Minimising semantic drift with Mutual Exclusion Bootstrapping

James R. Curran and **Tara Murphy** and **Bernhard Scholz**
School of Information Technologies
University of Sydney
NSW 2006, Australia
{james,tm,scholz}@it.usyd.edu.au

Abstract

Iterative bootstrapping techniques are commonly used to extract lexical semantic resources from raw text. Their major weakness is that, without costly human intervention, the extracted terms (often rapidly) *drift* from the meaning of the original seed terms.

In this paper we propose *Mutual Exclusion bootstrapping* (MEB) in which multiple semantic classes compete for each extracted term. This significantly reduces the problem of semantic drift by providing boundaries for the semantic classes. We demonstrate the superiority of MEB to standard bootstrapping in extracting named entities from the Google Web 1T 5-grams. Finally, we demonstrate that MEB is a *multi-way cut problem* over semantic classes, terms and contexts.

1 Introduction

Extracting lexical resources from text is a central problem in Natural Language Processing. These resources are the key to overcoming the knowledge bottleneck in tasks ranging from Word Sense Disambiguation to Question Answering.

Template-based approaches have been very successful – they can be implemented efficiently, work on small- and large-scale datasets, and require minimal linguistic pre-processing, so are largely language independent. Template-based relation extraction was pioneered by Hearst (1992), who demonstrate that hyponyms could be extracted using templates like X, . . . , Y and/or other Z where X, . . . , Y are hyponyms of Z. Berland and Charniak (1999) use a similar approach to identify whole-part relations and Caraballo (1999) uses the extracted hyponyms to build a hierarchy. The disadvantage of these fixed templates is that matches are very rare, resulting in low term recall.

Riloff and Shepherd (1997) propose *iterative bootstrapping* where related terms that are frequent neighbours to terms in the semantic class are extracted and Roark and Charniak (1998) improve accuracy by altering the bootstrapping parameters. In *mutual bootstrapping* (Riloff and Jones, 1999), both the terms and the contexts they occur in are extracted. Agichtein and Gravano (2000) and Agichtein et al. (2000) use similar approaches for Information Extraction (IE), such as identifying company headquarters, and Sundaresan and Yi (2000) identify acronyms and their expansions.

Bootstrapping has the advantage that it can identify new templates or *contexts*, which in turn can identify new terms, significantly increasing recall. Unfortunately, adding only a term with a different predominant sense, or a context that weakly constrain the terms, can quickly introduce errors. Therefore, a common theme in the evaluation of bootstrapping is *semantic drift*, when these erroneous terms or contexts infect the semantic class.

We propose a new stricter form of bootstrapping, *Mutual Exclusion Bootstrapping* (MEB), which minimises semantic drift using *mutual exclusion* between semantic classes. Each class is extracted in parallel using separate bootstrapping instances that compete to extract terms and contexts. We add *stop classes* that collect terms known to cause drift in particular semantic classes.

We compare MEB against mutual bootstrapping for extracting BBN named-entity types (Weischedel and Brunstein, 2005) from the 5-grams of the Google Web 1T corpus. We demonstrate that MEB outperforms mutual bootstrapping, can scale to massive datasets, and works well on noisy web text. We also evaluate distributional similarity approaches on this dataset, finding that bootstrapping is faster and more accurate.

Finally, we show that the MEB algorithm is an instance of *multi-way cut*, the generalisation of the

min-cut graph problem. Although multi-way cut is NP-hard, we demonstrate the feasibility of using approximation algorithms to find near optimal partitions of contexts and terms into semantic classes.

2 Mutual and Multi-level Bootstrapping

Riloff and Jones (1999) have proposed *mutual bootstrapping* (MB), where both the terms, and the contexts used to extract terms, are extracted in alternating bootstrap iterations. First a small set of seed words are used to find possible contexts. These contexts are ranked according to

$$\text{score}(c_i) = \frac{\text{seen}(c_i)}{\text{new}(c_i)} \log_2(\text{seen}(c_i)) \qquad (1)$$

where $\text{seen}(c)$ is the number of terms (by type) extracted with context c that are already in the semantic class, and $\text{new}(c)$ is the total number of terms (by type) extracted with context c. MB is designed to balance reliability and productiveness of the context. The highest scoring context is added to the semantic class. The terms that occur in the context are then added to the semantic class.

