NLP & IR 2013 PROJECTS

Question Classifier

Task: classify a question into one of the pre-defined categories, e.g. HUM - human, LOC - location, NUM - number, etc.
Data: TREC QA 2005 from (Li & Roth, 2005)
Software: SVM with Tree Kernels (available under the course repo)
State-of-the-art: Moschitti + PhD student paper
Ideas: use of the dependency parsers, enable fine grained classification collecting more data, use the EMNLP 2011 Model (LSI, PTK, SPTK)
References: Danilo Croce, Alessandro Moschitti, and Roberto Basili. Structured lexical similarity via convolution kernels on dependency trees. In Proceedings of EMNLP, 2011

Focus Identification

Task: automatically detect the key/focus word in the question that directly points to what is being asked.
Data: GeoMooney, Seco600, Radescu (available under the course repo)
Software: SVM-TK
Focus classifier, additional data, even manually annotated, to learn a model. Experiment with kernel methods and parametrization.

Wikification

Task: identify important entities and concepts in text, disambiguates them and link them to the related Wikipedia concepts.
Example: demo of the Illinois Wikifier system
Approach: One possible approach is to run a shallow parser, i.e. chunker, to identify noun phrases. Then for each noun phrase extract all possible subsets which are then queried across a dictionary, where each wikipedia concept corresponds to a number of mentions. To improve the accuracy of disambiguation use Lucene to index a Wikipedia dump. Given a sentence return top N wiki concepts, which can be used as a good prior on the top entities mentioned in the text.
Useful papers:

STS-2012 challenge

More information about the challenge and related papers: STS website and challenge wiki

Datasets: data and the baseline system is on the repo

- Semantic features extracted from Wikipedia

Task: Use Wikipedia to define semantic similarity features for a pair of text snippets, i.e. sentences.

Possible ideas:

* Similarity based on top concepts for a given sentence returned from the Search Engine -- Explicit Semantic Analysis (available implementations)
* Exploiting Wikipedia categories to establish the matching

- Refined syntactic/shallow semantic trees.

Augment constituency/dependency parse trees with some kind of semantic annotations, e.g. WordNet class/hypernyms/etc., NERs, SuperSense tags.

- Exploring SRL for semantic similarity.

SRL demo: in class

Tools for SRL: ClearNLP, Siena

Build a representation using SRL to construct shallow semantic structures for a given sentence + tree kernels to generate the feature spaces.

- Sentence type identification.

Task: Perform clustering/classification to learn the sentence types. Explore different sets of lexical/syntactic/semantic features to learn an accurate classifier. The obtained features can then be pluggged into the final classifier. Use additional corpora to learn useful priors.

Useful Software

Machine Learning

SciKits - lots of machine learning tools for supervised and unsupervised learning
Mallet (Java) - classification, clustering, topic modeling (LSA and LDA), etc.
SVM-TK (C and Java (still in development)) - grab from them course repo
LibSVM (C and Java)
Gensim (Python)

NLP

Stanford CoreNLP (Java): sentence segmentation, tokenization, part-of-speech, ner, syntactic and dependency parsing
NLTK

Final Report

Guidelines for the final paper: http://disi.unitn.it/moschitti/Projects/reports.html