Seeded Discovery of Base Relations in Large Corpora

Nicholas Andrews
BBN Technologies*
noa@bbn.com

Naren Ramakrishnan
Virginia Tech
naren@cs.vt.edu

Abstract

Relationship discovery is the task of identifying salient relationships between named entities in text. We propose novel approaches for two sub-tasks of the problem: identifying the entities of interest, and partitioning and describing the relations based on their semantics. In particular, we show that term frequency patterns can be used effectively instead of supervised NER, and that the p-median clustering objective function naturally uncovers relation exemplars appropriate for describing the partitioning. Furthermore, we introduce a novel application of relationship discovery: the unsupervised identification of protein-protein interaction phrases.

1 Introduction

Relationship extraction (RE) is the task of extracting named relationships between entities in text given some information about the relationships of interest. Relationship discovery (RD), on the other hand, is the task of finding which relations exist in a corpus without any prior knowledge. The discovered relationships can then be used to bootstrap RE, which is why RD has also been called unsupervised relation extraction (Rosenfeld and Feldman, 2006). RD generally involves three sub-tasks: entities of interest are either supplied or recognized in the corpus; second, of all phrases in which entities co-occur, those which express a relation are picked out; finally, these relationship phrases are partitioned based on their semantics and described. This work considers only binary relations (those between exactly two entities).

Finding entities of interest has involved either named entity recognition (NER) or general noun phrase (NP) chunking, to create the initial pool of candidate entities. In Section 2, we describe a corpus statistics approach, previously applied for web mining (Davidov and Rappoport, 2006), which we extend for relation discovery. Unlike supervised machine learning methods, this algorithm does not need training, is computationally efficient, and requires as input only the raw corpus and a small set of seed entities (as few as two). The result is a set of entities likely to be related to the seeds.

An assumption commonly held in RD work is that frequently co-occurring entity tuples are likely to stand in some fixed relation (Hasegawa et al., 2004; Shinyama and Sekine, 2006; Rosenfeld and Feldman, 2006; Rosenfeld and Feldman, 2007). Tuples which share similar contexts (the exact definition of context varies) are then grouped together in clusters of relations using variants of hierarchical agglomerate clustering (HAC). However, to our knowledge, no prior work has satisfactorily addressed the problem of describing the resulting clusters. In Section 3, we propose an approach which incorporates this requirement directly into the clustering objective: to find relation clusters which are well-described by a single exemplar.

In Section 4, we apply RD to recognize protein-protein interaction (PPI) sentences, using proteins as seeds for the entity discovery phase. We compare our results against special-purpose methods in terms of precision and recall on standard data sets.

The remainder of this paper is outlined below: Section 2 describes how a small number of input words (the entities of interest) are used as seeds for unsupervised entity discovery. Section 3 describes how discovered entities are used to discover relationships. Section 4 describes evaluation methodology and results. Section 5 describes related work. Section 6 concludes and discusses

[*] This work was conducted while author was at Virginia Tech.

directions for future work.

2 Entity discovery

For a corpus C, each sentence $s \in C$ with words $s = (w_1, w_2, ..., w_n)$, is mapped to the sequence $s' = f(s)$. The function f maps each word $w \in s$ to a symbol based on its frequency in C as follows:

$$f(w) = \begin{cases} S & \text{if } w \text{ is a seed word} \\ H & \text{otherwise if } w \text{ is a frequent word} \\ X & \text{otherwise} \end{cases}$$

For example, the sentence:

> A and B are usually mediated by an overproduced C.

might be mapped to the sequence $(S, H, X, H, H, X, H, H, X, X)$, which we will write as $SHXHHXHHXX$ for brevity. In this case, A is a seed term, while B and C are not. The underlying assumption is that content words can be distinguished from other words based on their frequency in the corpus.