Riloff and Jones (1999) also introduce *multilevel bootstrapping* to overcome the problem of semantic drift. Rather than adding all of the extracted terms, multi-level bootstrapping only adds the five most reliable terms in each iteration. If a term is extracted by more contexts already in the semantic class then it is more reliable, with a small additional weighting for the score for each context.

We simplify the scoring functions in our implementation, making the scoring symmetrical for terms and contexts. The contexts are ordered by the number of terms in the semantic class they extract (*reliability*). Ties are broken by taking the context that would add the most new terms (*productivity*). In this way, the scoring function prefers precision over recall as much as possible.

Terms are ordered in the same way with respect to contexts. In each iteration a fixed number of contexts and then terms are added to the semantic class, thus we perform multi-level bootstrapping on both the terms and contexts.

3 Mutual Exclusion Bootstrapping

Mutual Exclusion Bootstrapping (MEB) attempts to minimise semantic drift in both the terms and contexts. It does this by extracting multiple semantic classes in parallel, using multiple independent bootstrapping instances, except that a term or

in : Seed word lists $S_k \; \forall$ categories k
in : Raw contexts $\mathscr{C}$ and terms $\mathscr{T}$
in : # terms N_T and contexts N_C per iteration
out: Term T_k and context C_k lists $\forall$ category k
$T_k \longleftarrow S_k \forall$ categories k;
foreach *iteration* **do**
  foreach $c \in \mathscr{C}$ **do**
    count the number of times c occurs with $t \in T_k$;
    discard c if occurs with multiple classes;
  foreach *class k* **do**
    sort set of c by above occurrence counts;
    add top N_C contexts to C_k;
  foreach $t \in \mathscr{T}$ **do**
    count the number of times t occurs with $c \in C_k$;
    discard t if occurs with multiple classes;
  foreach *class k* **do**
    sort set of t by above occurrence counts;
    add top N_T terms to T_k;

Algorithm 1: Mutual Exclusion Bootstrapping

context *must only be used by one* bootstrapping instance. We assume that the terms only have a single sense and that contexts only extract terms with a single sense, that is, the semantic classes are *mutually exclusive* with respect to terms and contexts.

This assumption is far from correct, although for many terms including the named entities we consider here, there is a clearly dominant semantic class. For some pairs of semantic classes, e.g. nationalities and languages, have a significant lexical overlap and are far from mutually exclusive. Interestingly, we see the best results by *artificially forcing these categories apart*. As our experiments show, this enables us to distinguish classes which are quite hard to distinguish otherwise.

The MEB algorithm is shown in Algorithm 1. In each iteration, contexts and then terms are added to each semantic class. If more than one class attempts to extract a context or term then it is eliminated, leading to mutual exclusion between the semantic classes. The terms and contexts are scored and ordered in the same way as our mutual bootstrapping implementation – the only addition in MEB is the parallel mutual exclusion constraint.

The mutual exclusion is very strict and so a large number of terms and contexts are thrown away. This is not a major issue when we are using such a large dataset as the Web 1T corpus, but could be a more significant problem on smaller datasets. It is also more of a problem if there is sig-

nificant lexical overlap between semantic classes.

Notice that the algorithm is sensitive to the order in which contexts and terms are added to the semantic classes, since once they are added to a class they cannot be used elsewhere. For example, if a minority sense of a term is identified by a context first, it may be added to the minority class rather than dominant class for that term. This has the potential to cause drift in the same way as occurs in the original bootstrapping algorithms.

