Discovering Negative Categories to Improve Semantic Lexicon Induction

- Learning multiple semantic categories simultaneously improves bootstrapping because the categories constrain each other.
- Nevertheless, bootstrappers often begin to acquire instances of new, undesired categories.
- When this behavior is observed, additional "negative" semantic categories can be manually defined to draw away the undesired words and contexts.
- But … manually defining negative categories is a form of human supervision. And it typically requires refinement by iteratively observing the system's behavior.

Discovering Negative Categories by Clustering Drifted Terms

- McIntosh's NEG-FINDER system automatically discovers negative categories by clustering terms that have semantically drifted.
- WMEB detected terms that have drifted from the original semantic category, but they were simply discarded.
- NEG-FINDER caches the drifted terms and then groups similar drifted terms via clustering.
- The goal is to automatically identify groups of drifted terms that represent *cohesive* and *competing* categories.

NEG-FINDER Flowchart

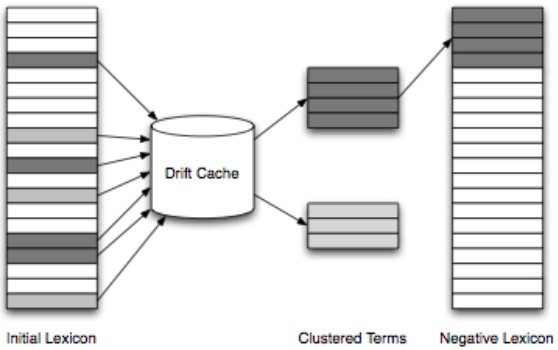

Figure 1: NEG-FINDER: Local negative discovery

Clustering Drifted Terms

- Hierarchical agglomerative clustering is used to group similar terms.
  - initially, each term is assigned to an individual cluster.
  - the clusters are iteratively merged based on a similarity metric, until just one cluster (containing everything) remains.
- The similarity of 2 clusters is the average distributional similarity between all pairs of terms across the clusters.
  - they used the similarity metric for detecting semantic drift: context vectors with t-test weights & weighted Jaccard metric
- Clustering performed when drift cache has 20+ terms.

Identifying Negative Clusters

- Two strategies were tried to identify useful negative category clusters.
- General observation - in agglomerative clustering, the most similar terms are merged first.
- **Maximum Clustering:** identify the *k* most similar terms by exiting the clustering process as soon as a cluster of size *k* is formed.
- **Outlier Clustering:**
  1. identify the drifted term ***t*** that is least similar to the first *n* terms in the lexicon (this has already been pre-computed for drift detection).
  2. exit the clustering process when a cluster of size k is formed that contains term ***t***.

Harvesting Patterns for the Negative Categories

- When a negative cluster is identified, the terms in the cluster become the *seed words* for the new category.
- Patterns must then be extracted for the category.
  - All patterns that co-occur with a negative seed are extracted and ranked with respect to the seeds.
  - The top-scoring *m* patterns are saved for the negative category.
- If a pattern previously used for another category co-occurs with a negative seed, the pattern is discarded.

Local vs. Global Discovery

- Different strategies were also tried for learning negative categories **locally** (based on individual categories) and **globally** (based on the entire lexicon).
- **Local Discovery:** each category has its own *local* drift cache, which is clustered independently from the others.
- **Global Discovery**: all drifted terms are pooled in a single, *global* cache. This may be beneficial if multiple categories drift into the same undesired semantic classes.
- **Mixture Discovery:** *both* local and global drift caches are maintained (i.e., a drifted term goes into both caches). Clustering is performed on both caches.

Manually Defined Negative Categories

Author identified categories by observing the behavior of WMEB

New Category	Drifted from
AMINO ACID	MUTATION
ANIMAL/BODY ORGANISM	CELL/DIS/SIGN DIS

Independent domain expert identified categories

	CATEGORY	SEED TERMS
1	AMINO ACID	arginine cysteine glycine glutamate histamine
	ANIMAL	insect mammal mice mouse rats
	BODY PART	breast eye liver muscle spleen
	ORGANISM	Bartonella Borrelia Cryptosporidium Salmonella toxoplasma
2	AMINO ACID	Asn Gly His Leu Valine
	ANIMAL	animals dogs larvae rabbits rodents
	ORGANISM	Canidia Shigella Scedosporium Salmonella Yersinia
	GENERIC MODIFIERS	decrease effects events increase response acute deep intrauterine postoperative secondary
	PEOPLE	children females men subjects women
	SAMPLE	biopsies CFU sample specimens tissues

