Course Objectives

The main objective of the course is to learn how to develop practical computer systems capable of performing intelligent tasks on natural language: analyze, understand and generate written text. This task requires learning material from several fields: linguistics, machine learning with a focus on deep learning techniques and statistical analysis, and core natural language techniques.

- Acquire basic understanding of linguistic concepts and natural language complexity:
  - *variability* (the possibility to express the same meaning in many different ways) and
  - *ambiguity* (the fact that a single expression can refer to many different meanings in different contexts); levels of linguistic description (word, sentence, text; morphology, syntax, semantics, pragmatics). Schools of linguistic analysis (functional, distributional, Chomskyan); Empirical methods in Linguistics; Lexical semantics; Syntactic description; Natural language semantics issues.

- Acquire basic understanding of machine learning techniques as applied to language:
  - supervised vs. unsupervised methods; training vs. testing; classification; regression;
  - distributions, KL-divergence; Perceptron; HMM; CRF; Deep learning methods: back propagation, word embeddings (word2vec), RNN; LSTM; Sequence to Sequence models;

- Acquire hands-on proficiency in applying Natural language processing techniques using Python as a development platform: word and sentence tokenization; parts of speech tagging; lemmatization and morphological analysis; chunking; named entity recognition; n-gram language models; probabilistic context free grammars; probabilistic dependency grammars; parsing accuracy metrics; Summarization; Text generation.
Course Requirements
1. 13 x 2-hours lectures / Participation mandatory
2. 2 programming assignments in pairs (about 30 hours each) [each 30% of the grade]
3. Final project in groups of 2 to 4 students [40% of the grade] with programming component and written report.

Detailed Syllabus

Introduction

1. **What is Natural Language Processing**: Practical applications vs. theoretical investigation; Tasks: text analysis, understanding, generation;
2. **Applications**: machine translation, summarization, semantic search, report generation, text simplification, sentiment analysis, information extraction;
3. **Linguistic description**: variability vs. ambiguity; prescriptive vs. descriptive linguistics; levels of description (letter, word, sentence, text); morphology, syntax, semantics, pragmatics; Linguistic method: empirical methods, knowledge-based methods, psycho-linguistics, cognitive basis; socio-linguistics. Diachronic vs. synchronic description; universals in language;
4. **Mathematical foundations**:
   1. **Formal languages**: language hierarchy; regular, context-free, mildly context-dependent languages; natural language and context-freeness;
   2. **Statistical analysis**: corpus based linguistics; distribution, random variables; word types vs. word instances; Zipf's distribution and power laws (long tail effects); variance and bias; information theory: entropy and coding; mutual information; KL-divergence; hypothesis testing; Statistical inference; Markov Models;

From Words to Tags: Building a Classifier for Parts of Speech Tagging
1. Word Tokenization, Sentence tokenization, text encoding (Unicode).
2. Counting things: frequency distribution; perplexity
3. Corpora: characterization, examples (the Brown Corpus); descriptive metrics.
4. Parts of speech tagging: defining the task, terminology, measuring success, accuracy, training, test;
5. Classification methods: which knowledge source is exploited; training method; evaluation metrics: accuracy, precision, recall, F-measure.
6. Combining classifiers: backoff, ensemble methods;
7. Knowledge sources for Parts of Speech tagging: lexical, morphological, context;
8. Language models: n-gram models; estimation and smoothing methods;
10. Neural Network Language Models: RNNs, CNNs; Using CNNs for Morphological analysis; LSTMs.

Parsing
1. Why parsing is interesting and difficult: applications; structural ambiguities (preposition attachment, control, conjunctions);
2. Constituent vs. Dependency parsing;
3. Context Free Parsing: Chomsky Normal Form; CKY parsing; Shift-Reduce parsing; accuracy metrics for constituency parsing;
4. Probabilistic CFG Parsing: Viterbi algorithm; Charniak's parser; Independence assumptions in parsing; the importance of lexical and bi-lexical features;
5. Introducing mild context in PCFGs: Split categories; markovization;
7. Neural Network models for Dependency Parsing.

**Summarization**
1. Definition of the summarization task: single vs. multiple document; informative vs. indicative summarization; generic vs. query-focused.
2. Evaluation metrics for summarization: human, gold-standard, task-based; coherency, faithfulness, redundancy; ROUGE metric; Pyramid methods.
3. SUM-Base: impact of frequency;
4. KL-SUM: minimizing vocabulary divergence;
5. Graph-based methods: centroid, Lexrank.
7. Neural Network models for headline generation.

**Text Generation**
1. Text generation vs. interpretation: where do we start from? Semantic representation (logical forms); text to text generation;
2. Architecture for generation: content planning; micro-planning (aggregation, reference planning); rhetorical planning; lexical choice; syntactic realization.
3. Content planning: communication activities and planning;
4. Lexical choice methods: lexical resources for generation; statistical language models of lexical choice;
5. Syntactic realization: unification-based grammars; statistical models of syntactic realization;
6. Paraphrase generation: paraphrase detection; collecting a dataset of paraphrases; learning paraphrasing rules.

**Semantics**
1. Computing semantic representations from natural language and performing inference; why is semantic representation interesting? Semantic analysis vs. textual entailment (variability and ambiguity).
2. Semantic representations: first order logic; models; lambda calculus; Montague grammars and mapping syntax to semantic representations. Discourse Representation Theory (DRT)
3. Underspecified representations: scope ambiguities; quantifier raising; Cooper storage; Hole semantics;
4. Lexical semantics: Wordnet; distributional semantics; Word Embeddings; word2vec;
5. Compositional models of distributional semantics; learning composition functions over embeddings;
6. Text entailment: task definition; resources and models;
7. Combining statistical dependency syntax parsing and semantic analysis;
References
Textbooks


- **Natural Language Processing with Python** by Steven Bird, Ewan Klein and Ed Loper, O’Reilly, 2009

Other References

- **Pattern Recognition and Machine Learning** by Chris Bishop, Springer, 2007


- **Deep Learning** by Yoshua Bengio and Ian J. Goodfellow and Aaron Courville; 2015; http://www.iro.umontreal.ca/~bengioy/dlbook/