4 Using the Google Web 1T n-grams

Riloff and Jones (1999) used contexts extracted by AutoSlog-TS (Riloff, 1996) from text that had been shallow parsed to identify NPs, VPs and PPs. This means a POS tagger and chunker must be available in the target language, making their approach language dependent. In our experiments, we wanted to take a completely language independent approach where possible. We also wanted to demonstrate that MEB could scale efficiently to extremely large datasets, because these datasets provide the levels of redundancy needed to overcome the sparseness of the extracted contexts.

Google has recently released the Web 1T corpus (Brants and Franz, 2006), which consists of unigram to 5-gram counts calculated over 1 trillion words of web page text collected in January 2006. The text was tokenised following the Penn Treebank tokenisation, except that words are usually split on hyphens, and dates, email addresses and URLs are kept as single tokens. The sentence boundaries are marked with two special tokens <S> and </S>. The individual terms in the n-grams occurred at least 200 times otherwise they were replaced with the special token <UNK>. The n-grams themselves must appear at least 40 times to be included in the Web 1T corpus.

We use the 5-grams from the Web 1T corpus as our raw text, such that the middle token is the *term* and the two tokens on either side form the *context*. The advantage of this context definition is that it is quite language independent. The disadvantage is that we can only extract terms consisting of a single word and the contexts are noisier than those extracted from the shallow parsed text.

We filter out 5-grams in several ways. We remove all 5-grams where the middle token is not title case because we are only extracting proper noun named-entity types. We also remove all contexts that include numbers. Finally, we eliminate

TYPE	COUNT
Number of terms	694 047
Number of contexts	10 597 784
Number of unique instances	42 807 058
Number of instances	21 308 744 742

Table 1: Filtered Web 1T dataset statistics.

contexts that only appear with one term and thus terms that only appear with one context, since they cannot be reached by the bootstrapping algorithm. The size of the resulting dataset is shown in Table 1. We have reduced the 1 trillion n-grams down significantly with filtering, so we only using 2% of the data by type and 3.6% of the data by token. However, the number of terms and contexts by type is still extremely large. The dataset is 666MB on disk which all needs to be loaded into memory at once.

5 Implementation

The MEB implementation has been optimised to be as time and space efficient as possible. Each unique term that appears with a context requires only 4 bytes of storage, which means the program requires around 1GB of RAM to run. The terms and contexts that co-occur are completely cross-indexed which makes updating the term and context extraction counts very efficient. Finally, the mutual exclusion property means that the term and context sets for each semantic class can be represented implicitly using flags, so the many set membership tests are also extremely fast. The bootstrapping experiments described here take only minutes to run and much of that time is spent loading the data into memory.

6 Selecting semantic classes

In these experiments, we wanted to extract semantic classes corresponding to proper-noun named entities only. We based our semantic classes on the 29 entity types used to annotate the BBN Pronoun Coreference and Entity Type Corpus (Weischedel and Brunstein, 2005) distributed by the LDC. The BBN corpus includes detailed entity annotation guidelines which helped with the evaluation process described below.

We ignored many entity types that did not primarily involve proper nouns, including DESCRIPTION types, CHEMICALS and SUBSTANCES, TIMES, MONETARY amounts and QUANTITIES

LABEL	DESCRIPTION
FEM	Person: female first name
	Mary Patricia Linda Barbara Elizabeth
MALE	Person: male first name
	James John Robert Michael William
LAST	Person: last name
	Smith Johnson Williams Jones Brown
TTL	Honorific title
	President Dr Lord Miss Major
NORP	Nationality, Religion, Political (adjectival)
	American European Indian Republican Christian
FAC	Facility: names of man-made structures
	Broadway Legoland Capitol Boomers SeaWorld
ORG	Organisation: e.g. companies, governmental
	Intel Microsoft Sony IBM Ford
GPE	Geo-political entity
	Canada America China Washington London
LOC	Locations other than GPEs
	Europe Africa Asia Pacific Earth
DAT	Reference to a date or period
	January May Friday Monday Easter
LANG	Any named language
	English Chinese Arabic Spanish Hebrew

Table 2: The semantic classes

etc. We ignored entity types that were nonsensical without multi-word terms including WORKS OF ART, LAWS and EVENTS. We were also interested in more fine-grained distinctions for the PERSON type, which we split into MALE and FEMALE first names, and LAST names. This resulted in the semantic classes listed in Table 2, which we used for all experiments unless otherwise noted.