Table 3: Manually crafted negative categories

Influence of Manually Defined Negative Categories

First, they measured the impact of the manually defined negative categories as average precision over the 10 target categories:

	1-500	1-1000
WMEB-DRIFT	74.3	68.6
+negative 1	87.7	82.8
+negative 2	83.8	77.8

Table 4: Influence of negative categories

Adding negative categories clearly improves performance!

Comparative Results with Different Drift Cache Strategies

	1-200	201-400	401-600	601-800	801-1000	1-1000
WMEB-DRIFT	79.5	74.8	64.7	61.9	62.1	68.6
NEG-FINDER						
First discovered	79.5	74.3	64.8	67.8	66.6	70.7
Local discovery						
+maximum	79.5	74.8	67.3	69.3	70.5	72.2
+outlier	79.5	73.9	64.8	67.8	71.0	71.5
Global discovery						
+maximum	79.5	73.9	65.7	73.2	72.7	73.4
+outlier	79.5	74.7	65.6	71.4	68.2	72.1
Mixture discovery						
+maximum	79.5	74.7	69.3	73.3	72.8	74.0
+outlier	79.5	75.2	69.7	72.0	69.4	73.2

Table 5: Performance comparison of WMEB-DRIFT and NEG-FINDER

Restarting with the Discovered Negative Categories

Previously, the bootstrapper could only benefit from the discovered categories after they were learned (i.e., after many iterations).

These experiments restart the bootstrapping process, providing it with the automatically discovered negative categories initially.

	1-200	201-400	401-600	601-800	801-1000	1-1000
WMEB-DRIFT						
+negative 1	90.5	87.3	82.0	74.6	79.8	82.8
+negative 2	87.8	82.2	78.7	76.1	63.3	77.8
WMEB-DRIFT						
+restart +local	85.5	82.6	76.5	75.7	68.5	78.4
+restart +global	84.0	83.8	79.1	74.8	69.5	79.7
+restart +mixture	85.2	85.0	82.3	72.5	72.7	81.4

Table 7: Performance of WMEB-DRIFT using negative categories discovered by NEG-FINDER

Combining Manually Defined and Automatically Discovered Negative Categories

Question: Can NEG-FINDER learn useful negative categories beyond what a human expert defines?

The system was initialized with the 10 target categories AND the manually defined negative categories:

	601-800	801-1000	1-1000
WMEB-DRIFT			
+negative 1	74.6	79.8	82.8
NEG-FINDER			
+negative 1 +local	76.4	80.1	83.2
+negative 1 +global	77.5	76.0	82.7
+negative 1 +mixture	76.7	79.9	83.2

Table 8: Performance of NEG-FINDER with manually crafted negative categories

Analysis of Results for Individual Semantic Categories

	ANTI	CELL	DISE	SIGN	TUMR
WMEB-DRIFT	92.9	47.8	49.3	27.9	39.5
+negative 1	91.6	73.1	87.8	76.5	48.7
+negative 2	85.8	68.0	84.2	71.3	16.3
NEG-FINDER					
+mixture	94.9	73.9	56.0	41.0	42.2
+mixture +negative 1	90.8	77.2	87.8	78.2	48.2
WMEB-DRIFT					
+restart +local	89.9	78.8	71.6	73.1	32.2
+restart +global	94.6	79.0	81.9	62.6	35.2
+restart +mixture	92.6	81.1	91.1	63.6	47.5

Table 9: Individual category results (1-1000 terms)

Semi-Automatic Entity Set Refinement

[Vyas and Pantel, NAACL 2009]

- Some search engine companies maintain lists of named entities to improve search results.
- Manually constructing and maintaining named entity lists is expensive, so they are interested in automated **set expansion** techniques.
- Semi-supervised techniques are useful for targeting specific desired categories, with minimal human input.
- But manual refinements and error correction are often needed since these techniques are not perfect and can suffer from semantic drift.

