MontyLingua (and ConceptNet) to simplify natural text processing tasks

Maurice Ling

Format of Presentation

- Common text processing tasks
- Text processing pipeline
- What is MontyLingua and where it fits in?
- Comparing MontyLingua with GATE and NLTK
- Relationship between MontyLingua and ConceptNet
- Some uses of ConceptNet

Common Text Processing Tasks

- Ad hoc query
- Document classification (pre-defined query)
- Text summarization
- Information extraction from text
- Finding associations from text (text mining)
- Emotions sensing

Common Text Processing Tasks

- Nouns (actors) and relations between them
  - Ad hoc query, text mining, information extraction
- Nouns (actors) and adjectives (descriptives)
  - Text mining, information extraction, summary
- Mapping onto epistemological knowledge
  - Emotion sensing, query expansion, information extraction, text mining

General Text Processing Pipeline

- Text
- Tokenize (split into words and punctuations)
  - Mr., a.m., $5.24, M.B.B.S, MB,BS, hexa-1,2-ol
- Parts of speech tagging (syntactic roles)
- Breaking sentences into phrases (chunking)
- Extract information

Converting unstructured format (text) into structured format (specific lists)

Where MontyLingua fits in?

- MontyLingua is a text processor (Python and Jython)
- Takes in text
- Tokenize sentences
- Tokenize words
- Parts of speech tagging
- Chunk parsing

- Outputs a set of lists

MontyLingua's Outputs

- List of Nouns
- List of Verbs
- List of Adjectives
- List of Prepositions
- Subject-Verb-Objects
- Summary

- Do an example

Can I change the components of MontyLingua?

- Yes.

```
def jist(self, text):
        sentences = self.split_sentences(text)
        tokenized = map(self.tokenize, sentences)
        tagged = map(self.tag_tokenized, tokenized)
        chunked = map(self.chunk_tagged, tagged)
        #print "CHUNKED: " + string.join(chunked,'\n          ')
        extracted = map(self.extract_info, chunked)
        return extracted
```

How does MontyLingua compare with other tools (e.g. GATE and NLTK)?

Main Differences between MontyLingua and GATE

- MontyLingua
  - Natural language processing
  - Full pipeline
  - No configurations needed
- GATE
  - Template matching engine
  - Set of components
  - Configure templates and components

Main Differences between MontyLingua and NLTK

- MontyLingua
  - Full pipeline
  - High level
  - Simplify text processing

- NLTK
  - Set of components
  - Low level
  - Teaching and research toolkit

Comparing MontyLingua and NLTK is like comparing a GUI application with a GUI toolkit,
you can build MontyLingua from NLTK.

Part 3
ConceptNet and its uses in text processing

What is ConceptNet?

- Database of epistemological knowledge
- Learnt through Open Minds project
- Formally known as OMCSNet
- MontyLingua is the text processor to ConceptNet

What can ConceptNet provide?

- Estimate the topics of the input text
- Estimate the concepts of input text
- Estimate the mood of the input text
- Related concepts
- Context of the text

Short demonstration

In Summary

- MontyLingua is a text processing module to process unstructured text into structured format
- Components of MontyLingua can be changed quite easily
- MontyLingua is an integral part of ConceptNet
- ConceptNet may has a lot of offer by merging epistemology with natural text processing (such as, processing interview transcripts)