We found that the mutual exclusion bootstrapping was most accurate when additional *stop classes* (like stop-lists) were included to help bound the semantic classes. These classes were selected based on observed semantic drift in specific categories. For instance, the JEWEL class was added to stop FEMALE from drifting when it reached names like *Ruby*. The stop classes we included were ADDRESS, BODY PART, CHEMICAL, COLOUR, DRINK, FOOD, JEWELS and WEB terms.

7 Selecting seed lists

To create seed lists we collected named entity lists from a variety of sources. The basis for each collection was the list of most frequent entities for that category from the BBN corpus. This was supplemented with external sources e.g. lists of Fortune 500 companies for ORG; the largest cities from Wikipedia for GPE; and names from the US Census for FEM, MALE and LAST.

We then extracted the frequency of each term in these lists from the Web 1T corpus. Seed lists were created using the top 50, 20, 10 and 5 most frequent single-word terms from these lists. Words

that occurred in multiple categories (e.g. *French* in NORP and LANG) were assigned to one category or the other, to ensure each seed list was mutually exclusive. We also created seed lists for the stop classes based on our initial experiments.

8 Evaluation

Our evaluation process involved manually inspecting each extracted term and judging whether it was a member of the semantic class, following Riloff and Jones (1999). To make this more efficient, we stored a cache of previous evaluator decisions for each class so that once a decision had been made for a particular term in a particular class it would be made automatically in future instances.

Although the seed lists were mutually exclusive, for the purposes of evaluation ambiguous words such as *French* were counted as correct if they appeared in either valid category (NORP or LANG). This means that MEB has a minor disadvantage in the evaluation because terms may belong to multiple classes with other approaches.

Evaluation was made more difficult by the fact that we had only single word terms and yet many company names, facility names, etc. are typically multi-word terms. When the single word was an clearly part of a multi-word term we counted it as correct (eg. *Coast* as a LOC). However, if the word was not strongly correlated with the semantic class (e.g. *The* or *Next*) it was not counted as correct. Obvious mis-spellings of words (eg. *Januray*) were also counted as correct. The extracted terms that were unrecognised by the evaluator were checked using Wikipedia and Google.

To compare approaches and parameters we used accuracy at n – the percentage of correct terms in the top n ranked terms for a given category. This evaluation gives a realistic measure of the practical usefulness of the results since the ranked list of bootstrapped terms will be used directly in downstream NLP components. For many experiments this is averaged over the semantic classes ($Av(n)$). We also we calculated the *inverse rank* (InvR) – the sum of the inverse rank of all correct terms. InvR provides a summary of both the number of correct terms and their ranking in the list.

For comparing the accuracy of different approaches and parameter settings, we manually evaluated all 11 semantic categories down to $n = 50$, which was enough to discriminate between most results. For the final results we evaluated

TYPE	nS	nT	nC	Av(10)	Av(50)
MB	5	5	5	55	21
MB	5	5	10	58	28
MB	5	5	100	79	59
MB	5	5	200	80	**68**
MB	5	5	300	**84**	66
MEB-NS	5	5	5	84	67
MEB-NS	5	5	10	**89**	68
MEB	5	5	5	86	67
MEB	**5**	**5**	**10**	**90**	**78**

Table 3: Results comparing approaches.

down to the point where MEB was still producing good results, with a maximum depth of $n = 400$.

9 Results

There are three main parameters to vary in the MEB algorithm – the number of terms in each seed list (nS), and the number of terms (nT) and contexts (nC) to add in each iteration. Our default parameters are 5 for nS, nT and nC. For the experiments below we compare the average semantic class accuracy at 10 and 50 terms.

