A very short introduction to Language Technologies and Natural Language Processing

Jose F Quesada & Jose Luis Pro

Language

Set of conventional spoken or written symbols used for commucation between entities.

So we can see a language as the linking between meaning (semantic side) and expression (syntantic side).

Types of languages

- **Natural languages:** Used for the communication between human beings.
- **Formal languages:** Used by computers and in mathematical areas.

Language

Set of conventional spoken or written symbols used for commucation between entities.
So we can see a language as the linking between meaning (semantic side) and expression (syntantic side).

Types of languages

- **Natural languages:** Used for the communication between human beings.
- **Formal languages:** Used by computers and in mathematical areas.

Language

Set of conventional spoken or written symbols used for commucation between entities.
So we can see a language as the linking between meaning (semantic side) and expression (syntantic side).

Types of languages

- **Natural languages:** Used for the communication between human beings.
- **Formal languages:** Used by computers and in mathematical areas.

Differences between natural and formal languages:

1. Computers don't understand natural languages, (normal) humans don't understand computer languages.
2. Formal languages shouldn't have ambiguities, but natural languages do have.

Language Technologies (LT's) are the set of technologies that aim to create software that has some kind of knowledge about natural languages.

Natural language processing (NLP) is the scientific field concerned with the interactions between computers and human by means of natural languages.

Differences between natural and formal languages:

1. Computers don't understand natural languages, (normal) humans don't understand computer languages.
2. Formal languages shouldn't have ambiguities, but natural languages do have.

Language Technologies (LT's) are the set of technologies that aim to create software that has some kind of knowledge about natural languages.

Natural language processing (NLP) is the scientific field concerned with the interactions between computers and human by means of natural languages.

Differences between natural and formal languages:

1. Computers don't understand natural languages, (normal) humans don't understand computer languages.
2. Formal languages shouldn't have ambiguities, but natural languages do have.

Language Technologies (LT's) are the set of technologies that aim to create software that has some kind of knowledge about natural languages.

Natural language processing (NLP) is the scientific field concerned with the interactions between computers and human by means of natural languages.

Differences between natural and formal languages:

1. Computers don't understand natural languages, (normal) humans don't understand computer languages.
2. Formal languages shouldn't have ambiguities, but natural languages do have.

Language Technologies (LT's) are the set of technologies that aim to create software that has some kind of knowledge about natural languages.

Natural language processing (NLP) is the scientific field concerned with the interactions between computers and human by means of natural languages.

Applications of Language Technologies:

1. Machine Translation.
2. Question - Answering.
3. Automatic Text Classification.
4. Automatic Text Summarization.
5. Social Analytics.
6. Sentiment Analysis.
7. **Dialogue Systems.**

Applications of Language Technologies:

1. Machine Translation.
2. Question - Answering.
3. Automatic Text Classification.
4. Automatic Text Summarization.
5. Social Analytics.
6. Sentiment Analysis.
7. **Dialogue Systems.**

Applications of Language Technologies:

1. Machine Translation.
2. Question - Answering.
3. Automatic Text Classification.
4. Automatic Text Summarization.
5. Social Analytics.
6. Sentiment Analysis.
7. **Dialogue Systems.**

Applications of Language Technologies:

1. Machine Translation.
2. Question - Answering.
3. Automatic Text Classification.
4. Automatic Text Summarization.
5. Social Analytics.
6. Sentiment Analysis.
7. **Dialogue Systems.**

Applications of Language Technologies:

1. Machine Translation.
2. Question - Answering.
3. Automatic Text Classification.
4. Automatic Text Summarization.
5. Social Analytics.
6. Sentiment Analysis.
7. **Dialogue Systems.**

Applications of Language Technologies:

1. Machine Translation.
2. Question - Answering.
3. Automatic Text Classification.
4. Automatic Text Summarization.
5. Social Analytics.
6. Sentiment Analysis.
7. **Dialogue Systems.**

Applications of Language Technologies:

1. Machine Translation.
2. Question - Answering.
3. Automatic Text Classification.
4. Automatic Text Summarization.
5. Social Analytics.
6. Sentiment Analysis.
7. **Dialogue Systems.**

Dialogue Systems (written or spoken) are also known as:

- Conversational interfaces.
- Chatbots.

Applications of Dialogue Systems:

1. Information providers.
2. Transactional agents.
3. Educational and learning tutoring.

Dialogue Systems (written or spoken) are also known as:

- Conversational interfaces.
- Chatbots.