2.1 Pattern induction

In the example sentence, 'A and B are usually mediated by an overproduced C', 'and' is a good indicator that A,B share some aspect of their semantics; in this case, that they are both mediated by an overproduced C, and are therefore also likely to belong to same family or type of entities. The indicators 'and' and 'or' have together been used to discover word categories in lexical acquisition (Dorow et al., 2005). However, there can be many other such indicators, many discourse or corpus specific. To discover them, we use a slightly modified version of the method presented in (Davidov and Rappoport, 2006). In particular, in this work we consider named entities of arbitrary length (i.e., longer than a single token).

The corpus is searched for all instances of the frequency pattern $H_1\mathbf{S_1}H_2\mathbf{S_2}H_3$, for seed words S_1, S_2, and pattern (H_1, H_2, H_3). Of all these pattern instances, we keep those which also appear as $H_1\mathbf{S_2}H_2\mathbf{S_1}H_3$. If seed words appear on either side of the pattern, it is a good indication that the symmetric pattern expresses some sort of a conjunction, often domain specific. This procedure is repeated for variations of $HSHSH$ with the goal of capturing different forms of speech; for example, $HSHSH$ will capture '; A , B and', while $HSHHSH$ will capture '; A but not B ,' and so on. We enforce that frequent words appear before and after (i.e., surround) the two seed words to ensure they are standalone entities, and not part of a longer noun phrase. For example, the phrase 'IFN-gamma mRNA and IL-6 are' maps to $XXHSH$, and therefore 'mRNA' would (correctly) not be added to the entity pool.

New entities are added to the initial set of seed by matching symmetric patterns. If a seed word S is found to occur with an infrequent word X in any discovered symmetric pattern (as $H\mathbf{S}HXH$ or $HXH\mathbf{S}H$), then we add X to the pool of entities. This process can be bootstrapped as needed.

2.2 Chunking

In Section 3.1, sentences in which entities co-occur are clustered based on a measure of pairwise similarity. The features used in this similarity calculation are based on the surrounding or connecting words in the sentence in which entities co-occur. To ensure the context is not polluted with words which actually belong the entity NP (such as 'IFN-gamma *mRNA*') rather than the context, we use frequency patterns to search the corpus for common NP chunks.

In each sentence in which entities occur, we form a candidate chunk by matching the regular expression HX^*SX^*H, which returns all content-words X bracketing the entity S. Of all candidate chunks, we keep those which occur frequently enough to significantly affect the similarity calculations. The remaining chunks are pruned based on the entropy of the words appearing immediately before and after the chunk in the corpus; if a given chunk appears in a variety of contexts, it is more likely to express a meaningful collocation (Shimohata et al., 1997). Therefore, as an efficient filter on the candidate chunks, we discard those which tend to occur in the same contexts (where the context is $H...H$).

3 Identifying relation phrases

Once the pool of entities has been recognized in the corpus, those which frequently co-occur are taken as likely to stand in a relation. Order matters in that $S_1..S_2$ is considered a different entity co-occurrence (and therefore potential relation) than $S_2..S_1$. The effect of the co-occurance threshold on the resulting relations is investigated in Section 4.

3.1 Clustering relation phrases

Partitioning the candidate relationships serves to identify groups of differently expressed relationships of similar semantics. The resulting clusters should cover the most important relations in a corpus between the entities of interest. The phrases in each cluster are expected to capture most syntactic variation in the expression of a given relationship. Therefore, the largest clusters are well suited as positive examples for training a relationship extractor (Rosenfeld and Feldman, 2006).

We take the context of a co-occurring tuple to be the terms connecting the two entities within the sentence in which they appear, and call the connecting terms a relation phrase (RP). Each RP is treated separately in the similarity calculations and the clustering. Relations are modeled using a vector space model. Each relation is treated as a vector of term frequencies (tf) weighted by tf × idf. RPs are preprocessed by filtering stopwords[1]. However, we do not stem the remaining words, as suffixes can be highly discriminative in determining the semantics of a relation (e.g., 'production' vs 'produced'). After normalizing vectors to unit length, we compute a similarity matrix by computing the dot product between the vectors for each distinct RP pair. The similarity matrix is then used as input for the clustering.