Table 3 summarises the comparison of mutual bootstrapping (MB) including multi-level bootstrapping, with both mutual exclusion bootstrapping with (MEB) and without (MEB-NS) stop classes. The main results are that MEB significantly outperforms MB and that stop classes play a significant role in bounding semantic classes reducing semantic drift. An interesting new result is that mutual bootstrapping performs badly when few contexts are added, but performs much better when many contexts, e.g. 200, were added in each iteration.

We intend to do further analysis on the many contexts result for MB, but it appears that since MB is very susceptible to semantic drift using many pieces of contextual evidence extracted using the initial seed words is crucial for good performance.

9.1 Parameter settings

For the remainder of the experiments we use MEB with stop classes. In Table 4, we see the results we would expect for increasing the number of seed words for each semantic class. The accuracy is highest when we use 50 seed terms, although collecting 50 seed terms this would take significantly effort than the default of 5. Of course, we can use MEB to extract terms and then manually correct to create larger seed sets quickly.

nS	nT	nC	Av(10)	Av(50)
2	5	5	65	50
5	5	5	86	67
10	5	5	94	67
20	5	5	95	84
50	**5**	**5**	**95**	**91**

Table 4: Results for different seed list size.

nS	nT	nC	Av(10)	Av(50)
5	1	5	86	63
5	2	5	86	69
5	5	5	86	67
5	**10**	**5**	**84**	**70**

Table 5: Results for terms added per iteration.

nS	nT	nC	Av(10)	Av(50)
5	5	1	76	64
5	5	2	77	59
5	5	5	86	67
5	**5**	**10**	**90**	**78**
5	5	15	90	74
5	5	20	90	72
5	5	100	90	62

Table 6: Results for contexts added per iteration.

In Tables 5 and 6 the number of terms or contexts parameters are varied. Adding 10 terms per iteration is more effective than the default of 5, and both outperform the more conservative strategy of only adding one term per iteration. Adding 10 contexts per iteration is also more effective than the one context per iteration used by Riloff and Jones (1999). However, adding 10 terms and 10 contexts per iteration is not as accurate, so the 5–5–10 settings are used for the remaining experiments unless noted.

We investigate the robustness of the results to the quality of the seed sets in Table 7. To experiment with this we created three sets of seed sets with HIGH, MID, and LOW frequency terms as calculated from the Web 1T unigram counts. The HIGH counts are the default set used for the other experiments. We also created a set that was manually selected to best represent the semantic class. This significantly outperformed frequency-based seed sets demonstrating that selecting good seed terms is crucial to high accuracy.

TYPE	nS	nT	nC	Av(10)	Av(50)
HIGH	5	5	5	86	67
MID	5	5	5	90	70
LOW	5	5	5	88	70
MANUAL	**5**	**5**	**5**	**92**	**79**
MANUAL	5	5	10	92	75

Table 7: Results for different seed lists.

9.2 Distributional approaches

Another standard approach to extracting lexical semantic resources is *distributional similarity*, based on the distributional hypothesis that similar terms appear in similar contexts. In distributional approaches, all of the contextual information is summarised in weighted context vectors which are compared using measures of similarity in vector space. We wanted to compare these approaches since this hasn't been done previously using exactly the same data. Hearst and Grefenstette (1992) experimented with combining template methods with the Grefenstette (1994) distributional approach.

We use the distributional similarity approach presented in Curran (2004). The same filtered set of Web 1T 5-grams is converted into context vectors, which corresponds to a window-based context, and the standard t-test weighting and Jaccard measure functions were used (Curran, 2004). Synonym lists of length 200 were generated for head terms that occurred with frequency ≥ 1000.