Applications of Dialogue Systems:

1. Information providers.
2. Transactional agents.
3. Educational and learning tutoring.

Dialogue Systems (written or spoken) are also known as:

- Conversational interfaces.
- Chatbots.

Applications of Dialogue Systems:

1. Information providers.
2. Transactional agents.
3. Educational and learning tutoring.

Dialogue Systems (written or spoken) are also known as:

- Conversational interfaces.
- Chatbots.

Applications of Dialogue Systems:

1. Information providers.
2. Transactional agents.
3. Educational and learning tutoring.

Dialogue systems main issue

The most difficult challenge in the design of conversational interfaces are related with the highly ambigous nature of spoken languages.

Example

```
Peter come yesterday.
Yesterday Peter come.
```

Two syntatic expressions $\Longleftrightarrow$ One semantic form

Dialogue systems main issue

The most difficult challenge in the design of conversational interfaces are related with the highly ambigous nature of spoken languages.

Example

`Peter come yesterday.`
`Yesterday Peter come.`

Two syntatic expressions $\Longleftrightarrow$ One semantic form

Dialogue systems main issue

The most difficult challenge in the design of conversational interfaces are related with the highly ambigous nature of spoken languages.

Example

```
Peter come yesterday.
Yesterday Peter come.
```

Two syntatic expressions $\Longleftrightarrow$ One semantic form

Example

`Peter said John came yesterday.`

- Was it yesterday when Peter said that?
- Was it yesterday when John came?

One syntatic expression $\Longleftrightarrow$ Two semantic forms

Example

`Peter said John came yesterday.`

- Was it yesterday when Peter said that?
- Was it yesterday when John came?

One syntatic expression $\Longleftrightarrow$ Two semantic forms

Example

`Peter said John came yesterday.`

- Was it yesterday when Peter said that?
- Was it yesterday when John came?

One syntatic expression $\Longleftrightarrow$ Two semantic forms

Humans can deal with these ambiguities applying what is called "psicolinguistic preferences" and, of course, logic and common-sense reasoning:

Example

`Peter said John will come yesterday.`

From the computer point of view this sentence is such ambigous like previous one but humans know that nobody *"will come yesterday"*.

Humans can deal with these ambiguities applying what is called "psicolinguistic preferences" and, of course, logic and common-sense reasoning:

Example

`Peter said John will come yesterday.`

From the computer point of view this sentence is such ambigous like previous one but humans know that nobody *"will come yesterday"*.

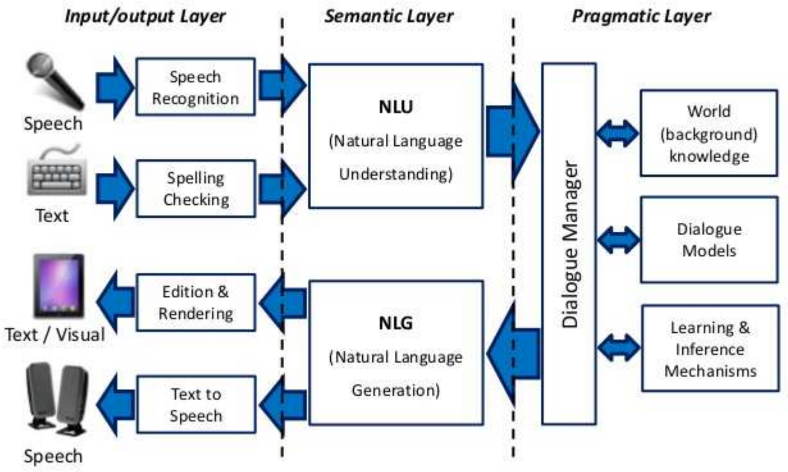
Humans can deal with these ambiguities applying what is called "psicolinguistic preferences" and, of course, logic and common-sense reasoning:

Example

`Peter said John will come yesterday.`

From the computer point of view this sentence is such ambigous like previous one but humans know that nobody *"will come yesterday"*.

Dialogue System architecture



Picture from Institute for Infocomm Research (Singapore)

Natural Language Understanding (NLU)

NLU main goal

The goal of NLU stage is to transform an input string (let's say *user proference*) in an abstract representation of its meaning easier for computer programs to manipulate it, in order to execute some kind of reasoning.

There is a wide variety of possible meaning representations.

