing

Overview

Rationale:

- Ease implementation of subsequent processing steps
  A unified input format simplifies implementing processing steps. Example: In web search engines, all documents are converted to HTML. All indexing steps can expect HTML as input.
- Avoid processing errors and model bias
  Many rule-based and learning-based processing steps in an NLP pipeline may fail or be misled because of random text artifacts. High-level processing steps presume clean text. Examples: Text classifications may learn to exploit PDF-to-text conversion artifacts rather than a text's contents; parsers require grammatical text.

Constraints:

- Task-dependence
  The canonical form depends on the task at hand and its requirements.
- Provenance
  The possibility to determine from where in a raw text corpus, a preprocessed text originated.
- Reversibility
  The capability to render a preprocessed text in human-readable form.

Text Preprocessing

Overview

Common preprocessing steps:

- Conversion to plain text
- Encoding detection and unification
- Line break unification (\n – UNIX, \r\n – Windows)
- Extraction of main content and meta information
- Normalization and/or paraphrasing
- Annotation

Plain text formats:

- unformatted, raw
- formatted and/or markup
- inline or external annotations

Text Preprocessing

Overview

Normalization and/or paraphrasing:

- Faithful to the original text
- Departure from the original text
  - Unification across texts
  - Canonicalization
    Whitespace, spelling, grammar; translation of text messages to common text norms
  - Expansion and/or abstraction
    Abbreviations, anaphora, translation to spoken language, canonicalization of tokens

Annotation:

- Syntactic units: phonemes, morphemes, tokens (esp. words), sentences
- Discourse units: paragraphs, sections, chapters
- Typographic units: lines, pages (layout, meta-information), documents
- Meta-information: title, authors, date, properties, ...

Text Preprocessing

Tokenization

Tokenization turns a sequence of characters into a sequence of tokens.

Example:

```
Friends, Romans, Countrymen, lend me your ears !
```

`Friends` `,` `Romans` `,` `Countrymen` `,` `lend` `me` `your` `ears` `!`

Terminology: (simplified)

- A token is a character sequence forming a useful semantic unit.
- A type is to a token what a class is to an object.

Token-granularity:

- Word-level: may or may not inlcude whitespace between words
- Phrase-level: identification of multi-term named entities and common phrases
- Sentence-level: one token corresponds to one clause, or one sentence

Remarks:

- A related philosophical concept is the type-token distinction (see unit about corpus linguistics in this course). Here, a token is a specific instance of a word (i.e., its specific written form), and a type refers to its underlying concept as a whole. This is comparable to the distinction between class and object in object-oriented computer programming. For example, the sentence “A rose is a rose is a rose.” comprises nine token instances but only four types, namely “a”, “rose”, “is”, and “.”. [Wikipedia]
- Tokenization is strongly language-dependent. English is already among the easiest languages to be tokenized, and there are still many problems to be solved. In Chinese, for example, words are not separated by a specific character, rendering the process of determining word boundaries much more difficult.

Text Preprocessing

Tokenization: Special Cases

- **Contractions**
  Apostrophes can be a part of a word, a part of a possessive, or just a mistake: `it's`, `o'donnell`, `can't`, `don't`, `80's`, `men's`, `master's degree`, `shriner's`
- **Hyphenated compounds**
  Hyphens may be part of a word, a separator, and some words refer to the same concept with or without hyphen: `winston-salem`, `e-bay`, `wal-mart`, `active-x`, `far-reaching`, `loud-mouthed`, `20-year-old`.
- **Compounds**
  English: `wheelchair`, German: `Computerlinguistik` for computational linguistics.
- **Other special characters**
  Special characters may form part of words, especially in technology-related text: `M*A*S*H`, `I.B.M.`, `Ph.D.`, `C++`, `C#`, ` `, `http://www.example.com`.
- **Numbers**
  Numbers form tokens of their own, and may contain punctuation as well: `6.5`, `1e+010`.
- **Phrase tokens: named entities, phone numbers, dates**
  `San Francisco`, `(800) 234-2333`, `Mar 11,1983`.

Text Preprocessing

Tokenization Approaches

- Heuristics
  - Whitespace: A token is every character sequence separated by whitespace characters.
  - TREC: A token is every alphanumeric sequence of characters of length > 3, separated by a space or punctuation mark.
- Rule-based

  Applications of rules to a text so that tokens are separated by whitespace from each other. This allows subsequent processing steps to apply the whitespace heuristic
- Machine learning-based

  Based on sufficiently large training data of correctly tokenized text, a model can be trained to decide at every character position whether to split tokens.

Text Preprocessing

Rule-based Tokenization [Jurasky and Martin, 2007] [Grefenstette, 1999]

Algorithm: Tokenization with Regular Expressions.

Input: d. Document in the form of a string.
A. Dictionary of abbreviations.

Output: The document with space in-between its tokens.