To map from head terms to semantic classes, we experimented with three methods used in the similar task of supersense tagging (Curran, 2005), where each term from the seed list can vote for synonyms for that class. There are three weighting schemes: with SET each synonym is equally weighted; with SCORE the distributional similarity score weights each synonym; and with RANK the inverse rank weights each synonym. The collected synonyms for each semantic class are then sorted by weighted votes and the top *n* selected.

The results are shown in Table 8. The SET method performs significantly worse than SCORE and RANK, but none of the methods are competitive with the best MEB system on the top 50 extracted terms.

9.3 Semantic Classes

We evaluate the performance of individual semantic classes in Table 9. The evaluation includes accuracy at depths of up to 400 terms per semantic

TYPE	Av(10)	Av(50)
SET	67	58
SCORE	86	70
RANK	**88**	**72**

Table 8: Results for distributional similarity.

class. We stopped evaluating each semantic class after MEB stopped finding new terms in that class. We have also calculated the inverse rank for the individual classes. The results show that some semantic classes are considerably more difficult than others, showing drift after far fewer iterations than other classes. This evaluation is harsh on classes with fewer than 400 terms, e.g. honorific titles. For the four most reliable classes we also manually checked down to 750 terms, where MEB still performed extremely well with FEM 63%, MALE 88%, LAST 95% and GPE 96%.

Some pairs of semantic classes, especially FAC and ORG, and LOC and GPE, require much more subtle semantic distinctions than previous bootstrapping evaluations. The evaluators had considerable difficulty distinguishing between a facility and an organisation based on single-word terms. We merged these problematic categories into more general categories to see if this improved the results. We merged FAC and ORG to form the FOG class, and LOC and GPE to form PLACE.

The two merged classes appear in Table 9. Merging improved the performance dramatically with FOG and PLACE 95% and 100% accurate (respectively) at 400 extracted terms. However, we noticed a slight decrease in performance for the NORP and DATE which demonstrates the boundary interactions that can occur with MEB.

9.4 Resource coverage

There is a suspicion that automatically extracted lexical semantic resources tend to contain the same terms that are available in existing manually created resources. By using existing resources to speed up the manual evaluation process we were able to identify interesting terms that would typically not be contained in existing resources, e.g.:

- foreign translation terms. MEB found non-English months including *Oktober* and *Chwefror* (February in Welsh);
- names missing from the US census lists, which covered names down to 0.001% of the population, e.g. *Uday* and *Igor*;
- programming languages, e.g. *Python*;

n	FEM	MALE	LAST	TTL	NORP	FAC	ORG	GPE	LOC	DAT	LANG	FOG	PLACE
10	100	100	100	100	90	70	50	100	80	100	100	100	100
20	100	100	100	100	90	60	35	100	80	100	100	100	100
50	100	100	100	66	90	48	16	100	64	78	100	100	100
100	99	100	100	51	67	32	8	100	39	56	85	100	100
150	99	100	100	38	61	23	5	99	31	65	66	100	100
200	95	100	100	31	57	18	-	99	27	60	63	99	100
250	91	100	100	27	49	-	-	98	22	-	58	99	100
300	91	97	100	-	42	-	-	98	-	-	58	97	100
350	88	94	100	-	-	-	-	98	-	-	53	95	100
400	87	94	99	-	-	-	-	98	-	-	47	95	100
InvR	5.92	6.27	6.30	4.35	4.95	3.09	2.39	6.26	3.89	4.58	5.38	6.50	6.57

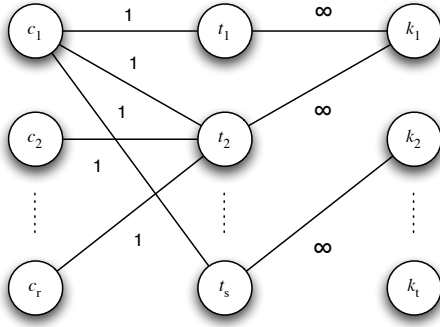Table 9: Results for our 11 original categories. The maximum inverse rank possible is 6.57.

Figure 1: Multi-partite graph for MEB.