- Topic maps.
- Concepts maps.
- Mind maps.
- Onthologies.
- **Feature structures.**

Natural Language Understanding (NLU)

NLU main goal

The goal of NLU stage is to transform an input string (let's say *user proference*) in an abstract representation of its meaning easier for computer programs to manipulate it, in order to execute some kind of reasoning.

There is a wide variety of possible meaning representations.

- Topic maps.
- Concepts maps.
- Mind maps.
- Onthologies.
- **Feature structures.**

Example

$$\texttt{John came yesterday} \longrightarrow \begin{bmatrix} \textsc{Subject:} & \texttt{John} \\ \textsc{Action:} & \texttt{come} \\ \textsc{Tense:} & \texttt{past} \\ \textsc{OffsetDate:} & \texttt{-1 day} \end{bmatrix}$$

Example

$$\texttt{John will talk in two days} \longrightarrow \begin{bmatrix} \textsc{Subject:} & \texttt{John} \\ \textsc{Action:} & \texttt{talk} \\ \textsc{Tense:} & \texttt{future} \\ \textsc{OffsetDate:} & \texttt{+2 day} \end{bmatrix}$$

Feature structures

- A feature structure is a set of features.
- With no particular order between them.
- Every feature may have (but it's not required) an associated value.
- The value associated to every feature can be **atomic** or **complex**.

$$\texttt{comes} \longrightarrow \begin{bmatrix} \textsc{Action:} & \texttt{come} \\ \textsc{Tense:} & \texttt{present} \\ \textsc{Agreement:} & \begin{bmatrix} \textsc{Number:} & \texttt{singular} \\ \textsc{Person:} & 3 \end{bmatrix} \end{bmatrix}$$

Feature structures

- A feature structure is a set of features.
- With no particular order between them.
- Every feature may have (but it's not required) an associated value.
- The value associated to every feature can be **atomic** or **complex**.

$$\texttt{comes} \longrightarrow \begin{bmatrix} \textsc{Action:} & \texttt{come} \\ \textsc{Tense:} & \texttt{present} \\ \textsc{Agreement:} & \begin{bmatrix} \textsc{Number:} & \texttt{singular} \\ \textsc{Person:} & 3 \end{bmatrix} \end{bmatrix}$$

NLU components

Trying to convert user proference to feature structures is not trivial. So we need to divide the process in some functional modules:

- Tokenization.
- Speller checker.
- Part Of Speech tagging (POS tagging).
- Parsing.
- Unifier.

NLU components

Trying to convert user proference to feature structures is not trivial. So we need to divide the process in some functional modules:

- Tokenization.
- Speller checker.
- Part Of Speech tagging (POS tagging).
- Parsing.
- Unifier.

Tokenization

Goal

Convert a sequence of characters into a sequence of tokens.

We must take into account:

- Separators: (␣-_)
- Punctuation marks: (,.;:!?)
- Special symbols: ($€%º)
- Numbers and its own separators: (1234,.)
- Alphanumeric codes: (ES772024···)

Example

	tk1		tk2		tk3		tk4		tk5		tk6	
·	The	·	dog	·	is	·	in	·	the	·	park	·
0		1		2		3		4		5		6

Tokenization

Goal

Convert a sequence of characters into a sequence of tokens.

We must take into account:

- Separators: (␣-_)
- Punctuation marks: (,.;:!?)
- Special symbols: ($€%°)
- Numbers and its own separators: (1234,.)
- Alphanumeric codes: (ES772024···)

Example

	tk1		tk2		tk3		tk4		tk5		tk6	
·	The	·	dog	·	is	·	in	·	the	·	park	·
0		1		2		3		4		5		6

Tokenization

Goal

Convert a sequence of characters into a sequence of tokens.

We must take into account:

- Separators: (␣-_)
- Punctuation marks: (,.;:!?)
- Special symbols: ($€%°)
- Numbers and its own separators: (1234,.)
- Alphanumeric codes: (ES772024···)

Example

	tk1		tk2		tk3		tk4		tk5		tk6	
·	The	·	dog	·	is	·	in	·	the	·	park	·
0		1		2		3		4		5		6

Speller checker (only in written dialogue systems)

London

- Insertion: `Loondon`
- Deletion: `Lndon`
- Substitution: `Lpndon`
- Switching: `Lonodn`
- Bad separators: `Lon don`

Part Of Speech tagging (POS tagging)

Goal

To mark up lexical items with some lexical category depending on its definition and the context.

