WoSIT: A Word Sense Induction Toolkit for Search Result Clustering and Diversification

Daniele Vannella, Tiziano Flati and Roberto Navigli

http://icl.uniroma1.it/wosit

Introduction
The retrieval of any given piece of information on the Web is an arduous task which challenges even prominent search engines. It has been shown, however, that the automatic acquisition of the meanings of a word of interest (Word Sense Induction) can be successfully integrated into search result clustering and diversification so as to outperform non-semantic state-of-the-art Web clustering systems (Di Marco and Navigli, 2013).

WoSIT Workflow
1. WSI - Word Sense Induction;
2. Semantically-enhanced search result clustering and diversification;
3. Evaluation.

1) WoSIT - Word Sense Induction
The first phase consists of the automatic identification of the senses of a query of interest, i.e. the task of Word Sense Induction. The toolkit provides ready-to-use implementations of several graph-based algorithms that work with word co-occurrences. All these algorithms carry out WSI in two steps:

a) co-occurrence graph construction;
b) discovery of word senses.

3) WoSIT – Evaluation
The final component of our workflow is the evaluation of WSI when integrated into search result clustering and diversification (already used by Navigli and Vannella (2013)). Two kinds of evaluations are carried out (Di Marco and Navigli, 2013):

1. Evaluation of the clustering quality
   • Rand Index, Adjusted Rand Index, Jaccard Index
2. Evaluation of the clustering diversity
   • S-recall@K, S-precision@R

JAVA CODE SNIPPET
1. Dataset searchResults = Dataset.getInstance();
2. DBCConfiguration db = DBCConfiguration.getInstance();
3. for(String targetWord : dataset.getQueries()) {
   int WordGraph g = WordGraph.createWordGraph(targetWord, searchResults, db);
   BMST mst = new BMST(g);
   mst.makeClustering();
   SnippetAssociator snippetAssociator = SnippetAssociator.getInstance();
   SnippetClustering snippetClustering = snippetAssociator.associateSnippet(targetWord, searchResults, mst.getClustering(), AssociationMetric.WORD_OVERLAP);
   snippetClustering.exportOutput(outputMST.txt, true);
   WSIIEvaluator.evaluate(searchResults, "output/outputMST.txt");
}

References

The authors gratefully acknowledge the support of ERC Starting Grant MultiSER No. 259234.