- many rare languages - Aboriginal and African tribal languages, and *Klingon*!

10   MEB as Multi-way cut

Mutual exclusion bootstrapping can be posed as a multi-partite graph partitioning problem where semantic classes, terms and contexts are nodes and membership and cooccurrence for the edges. Theoretically, this approach allows terms and contexts to be optimally separated into semantic classes.

Figure 1 shows the multi-partite graph. Given the set of contexts C and the set of words T, the *word-context relation* $R \subseteq C \times T$ denotes pairs (c,t) for which term t appears in context c.

A word/context u is *connected* to word/context v, if there exists a path from u to v in graph $G\langle T \cup C, R\rangle$. A *seed semantic class* $\Gamma : K \to 2^T$ is a partial mapping from semantic class to a subset of terms. A word/context $u \in T \cup C$ is *associated* with semantic class k, if there exists term $t \in \Gamma(k)$ that is connected to u.

We seek for a *word/context labelling* $\Lambda : T \cup C \to K$ such that a minimal number of pairs are to be removed from R to make the classification unique, i.e. there exists neither a context nor a term for which we have multiple semantic class associations. Intuitively this corresponds to splitting the terms and contexts into mutually exclusive semantic classes by ignoring the minimum number of occurrences of terms with contexts.

MEB is reducible to a multi-way cut. For the reduction we construct a multi-partite graph as shown in Fig. 1. The first and second node layers represent the term-context relationship and the second and third node layers represent the seed semantic class mapping. The semantic classes are the multi-way cut terminal vertices and the multi-way cut of the multi-partite graph is optimal MEB.

10.1   Multi-Way Cut

Given a graph $G\langle U, E\rangle$ and a set $T \subseteq V$ of k terminal vertices, a *multi-way cut* (also known as k way-cut) (cf. (Bachour et al., 2005)) is a set $C \subseteq E$ of edges such that in $G'(V, E - C)$, no path exists between any two nodes of T, i.e., the terminal vertices become disconnected from each other. The multi-way cut problem seeks for a cut such that $|C|$ becomes minimal. The weighted multi-way cut problem seeks a cut C such that $\sum_{e \in C} w(e)$ is minimal where $w(e)$ is the weight of edge e.

For $k = 2$, the problem is reduced to the $s - t$ min-cut problem introduced by Ford and Fulkerson (Calinescu et al., 1998) that can be solved via its dual problem – the *max-flow* problem in polynomial time. Unfortunately, for undirected graphs the multi-way cut problem is NP-hard for $k \geq 3$. Dahlhaus et al. (1994) give a simple combinatorial isolation heuristic that approximates a solution with error bounded by $2 - \frac{2}{k}$ to the optimal solu-

tion. In this algorithm $k-1$ terminals are chosen and a $s-t$ min-cut separates the selected terminal from the other terminals. The union of these cuts give the approximation of the multi-cut.

The approximation algorithm in Dahlhaus et al. (1994) has the worst approximation bound but the the best worst-case complexity class. A deterministic algorithm for max-flow (Goldberg and Tarjan, 1988) results in a worst-case complexity of $\tilde{\mathscr{O}}(k \cdot m \cdot n)$ where n is the number of vertices and m the number of edges in graph G. A probabilistic algorithm for $s-t$ min-cut even improves the worst-case complexity to $\tilde{\mathscr{O}}(k \cdot m)$.

We have completed a practical implementation of $s-t$ min-cut MEB that can run on datasets of around 10 000 terms and the results are very promising. We believe that posing MEB as an optimal graph partitioning problem has great potential to improve the quality of our results further.

11   Conclusion

The MEB algorithm deserves further study as do the many contexts results for the existing mutual bootstrapping algorithm. For instance, the results may be sensitive to the ordering of semantic classes, and to the ranking of terms and contexts. Also, the results are dependent on the ambiguity and representativeness of the initial seed list for both semantic classes and stop lists. Since evaluation is very time consuming we haven't explored these problems yet. We would also like to investigate whether the mutual exclusion can be relaxed to some degree without losing the significant gains in performance. Finally, we hope to apply MEB to other tasks (e.g. common nouns) and languages.