In natural language we can have some common lexical categories:

- Determiners: `a, the`
- Nouns: `London, dog`
- Pronouns: `you, me`
- Prepositions: `to, for`
- Adjectives: `blue, long`

Part Of Speech tagging (POS tagging)

Goal

To mark up lexical items with some lexical category depending on its definition and the context.

In natural language we can have some common lexical categories:

- Determiners: `a, the`
- Nouns: `London, dog`
- Pronouns: `you, me`
- Prepositions: `to, for`
- Adjectives: `blue, long`

POS tagging

So in the lexicon definition we can classify lexical items into categories:

- `("the", det)`
- `("dog", noun)`
- `("me", pronoun)`
- `("to", preposition)`

But in natural languages we can have several lexical categories corresponding to a single lexical item (especially in little inflectional ones, like in english):

- `("plans", noun)` $\longrightarrow$ plural of plan.
- `("plans", verb)` $\longrightarrow$ present of third person (singular) of verb to plan.

POS tagging

So in the lexicon definition we can classify lexical items into categories:

- `("the", det)`
- `("dog", noun)`
- `("me", pronoun)`
- `("to", preposition)`

But in natural languages we can have several lexical categories corresponding to a single lexical item (especially in little inflectional ones, like in english):

- `("plans", noun)` $\longrightarrow$ plural of plan.
- `("plans", verb)` $\longrightarrow$ present of third person (singular) of verb to plan.

POS tagging: Garden path problem

Example

The	government	plans	to	raise	taxes	...
det	noun	**verb**	prep	verb	noun	

Example

The	government	plans	to	raise	taxes	were	defeated
det	noun	**noun**	prep	verb	noun	verb	adj

POS tagging: Garden path problem

Example

The	government	plans	to	raise	taxes	...
det	noun	**verb**	prep	verb	noun	

Example

The	government	plans	to	raise	taxes	were	defeated
det	noun	**noun**	prep	verb	noun	verb	adj

Formal grammars

- Only study the purely syntactical aspects of language.
- **Alphabet:** Set of terminal symbols. $\{a, b\}$
- **Sentences:** Strings of symbols. $a, b, aa, ab, bb, ba, aaa, \cdots$
- **Language:** Set of sentences. $L = \{ab, aabb, aaabbb\}$
- **Formal grammar:** Set of formation rules used to define a language. $\{S \rightarrow aA, A \rightarrow bA, S \rightarrow \varepsilon \cdots\}$ Where S, A are non-terminal symbols.
- Depending on the syntax of production rules, grammars can be classified (Noam Chomsky, 1956).

Chomsky grammar hierarchy

Type	Grammar accepted	Rules	Observations
Type 0	Unrestricted grammar	$X \to Y$	$X, Y \in (N \cup T)^*$
Type 1	Context-sensitive grammar	$\alpha X \beta \to \alpha a \beta$	$X \in N$ $\alpha, a, \beta \in (N \cup T)^*$
Type 2	Context-free grammar	$X \to a$	$X \in N$ $a \in (N \cup T)^*$
Type 3	Regular grammar	$X \to a$ $X \to aY$	$X, Y \in N$ $X \in T$

Where:

- N is the set of non-terminal symbols.
- T is the set of terminal symbols.
- S^* is a string of elements in set S.

Easy context-free grammar example: $a^n b^m$

Consider:

- **Alphabet:** $\{a, b\}$
- **Language:** $L = \{a^n b^m\}\ n, m \geq 1$ (i.e. all strings with at least one "a" followed by at least one "b").
- **Grammar:**

$$\begin{aligned} S &\rightarrow a\ A\ b\ B \\ A &\rightarrow \varepsilon \\ A &\rightarrow a\ A \\ B &\rightarrow \varepsilon \\ B &\rightarrow b\ B \end{aligned}$$

And let's see that a grammar defined with such production rules can be used to **generate** or either **recognize** the target language.

Generating $a^n b^m$ sentences

S

$$S \rightarrow a\ A\ b\ B$$
$$A \rightarrow \varepsilon$$
$$A \rightarrow a\ A$$
$$B \rightarrow \varepsilon$$
$$B \rightarrow b\ B$$

Generating $a^n b^m$ sentences

S

$$S \rightarrow a\ A\ b\ B$$
$$A \rightarrow \varepsilon$$
$$A \rightarrow a\ A$$
$$B \rightarrow \varepsilon$$
$$B \rightarrow b\ B$$