3.2 p-Median clustering

Prior approaches to relationship discovery have used HAC to identify relation clusters. HAC is attractive in unsupervised applications since the number of clusters is not required *a priori*, but can be determined from the resulting dendogram. On the other hand, a typical HAC implementation runs in $\Theta(N^2 \log(N))$, which can be prohibitive on larger data sets[2].

A further feature of HAC, and many other partitional clustering algorithms such as k-means and spectral cuts, is that the resulting clusters are not necessarily well-described by single instance. Relations, however, typically have a base or root form which would be desirable to uncover to describe the relation clusters. For example, in the following RPs:

induced transient increases in
induced biphasic increases in
induced an increase in
induced an increase in both
induced a further increase in

the phrase 'induced an increase in' is well suited as a base form of the relation and a descriptor for the cluster. The p-median clustering objective is to find p clusters which are well-described by a single exemplar. Formally, given an $N \times N$ similarity matrix, the goal is to select p columns such that the sum of the maximum values within each row of the selected columns are maximized.

Note that an exemplar can also be chosen *a posteriori* using some heuristic; for example, the most frequently occuring instance in a cluster can be taken as the exemplar. However, the p-median clustering objective is robust, and ensures that only those clusters which are well desribed by a single exemplar appear in the resulting partition of the relations. This means that the optimal number of clusters for the p-median clustering objective in a given data set will usually be quite different (usually higher) than the optimal number of groups according to the HAC, k-means, or normalized cut objectives.

Affinity propagation (AP) is the most efficient approximation for the p-median problem that we are aware of, which also has the property of not requiring the number of clusters as an explicit input (Frey and Dueck, 2007). Runtime is linear in the number of similarities, which in the worst case is N^2 (for N relations), but in practice many relations share no words in common, and therefore do not need to have their similarity considered in the clustering.

AP is an iterative message-passing procedure in which the objects being clustered compete to serve as cluster exemplars by exchanging two types of messages. The responsibility $r(x, m)$, sent from object $x \in \mathcal{X}$ (for set $\mathcal{X}$ of objects to be clustered) to candidate exemplar $m \in \mathcal{X}$, denotes how well-suited m is of being the exemplar for x by considering all other potential exemplars m' of x:

$$s(x, m) - \max_{m' \in \mathcal{X}, m' \neq m} a(x, m') + s(x, m')$$

where $s(x, m)$ is the similarity between x, m. The availability $a(x, m)$ of each object $x \in \mathcal{X}$ is initially set to zero. Availabilities, sent from candidate exemplar m to object x, increase as evidence for m to serve as the exemplar for x increases:

$$\min \left\{ 0, r(m, m) + \sum_{x' \in \mathcal{X}, x' \notin \{x, m\}} \max\{0, r(x', m)\} \right\}$$

Each object to be clustered is assigned an initial preference of becoming a cluster exemplar. If there are no *a priori* preferences for cluster exemplars, the preferences are set to the median similarity (which can be thought of as the 'knee' of the objective function graph vs. number of clusters), and exemplars emerge from the message passing procedure. However, shorter RP are more likely to contain base

[1] We use the English stopword list from the Snowball project, available at `http://snowball.tartarus.org/`

[2] An optimization to $\Theta(N^2)$ is possible for single-linkage HAC.

forms of relations (because longer phrases likely contain additional words specific to the sentence). Therefore, we include a slight scaling factor in the preferences, which assigns shorter RP higher initial values (up to $1.5\times$ the median similarity).

3.3 Pruning clusters

After clustering relation phrases with AP, we prune the resulting partition by evaluating the number of different relation instances appearing in each cluster, as well as the entities involved. In our experiments, we discard all clusters smaller than a certain threshold, since we ultimately wish to use the clustering to train RE, and small clusters do not provide enough positive examples for training (we investigate the effect of this threshold in Section 4.2). We further assume that for a relationship to be useful, a number of different entities should stand in this relation. In particular, we inspect the set of left and right arguments in the cluster, which (in English) usually correspond to the subject and object of the sentence. If a single entity constitutes more than two thirds ($\frac{2}{3}$) of the left or right arguments of a cluster, then this cluster is discarded from the results. Our assumption is that these clusters describe relations too specific to be useful.