Key Observations

- Ambiguous seed words often lead to semantic drift.

  Roman God Seeds: *Minerva, Neptune, Baccus, Juno, Apollo*

  Expanded List: *Mars, Venus, Moon, Mercury, asteroid, Jupiter, Earth, comet, Sonne, Sun, …*

- Ambiguous entities that share one sense usually do not share other senses that are semantically similar.
  - For example: *Apple* and *Sun* both share the sense COMPANY.
  - But their other senses (FRUIT and CELESTIAL BODY) are semantically different.

Semi-Supervised Refinement

Idea: incorporate *relevance feedback* that asks a human to identify (at most) one error in each iteration.

1. remove items that are distributionally similar to the manually identified errors.
2. dynamically change the feature space based on the error
3. recompute the similarity of each entity with respect to the seeds, and discard those with low similarity

PMI

Pointwise mutual information (PMI) measures the degree to which two words are statistically dependent.

$$PMI(w_1, w_2) = \log_2\left[\frac{P(w_1 \,\&\, w_2)}{P(w_1) * P(w_2)}\right]$$

If PMI = 0, then the words are independent
If PMI > 0, then the words are dependent (i.e., tend to co-occur)

Similarity Method (SIM)

- Create context vectors for each item using a window size of 1, *pointwise mutual information* (*PMI*) weighting, and the *cosine* similarity metric.
- Compute the similarity between each entity in the set and the manually identified error. Remove all entities are are semantically similar using a threshold.
- In the previous example, suppose *Earth* is labeled as an error.
  - *Moon, asteroid, comet, Sun* would be removed ✔
  - *Mars, Venus, Mercury, Jupiter* would also be removed ✗

Feature Modification Method (FMM)

- Idea: identify the features of the erroneous word that represent the unintended semantic class.
- For example, for *Earth*, you may find contextual features such as: *planet, observe, launch, orbit, …*

1. Create a **centroid context vector** for the seeds by taking a weighted average of the seed words' contexts.
2. Identify the features that intersect with the erroneous word and remove them.
3. Rescore all entities with the modified feature vector and discard entities that have a low similarity to the seeds.

Gold Standard Data Sets

- Gold standard evaluation data was created by scraping lists off Wikipedia.
- Lists for 50 semantic categories were generated. On average, each list contained 208 items (minimum of 11, maximum of 1,116).

  Example sets: *classical pianists, Spanish provinces, Texas counties, male Tennis players, first ladies, cocktails, bottled water brands, Archbishops of Canterbury*

- **Note**: these lists are undoubtedly incomplete! And requiring an exact match is very restrictive. So accuracy against these lists will be a lower bound.

Evaluation

- As a baseline, they evaluated the results of simply removing the first incorrect entry for each iteration.
- A distributional set expansion algorithm similar to [Sarmento et al., 2007] was used.
- They performed 1,000 trials with different seed sets. Results are reported for 10 bootstrapping iterations.
- The evaluation metric was R-precision, which is precision after generating *k* items, where *k* is the size of the gold standard set.

R-precision Results

Table 1. R-precision of the three methods with 95% confidence bounds.

ITERATION	BASELINE	SIM	FMM
1	0.219±0.012	**0.234±0.013**	0.220±0.015
2	0.223±0.013	**0.242±0.014**	0.227±0.017
3	0.227±0.013	**0.251±0.015**	0.235±0.019
4	0.232±0.013	**0.26±0.016**	0.252±0.021
5	0.235±0.014	**0.266±0.017**	**0.267±0.022**
6	0.236±0.014	0.269±0.017	**0.282±0.023**
7	0.238±0.014	0.273±0.018	**0.294±0.023**
8	0.24±0.014	0.28±0.018	**0.303±0.024**
9	0.242±0.014	0.285±0.018	**0.315±0.025**
10	0.243±0.014	0.286±0.018	**0.322±0.025**

Conclusions

- Bootstrapped learning of semantic categories often suffers from semantic drift.
- Automatically identifying negative, competing classes can help to draw away incorrect terms and steer the bootstrapping process.
- Distributional semantic similarity methods are useful and easy to apply because they don't require supervision.
- But, semantic lexicon induction is still far from perfect!
- And evaluating the quality of an induced lexicon is challenging, especially with respect to recall.