CACTUS: A User-friendly Toolkit for Semantic Categorization and Clustering in the Open Domain

Emily Jamison
Department of Linguistics,
The Ohio State University,
Columbus, OH 43210, USA
jamison@ling.osu.edu

Abstract

In this paper, we present a tool for the semantic categorization and clustering of open-domain named entities (NEs) and common nouns (CNs), the Categorization And Clustering Tool for User-defined Semantic classes, or *CACTUS*. The tool performs either of the two tasks, using Hearst-style (Hearst, 1992) search queries with a web search engine: assignment of NEs/CNs to user-provided semantic classes, and semantic clustering of the NEs/CNs into a user-defined number of classes. We evaluate our approach on a dataset of 400 NEs/CNs and obtain encouraging results indicating that webquery-based semantic categorization in the (nearly) untrained open domain acheives accuracy comparable to supervised, limited-domain systems.

1 Introduction

Semantic categorization[1] and clustering (i.e., "JetBlue" is a kind of COMPANY , and "JetBlue" and "AirTran" are more similar than "JetBlue" and "Bill Gates") are used in tasks such as coreference resolution and referring expression generation. Existing semantic dictionaries may lack the coverage to handle large open domains or rapidly changing categories: (Vieira et al., 2000) found that of antecedent/anaphoric coreferent pairs in the WSJ, only 56% in hyponymy relations were in WordNet as direct or inherited links.

Several named entity recognition shared tasks, such as CoNLL 2003 and BioCreative 2004, have focused community resources on the task of automatically identifying and categorizing named entities (NEs). However, these tasks use a fixed set of categories and a significant training set; the systems produced cannot be used with other categories for other purposes. The main contribution of this work is to present an adaptable, out-of-box tool for use in semantic classification and categorization with no training required. CACTUS performs categorization of both NEs *and* common nouns (CNs) with user-defined semantic classes, as well as clustering of the NEs/CNs into a user-defined number of clusters. To use CACTUS, the user simply supplies a list of NEs and/or CNs, and a list of desired categories and/or n number of desired clusters. No training is required.

[1] i.e., selection of the best hypernym from a list.

Hearst-style frames from Cimiano et al. (2007)
EXAMPLE is a kind of CLASS
EXAMPLE and other CLASSES
CLASSES such as EXAMPLE
CLASSES , including EXAMPLE

Table 1: Example frames used.

2 Algorithms Used

CACTUS combines 3 different categorization algorithms. Hearst (1992) showed that hyponymy information could be collected by using a series of hand-crafted frames to search a corpus (here, the internet). For the first categorization algorithm, the user-provided classes and the NE/CN in question are inserted into the Hearst-style frames developed by (Cimiano, 2007) (shown in table 1). The class associated with the highest web search count cumulatively from its frames is declared the correct class.

Kozareva et al. (2008) used a doubly-anchored pattern to generate a list of class members from web searches. To adapt this technique for semantic categorization, CACTUS's second algorithm uses the user-provided semantic class and gold-standard class member to generate 10 other most

frequent class members from 100 search results, using Kozareva's basic search term technique. Search counts for all 11 phrases per class are then collected. Class ranking is the same as the Hearst-style algorithm.

To boost coverage, a third algorithm based on conditional probability, the co-occurrence algorithm, is added. This algorithm chooses the class with the highest normalized class and NE/CN frequency, as shown in equation 1.

$$Score_{CLASS} = \frac{webcount_{CLASS+NEorCN}}{webcount_{CLASS}} \quad (1)$$

CACTUS is intended for the open domain, so supervised algorithms were avoided. CACTUS uses a simple back-off strategy to decide final categorization: if the high-precision, low coverage Kozareva algorithm produces a result, then it is used; if not, CACTUS backs off to the Hearst-style algorithm; if it produces no result, CACTUS backs off to the low-precision, high coverage co-occurrence algorithm.

For clustering, CACTUS collects the 10 most frequent categories from 100 web results, and clusters the NEs/CNs with Cluto (Steinbach et al., 2000), using the categories as predicates, similarly to (Evans, 2003).

3 Evaluation

To evaluate CACTUS's categorizer in the open domain, we created a corpus with 400 NEs/CNs, including 100 countries, 100 cities, 60 heads-of-state, 20 composers, 100 animals, and 20 trees. Overall F-measure was 95.37%, and coverage was 99.75%; a majority-class baseline is 25.00%. We also evaluated CACTUS's categorizer on a subset of the CoNLL 2003 dataset that included 10 people, 10 organizations, and 10 locations, with the resulting F-measure 76.67%; the majority class baseline is 33.33%.

CACTUS's clustering requires high NE/CN mention count. Only 17 of 30 items in the CoNLL subdataset could be clustered. Precision was 64.71%; f-measure was 46.81%. The random baseline was 33.33% precision and 50.00% f-measure.

Type of Categorizer	Prec.	Cov.	F-meas.
majority-class	25.00%	100.00%	25.00%
Kozareva	96.59%[a]	22.00%	34.84%
Hearst-style	97.56%	92.25%	93.63%
Co-Occurrence	73.68%	99.75%	73.59%
Total with back-off	95.49%	99.75%	95.37%

[a]Miscategorizations include Hong Kong:country, Singapore:country, horse:tree (" 'trees such as the horse' chestnut").

Table 2: Results of CACTUS Categorizer on 400 NE/CN dataset.

4 Future Work

The goal of CACTUS is to provide semantic information in the open domain. Therefore, future specialization to increase accuracy in domains poorly represented online is desirable. Two future tasks toward this end are semantic info collection in specialized websites such as Wikipedia, and the addition of contextual features to identify similarity between NEs/CNs (e.g. "Shakespeare wrote *MacBeth*" and "John Bunyan wrote *The Pilgrim's Progress* "; if Shakespeare is `AUTHOR`, then there is knowledge to help with Bunyan). We also hope to implement CACTUS to bootstrap NE annotation for further processing by semi-supervised learners in coreference resolution and referring expression generation.

Acknowledgments

The author wishes to thank Yannick Versley for his advice and support on this project.

References

P. Cimiano 2007. Automatic Acquisition of Ranked Qualia Structures from the Web. *Proc. of ACL-07*.

R. Evans. 2003. A Framework for Named Entity Recognition in the Open Domain. *Proc. of RANLP-2003*.

M. Hearst. 1992. Automatic Acquisition of hyponyms from large text corpora. *Proc. of the 14th conference on Computational Linguistics*.

Z. Kozareva, E. Reiloff, and E. Hovy. 2008. Semantic Class Learning from the Web with Hyponym Pattern Linkage Graphs. *Proc. of ACL-08: HLT*.

M. Steinbach, G. Karypis, and V. Kumar. 2000. A Comparison of Document Clustering Techniques. *KDD Workshop on Text Mining*.

R. Vieira and M. Poesio. 2000. An empirically-based system for processing definite descriptions. *Computational Linguistics, 26(4):539-593*.