4 Evaluation

RD systems are usually evaluated based on their results for a particular task such as RE (Rosenfeld and Feldman, 2006), or by a manual inspection of their results (Davidov et al., 2007; Rosenfeld and Feldman, 2007; Hasegawa et al., 2004), but we are not aware of any which examines the effects of parameters on performance exhaustively. In this section we test several hypotheses of RD using data sets which are already labeled for sentences which contain entities of a particular type and in a fixed relation of some kind. In particular, we adapt the output of the discovery phase to identify phrases which express PPIs. While this task is traditionally performed using supervised algorithms such as support vector machines (Erkan et al., 2007), we show that RD is capable of achieving similar levels of precision without any manually annotated training data.

4.1 Method

We construct a corpus of 87300 abstracts by querying the PubMed database with the proteins shown in Table 1. The 60 most frequent words are considered definate non-entities; all remaining words are candidate entities. This corpus serves as input for the relationship discovery. As seeds, we use the same 25 proteins used to query the database. Since all seeds are proteins, we expect the entities discovered to be proteins. The pattern induction found roughly 200 symmetric extraction patterns, which yield 4402 unique entities after 1 pass through the corpus. Depending on the frequency of the seeds in the corpus, more passes through the corpus might be needed (boostrapping with the discovered entities after each pass). We retain all chunks that appear at least 10 times in the corpus, yielding 3282 additional entities after entropy pruning.

Table 1: Proteins queried to create the evaluation corpus.

Seed entities (proteins)				
c-cbl	AmpC	CD18	CD54	CD5
CD59	CK	c-myc	CNP	DM
EBNA	GSH	IL-8	IL-1beta	JNK1
p38	PABP	PCNA	PP1	PP2a
PPAR	PSM	TAT	TNF-alpha	TPO

A PPI denotes a broad class of bio-medical relationships between two proteins. One example of an interaction is where the two proteins bind together to form a structural complex of cellular machinery such as signal transduction machinery. A second example is when one protein binds upstream of the DNA sequence encoding a gene which encodes the second protein. A final example is when proteins serve as enzymes catalyzing successive steps of a biochemical reaction. More categories of interactions are continually being catalogued and hence unsupervised identification of PPIs is important in biomedical text mining.

4.2 Experiment 1: PPI sentence identification

Method: To evaluate the performance of our system, we measure how well the relationships discovered compare with manually selected PPI sentences. To do so, we follow the same procedure and data sets used to evaluate semi-supervised classification of PPI sentences (Erkan et al., 2007). The two data sets are AIMED and CB, which have been marked for protein entities and interaction phrases[3].

For each sentence in which n proteins appear, we build $\binom{n}{2}$ phrases. Each phrase consists of the words between each entity combination, and is labeled as positive if it describes a PPI, or negative otherwise. This results in 4026 phrases for the

[3] Available in preprocessed form at `http://belabog.si.umich.edu/biocreative`

AIMED data set (951 positive, 3075 negative), and 4056 phrases for the CB data set (2202 positive, 1854 negative).

The output of the discovery phase is a clustering of RPs. For purpose of this experiment, we ignore the partition and treat the phrases in aggregate. A phrase in the evaluation data set is classified as positive (describing a PPI) if any substring of the phrase matches an RP in our output. For example, if the phrase is:

A significantly inhibited B

and the string 'inhibited' appears as a relation in our output, then this phrase is marked positive. Otherwise, the phrase is marked negative.