Tokenize(d, A)

1. `alnum = [A-Za-z0-9]; nalnum = [^A-Za-z0-9]; alwayssep = [?!()";/\|']`
2. `clitic = ('|:|-|'S|'D|'M|'LL|'RE|'VE|N'T|'s|'d|'m|'ll|'re|'ve|n't)`
3. `// Put whitespace around unambiguous separators.`
4. `// Put whitespace around commas that aren't inside numbers.`
5. `// Segment single quotes not preceded by letter (not apostrophes).`
6. `// Segment unambiguous word-final clitics and punctuation.`
7. Split d by whitespace (**/\s+/**) to obtain a list of tokens T.
8. `// Segment periods from each` $t \in T$ `that isn't an abbreviation in` A `or like one (letter period sequence or letter followed by consonants).`
9. `// Optionally expand clitics to normalize them.`
10. `Return a whitespace-separated string of` T.

Text Preprocessing

Rule-based Tokenization [Jurasky and Martin, 2007] [Grefenstette, 1999]

Algorithm: Tokenization with Regular Expressions.

Input: d. Document in the form of a string.
A. Dictionary of abbreviations.

Output: The document with space in-between its tokens.

Tokenize(d, A)

1. `alnum = [A-Za-z0-9]; nalnum = [^A-Za-z0-9]; alwayssep = [?!()";/\|']`
2. `clitic = ('|:|-|'S|'D|'M|'LL|'RE|'VE|N'T|'s|'d|'m|'ll|'re|'ve|n't)`
3. Apply **`s/$alwayssep/ $& /g`** to d.
4. Apply **`s/([^0-9]),/$1 , /g`** and **`s/,([^0-9])/ , $1/g`** to d.
5. Apply **`s/^'/$& /g`** and **`s/($nalnum)'/$1 '/g`** to d.
6. Apply **`s/$clitic$/ $&/g`** and **`s/$clitic($nalnum)/ $1 $2/g`** to d.
7. Split d by whitespace (**`/\s+/`**) to obtain a list of tokens T.
8. Apply **`s/\.$/ \./`** to $t \in T$ if t matches **`/$alnum\./`** and is not in A and doesn't match **`/^([A-Za-z]\.([A-Za-z]\.)+|[A-Z][bcdfghj-np-tvxz]+\.)$/`**.
9. Optionally expand clitics: **`s/'ve/have/`** and **`s/'m/am/`** and so on.
10. Return a whitespace-separated string of T.

Text Preprocessing

Stopping (Token Removal)

Stopping refers to the removal of tokens from a token sequence that are not useful in order to reduce data and improve performance of subsequent tasks:

- Frequent tokens (collection-specific)
  Example: `Wikipedia` when processing Wikipedia.
- Function word tokens (language-dependent)
  `the`, `of`, `and`, etc; strong overlap with frequent tokens.
  Problem: `to be or not to be` would be completely lost.
- Punctuation-only tokens
  Counter-example: `;-)`
- Number-only tokens
- Short tokens
  Short words may be important. Examples: `xp`, `ma`, `pm`, `ben e king`, `el paso`, `master p`, `gm`, `j lo`, `world war II`.

Stop word lists are typically customized to the text domain.

The top 100 most common words account for up to 50% of all words. [Wikipedia]

Text Preprocessing

Stopping (Token Removal) (continued)

Example:

```
The idea of giving computers the ability to process human language is as
old as the idea of computers themselves.  This book is about the
implementation and implications of that exciting idea.  We introduce a
vibrant interdisciplinary field with many names corresponding to its many
facets, names like speech and language processing, human language
technology, natural language processing, computational linguistics, and
speech recognition and synthesis.  The goal of this new field is to get
computers to perform useful tasks involving human language, tasks like
enabling human-machine communication, improving human-human
communication, or simply doing useful processing of text or speech.
```

Text Preprocessing

Stopping (Token Removal) (continued)

Example:

The idea of giving computers the ability to process human language is as old as the idea of computers themselves. This book is about the implementation and implications of that exciting idea. We introduce a vibrant interdisciplinary field with many names corresponding to its many facets, names like speech and language processing, human language technology, natural language processing, computational linguistics, and speech recognition and synthesis. The goal of this new field is to get computers to perform useful tasks involving human language, tasks like enabling human–machine communication, improving human–human communication, or simply doing useful processing of text or speech.

Text Preprocessing

Token Normalization

Application of heuristic rules to each token in an attempt to unify them.

- Lower-casing
  Problem: Capitalization may carry distinctions between word semantics.
  Examples: `Bush` vs. `bush`, `Apple` vs. `apple`.
- Removal of special characters
  Example: `U.S.A.` → `USA`
- Removal of diacritical marks
  Example: `café` → `cafe`
- Spelling correction
  Example: `My gramma got die of beaties` → `My grandma got diabetes`
- Reduction of morphology
  Lemmatization or stemming heuristics