Overview

- META is a tool for text analysis which implements some NLP functionalities.
- META provides the tools for semantic indexing and exploits WordNet as knowledge source in Word Sense Disambiguation processing.

Architecture
Plug-In

- The Plug-In is a component that is able to import data from different sources and inserts them in a database. Then, data are processed by META.
- In order to integrate this component into META it is necessary to:
  - develop a class that implements the interface `plugs.plugin.Plugin` and inserts it into package `plugs.plugin.pluginImpl.<new_package>`
- The developed Plug-In is able to import text from:
  - PDF, DOC, HTML, OpenOffice, XLS, RTF, TXT, XML

META Text-processing

- Automatic language recognition
- Text normalization
- Tokenization
- Stop word elimination
- Stemming
- Text summarization (TF-IDF based)
- POS-tagging
  - Based on ACOPOST T3 (HMM - Hidden Markov Model)
- Entity recognize
- WSD (Word Sense Disambiguation)
  - The WSD algorithm was evaluated by SENSEVAL framework
- TF-IDF parametric on features

Plug-Out

- The Plug-Out is a component that is able to export META output in a different format.
- In order to integrate this component into META it is necessary to:
  - develop a class that implements the interface `plugs.plugOut.PlugOut` and inserts it into package `plugs.plugOut.plugOutImpl.<new_package>`
- The developed Plug-Out is able to export META output in:
  - Database, RDF, XML, TXT
  - different indexing models: Vector Space Model, LSI (Latent Semantic Indexing), …
Future work

- Improve entity recognition through Machine Learning techniques
  - Currently, META includes an annotation tool for the development of tagged-corpus
- Exploit domain ontologies in the indexing process, in order to map entities to instances available in the ontologies

JIGSAW – WSD algorithm

- JIGSAW is a Word Sense Disambiguation algorithm that uses three different kinds of techniques to disambiguate nouns, verbs, adjectives and adverbs.
- Input:
  - $W = \{w_1, w_2, \ldots, w_n\}$ is the set of all words in text which will be disambiguated
  - synVerb is the algorithm for disambiguating verbs
  - synNoun is the algorithm for disambiguating nouns
  - synAdjAdv is a Lesk-based algorithm for disambiguating adjectives and adverbs

```
for each $w_i \in W$
  if ($w_i$ is a verb) execute synVerb on $w_i$

for each $w_i \in W$
  if ($w_i$ is a noun) execute synNoun on $w_i$
  else if ($w_i$ is a adjective or $w_i$ is a adverb)
    execute synAdjAdv on $w_i$
```

- $W = \{w_1, w_2, \ldots, w_n\}$ is the set of all words in text which will be disambiguated
- synVerb is the algorithm for disambiguating verbs
- synNoun is the algorithm for disambiguating nouns
- synAdjAdv is a Lesk-based algorithm for disambiguating adjectives and adverbs
LB (Lesk-based) algorithm

- LB algorithm uses WordNet glosses to create a relation between grammatical categories and the context.
- This allows to create semantic relations with words in the context of the word to be disambiguated (target word).
- The algorithm measures the similarity between glosses of target words and glosses of context words. This operation is performed in 4 steps:
  - keep target word context
  - create the string (glossContext) which contains the glosses of words in the context.
  - for each meaning of the target word, create a string (glossTarget) which contains the gloss of the meaning and all words related with target word and compute the similarity between glossContext and glossTarget.
  - select the meaning for target word which has the max similarity wrt the glossContext.

I play basketball.

contextGloss: participate in games or sports We played hockey all afternoon Play cards Pele plays for Brasilian teams in many important matchs act or have an effect in specified way...

targetGloss #1: basketball game a game played on a court by two opposing teams of five players points are scored by throwing the basketball through an elevated horizontal hoop court game athletic game sports athletic diversion recreation activity game basketball shot rebound fastbreak dribbling tip-off tap-off basketball play half

targetGloss #2: an inflated ball used in playing basketball ball game equipment instrumentality object whole

Algorithm for disambiguating nouns

- The algorithm is based on observation that when 2 words have more than one meaning, the common minimal meaning (Most Specific Subsumer - MSS - in the WordNet hierarchy) supplies information on the most appropriate meaning for each word.
- Let $W$ be the set of nouns in a sentence and $S$ the set of all possible meanings, the goal is to define a function $\varphi$ (between 0 and 1) that computes the probability of the meaning $s_j \in S$ for the word $w_i \in W$.
- For each word $w_i$ in $W$, the algorithm selects the most appropriate meaning $s_{i\text{\_max}}$.

$$s_{i\text{\_max}} = \arg \max_j \varphi(s_j)$$
Algorithm for disambiguating nouns

- In order to compute $\phi$, the MSS, the distance between words and the similarity are exploited.

$$\text{sim}_{\phi} = \max \left( -\log \left( \frac{N_a}{2D} \right) \right)$$

Leacock-Chodorow

$N_a = 5$ and $D = 16$ (in WordNet 1.7.1)

$\text{sim}(\text{cat,mouse}) = -\log(5/32) = 0.806$

Most Specific Subsumer

PLACENTAL MAMMAL

---

Algorithm for disambiguating nouns

Let $d$ be the noun to be disambiguated:

- for $j = 1, ..., |\text{CONTEXT}|$
  - $v[j] \leftarrow \text{sim}(d, w_j) \times G(\text{position}(d) - \text{position}(w_j))$
  - for $k = 1, ..., |S|$
    - if ($c[j]$ is hypernym of $k$-th meaning of $d$)
      - $\text{support}[k] \leftarrow \text{support}[k] + v[j]$
      - $\text{normalization} \leftarrow \text{normalization} + v[j]$
  - $\text{posmax} \leftarrow 1$
  - $\phi[k] \leftarrow \alpha \times \text{support}[k] / \text{normalization} + \beta \times R(k)$
    - if ($\phi[k] > \phi[\text{posmax}]$)
      - $\text{posmax} \leftarrow k$

Best meaning of $d$ is equal to $s_{\text{posmax}} \in S$

---

Algorithm for disambiguating nouns

Let $d$ be the noun to be disambiguated:

- $v[j] \leftarrow \text{sim}(d, w_j) \times G((\text{position}(d) - \text{position}(w_j))$

- $\text{G}$ is the Gaussian function which takes into account the difference between the noun's positions in the text.

- if ($\text{normalization}=0$)
  - $\phi[k] \leftarrow a / |S| + \beta \times R(k)$
  - if ($\phi[k] > \phi[\text{max}]$
    - $\text{max} \leftarrow \phi[k]$
    - $\text{posmax} \leftarrow k$

Best meaning of $d$ is equal to $s_{\text{posmax}} \in S$
Algorithm for disambiguating nouns

Let \( d \) be the noun to be disambiguated:

\[
\text{for } j=1,\ldots,|\text{CONTEXT}| \quad v[j] \leftarrow \text{sim}(d, w_j) \times \text{G(position}(d) - \text{position}(w_j))
\]

\[
c[j] \leftarrow \text{most_specific_subsumer}(d, w_j)
\]

\[
\text{for } k=1,\ldots,|S|
\]

\[
\text{if } (c[j] \text{ is hypernym of } k\text{-th meaning of } d)
\]

\[
support[k] \leftarrow support[k] + v[j]
\]

\[
\text{end if}
\]

\[
\text{end for}
\]

\[
posmax \leftarrow 1
\]

\[
\max \phi \leftarrow -\text{MAX_DOUBLE}
\]

\[
\text{for } k=1,\ldots,|S|
\]

\[
\text{if } (\text{normalization} \neq 0)
\]

\[
\phi[k] \leftarrow \alpha \times \frac{support[k]}{\text{normalization}} + \beta \times R(k)
\]

\[
\text{else}
\]

\[
\phi[k] \leftarrow \frac{\alpha}{|S|} + \beta \times R(k)
\]

\[
\text{if } (\phi[k] > \max \phi)
\]

\[
\max \phi \leftarrow \phi[k]
\]

\[
posmax \leftarrow k
\]

\[
\text{end if}
\]

\[
\text{end for}
\]

\[
\text{best meaning of } d \text{ is equal to } s_{posmax} \in S
\]

Verbs’ algorithm

In order to disambiguate a verb \( w \), the algorithm uses the nouns in the context of \( w \) and the nouns into the glosses and example phrases for \( w \) that WordNet utilizes to explain the use of the verb \( w \).

The algorithm extracts the nouns in the gloss and in the examples for each meaning of \( w \). \( nouns(w,k) \) is the set of nouns into the description of meaning \( k \) for \( w \). \( N \) is the set of nouns in the context in which \( w \) appears.
**Verbs’ algorithm**

- Each noun $w_i$ in $N$ is compared with each noun $w_j$ in $nouns(w,k)$ by computing a similarity value between $w_i$ and $w_j$

\[
\text{sim}(w_i, w_j) = -\ln \left( \frac{N_a}{2D} \right) - \ln \left( \frac{1}{2D} \right)
\]

**Verbs’ algorithm**

- The algorithm computes the similarity for each pair of nouns and then, for the $i$-th noun $w_i$ in $N$, computes the following value: $\max_i = \max (\text{sim}(w_i, w_j))$ for $j = 1, 2, ..., m$, where $m$ is $|nouns(w,k)|$.
- The function $\phi$ is computed by the following equation:

\[
\phi(v, k) = R(k) \times \sum_i G(d_i) \times \max_i
\]
- $R(k)$ is the same function in the nouns’ algorithm but has a different coefficient (0.9)

**Evaluation**

- All algorithms have been evaluated with SENSEVAL framework

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>LB (nouns)</td>
<td>0.246</td>
</tr>
<tr>
<td>LB (verbs)</td>
<td>0.295</td>
</tr>
<tr>
<td>LB (adjectives)</td>
<td>0.403</td>
</tr>
<tr>
<td>Verbs</td>
<td>0.405</td>
</tr>
<tr>
<td>Nouns</td>
<td>0.319</td>
</tr>
</tbody>
</table>
All algorithms have been evaluated with the SENSEVAL framework.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>LB (nouns)</td>
<td>0.246</td>
</tr>
</tbody>
</table>

**JIGSAW precision**
- sample task = 0.376
- all word task = 0.52