Performance is evaluated using standard metrics of precision (P), recall (R), and F-measure (F_1), defined as:

$$P = \frac{TP}{TP + FP}; \quad R = \frac{TP}{TP + FN}$$

where TP is the number of phrases correctly identified as describing a PPI, FP is the number of phrases incorrectly classified as describing a relation, and FN is the number of interaction phrases (positives) marked negative. F_1 is defined as:

$$F_1 = \frac{2PR}{P + R}$$

We calculate P, R, and F_1 for three parameters affecting which phrases are identified as expressing a relation:

- the minimum co-occurrence threshold that controls which entity tuples are kept as likely to stand in some fixed relation
- the minimum cluster size that controls which groups of relations are discarded
- the minimum RP length that controls the smallest number of words appearing in relations

The threshold on the length of the relations can be thought of as controlling the amount of contextual information expressed. A single term relation will be very general, while longer RPs express a relation very specific to the context in which they are written. The results are reported in Figures 1 through 6. Odd numbered figures use the AIMED corpus; even numbered figures the CB corpus. **Results:** Discarding clusters below a certain size had no significant effect on precision. However, this step is still necessary for bootstrapping RE, since machine learning approaches require a sufficient number of positive examples to train the extractor.

Table 2: Comparison with supervised methods–AIMED corpus

Method	P	R	F_1
RD-F_1	30.08	60.67	40.22
RD-P	**55.17**	5.04	9.25
(Yakushiji et al., 2005)	33.70	33.10	33.40
(Mitsumori et al., 2006)	54.20	42.60	47.70
(Erkan et al., 2007)	**59.59**	60.68	59.96

Table 3: Comparison with supervised methods–CB corpus

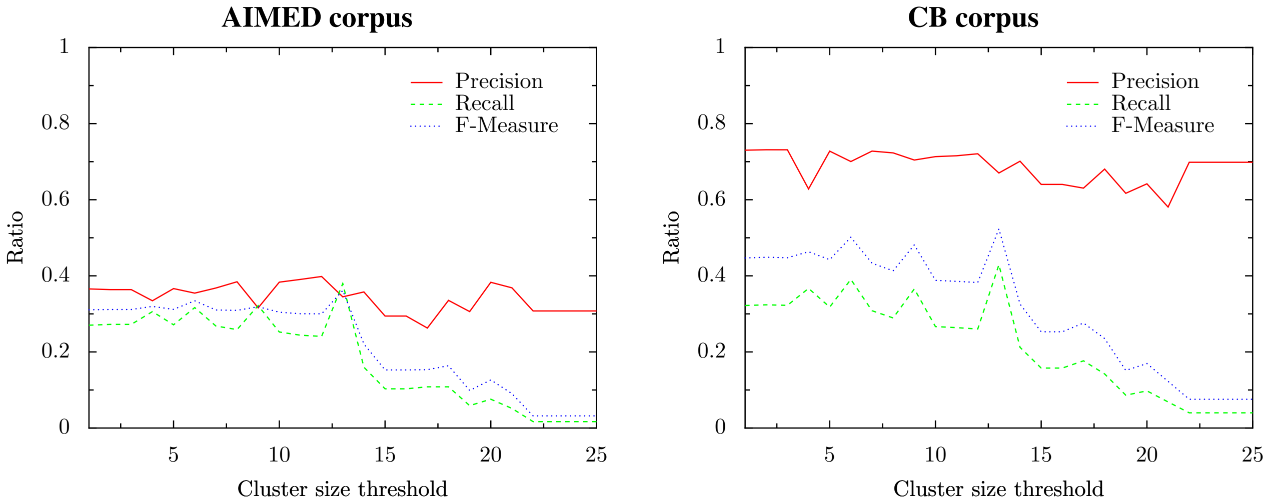
Method	P	R	F_1
RD-F_1	65.03	69.16	67.03
RD-P	**86.27**	2.00	3.91
(Erkan et al., 2007)	**85.62**	84.89	85.22

On the other hand, our results confirm the observation that frequently co-occurring pairs of entities are likely to stand in a fixed relation. On the CB corpus, precision ranges from 0.63 to 0.86 for phrases between entities co-occurring at least 50 times. On the AIMED corpus, precision ranges from 0.29 to 0.55 in the same threshold range.

The minimum phrase length had the most impact on performance, which was particularly evident in the CB corpus: this corpus reached perfect precision discarding all RPs of fewer than 3 words. Lower thresholds result in significantly more relations, at the cost of precision.