In this paper we have proposed *mutual exclusion bootstrapping* (MEB), based on the mutual bootstrapping algorithm proposed by Riloff and Jones (1999), which attempts to overcome the semantic drift common to iterative bootstrapping techniques. MEB extracts terms and contexts for multiple semantic classes in parallel, imposing a strict constraint that the classes must be mutually exclusive with respect to both terms and contexts.

Although this assumption is false for many pairs of semantic classes, it still significantly improves the quality of the extracted terms. We have evaluated our approach on a wide range of proper-noun named-entity classes using the massive Google Web 1T dataset, also demonstrating that MEB scales efficiently. We have also experi-

mented with a wide range of parameters that affect bootstrapping accuracy. The result is an algorithm that can extract large lexical semantic resources with a high degree of reliability. Finally, we have demonstrated that MEB can be posed as the multi-way cut optimisation problem from graph theory, solvable using approximation algorithms.

References

Eugene Agichtein, Eleazar Eskin, and Luis Gravano. 2000. Combining strategies for extracting relations from text collections. Technical Report CUCS-006-00, Department of Computer Science, Columbia University, New York.

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM Conference on Digital Libraries*, pages 85–94. San Antonio, TX USA.

Khaled Bachour, Eda Baykan, Wojciech Galuba, and Ali Salehi. 2005. Citation network partitioning. Technical report, Ecole Polytechnique Fédérale de Lausanne.

Matthew Berland and Eugene Charniak. 1999. Finding parts in very large corpora. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 57–64. College Park, MD USA.

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram version 1. Technical Report LDC2006T13, Linguistic Data Consortium.

Gruia Calinescu, Howard Karloff, and Yuval Rabani. 1998. An improved approximation algorithm for multiway cut. In *STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 48–52. ACM Press, New York, NY, USA.

Sharon A. Caraballo. 1999. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 120–126. College Park, MD USA.

James R. Curran. 2004. *From Distributional to Semantic Similarity*. Ph.D. thesis, University of Edinburgh, Edinburgh, UK.

James R. Curran. 2005. Supersense tagging of unknown nouns using semantic similarity. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 26–33. Ann Arbor, MI USA.

Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, P. D. Seymour, and Mihalis Yannakakis. 1994. The complexity of multiterminal cuts. *SIAM J. Comput.*, 23(4):864–894.

A.V. Goldberg and R.E. Tarjan. 1988. A new approach to the maximum flow problem. *J. of the ACM*, 35(4):921–940.

Gregory Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Boston.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th international conference on Computational Linguistics*, pages 539–545. Nantes, France.

Marti A. Hearst and Gregory Grefenstette. 1992. A method for refining automatically-discovered lexical relations: Combining weak techniques for stronger results. In *Statistically-Based Natural Language Programming Techniques: Papers from the AAAI Workshop*, Technical Report WS-92-01, pages 72–80. AAAI Press, Menlo Park.

Ellen Riloff. 1996. Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1044–1049.

Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 474–479. Orlando, FL USA.

Ellen Riloff and Jessica Shepherd. 1997. A corpus-based approach for building semantic lexicons. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 117–124. Providence.

Brian Roark and Eugene Charniak. 1998. Noun-phrase co-occurrence statistic for semi-automatic semantic lexicon construction. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th annual meeting of the Association for Computational Linguistics*, pages 1110–1116. Montréal, Québec, Canada.

Neel Sundaresan and Jeonghee Yi. 2000. Mining the web for relations. In *Proceedings of the 9th International World Wide Web Conference*. Amsterdam, Netherlands.

Ralph Weischedel and Ada Brunstein. 2005. BBN pronoun coreference and entity type corpus. Technical Report LDC2005T33, Linguistic Data Consortium.