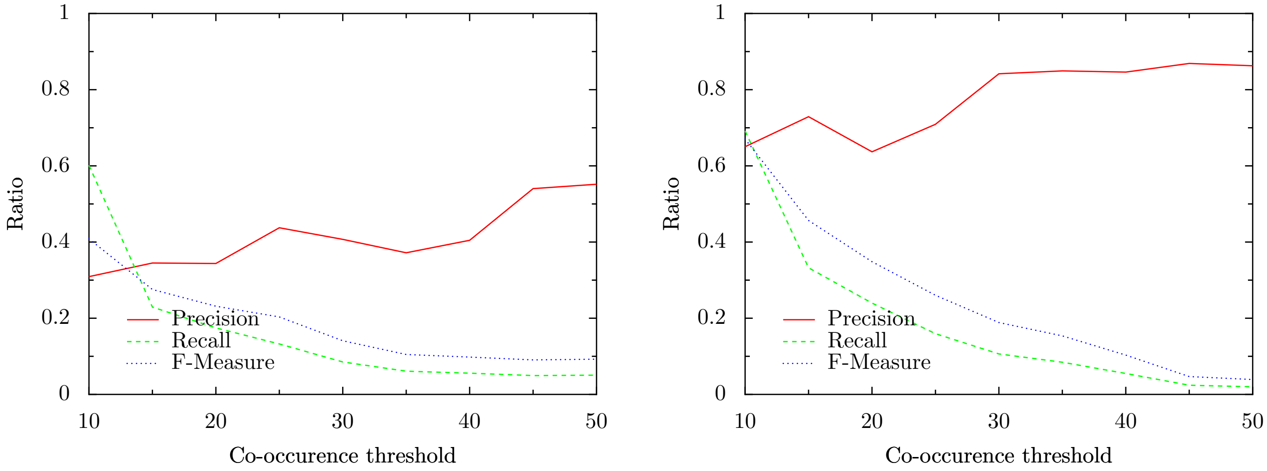
The generally lower performance on the AIMED corpus suggests that our training data (retrieved from the seed proteins) provided less coverage for those interactions than for the those in the CB corpus.

Table 2 and Table 3 compare our results at fixed parameter settings with supervised approaches. RD-F_1 reports parameters which give highest recall and RD-P highest precision. Specifically, both RD-F_1 and RD-P use a minimum RP length of 1, RD-F_1 uses a co-occurrence threshold of 10, and RD-P uses a co-occurrence threshold of 50. As expected, RD alone does not match combined precision and recall of state-of-the-art supervised systems. However, we show better performance than expected. RD-F_1 outperforms the best results of (Yakushiji et al., 2005). RD-P settings outperform or match the precision of top-performing systems on both datasets.

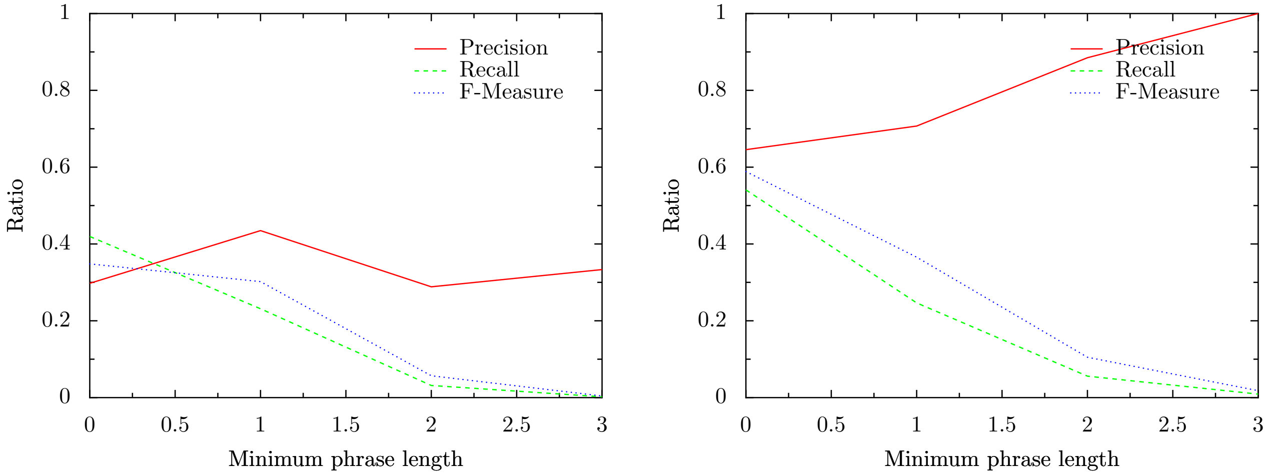

AIMED corpus

CB corpus

Figures 1 & 2: Performance as minimum cluster size is adjusted



Figures 3 & 4: Performance as co-occurence threshold is adjusted



Figures 5 & 6: Performance as minimum phrase length is adjusted

4.3 Experiment 2: clustering relations

Method: We evaluate the appropriateness of the p-median clustering as follows. For each cluster, we take the cluster exemplar as defining the base relation. If the base relation does not express something meaningful, then we mark each member of the cluster incorrect. Otherwise, we label each member of the cluster either as semantically similar to the exemplar (correct) or different than the exemplar (incorrect). Thus, clusters with inappropriate exemplars are heavily penalized. These results are reported in Table 4. For purpose of this experiment, we use the same paramaters as for RD-P, and evaluate the 20 largest clusters.

Results: In the 20 largest clusters, each cluster exemplar expressed something meaningful. 3 of the cluster exemplars were not representative of their other members. We found that most error was due to stopwords not being considered in our similarity calculations. For example, 'detected by' and 'detected in' express the same relationship in our similarity calculations; however, they are clearly quite different. Another source of error evident in Table 4 are mistakes in the pattern and entropy based chunking. The exemplar 'mrna expression in' includes the token 'mrna', which belongs with the left protein NP in the relation chosen as an exemplar.

5 Related work

RD is a relatively new area of research. Existing methods differ primarily in the amount of supervision required and in how contextual features are defined and used.

(Hasegawa et al., 2004) use NER to identify frequently co-occurring entities as likely relation phrases. As in this work, they use the vector model and cosine similarity to define a measure of similarity between relations, but build relation vectors out of *all* instances of each frequently co-occurring entity pair. Therefore, each mention of the same co-occurring pair is assumed to express the same relationship. These aggregate feature vectors are clustered using complete-linkage HAC, and cluster exemplars are determined by manual inspection for evaluation purposes. (Shinyama and Sekine, 2006) rely further on supervised methods, defining features over a full syntactic parse, and exploit multiple descriptions of the same event in newswire to identify useful relations.

(Rosenfeld and Feldman, 2006) consider the use of RD for unsupervised relation extraction, and use a more complex pattern-learning approach to define feature vectors to cluster candidate relations, reporting gains in accuracy compared with the tf × idf weighed features used in (Hasegawa et al., 2004) and in this work. They also use HAC, and do not address the description of the relations. Arbitrary noun phrases obtained through shallow parsing are used as entities. (Rosenfeld and Feldman, 2007) use a feature ranking scheme using separability-based scores, and compare the performance of different variants of HAC (finding single-linkage to perform best). The complexity of the feature ranking-scheme described can be greater than the clustering itself; in contrast, while we use simple features, our approach is much more efficient.

Table 4: Base relations identified using RP-P parameters

Exemplar	**Size**	P (%)
by activation of	33	87.9
was associated with	28	92.9
was induced by	24	83.3
was detected by	24	83.3
as compared with the	25	92.0
were measured with	23	87.0
mrna expression in	21	**9.5**
in response to	21	95.23
was determined by	21	90.4
with its effect in	19	**10.5**
was correlated with	18	100.0
by induction of	16	93.8
for binding to	16	75.0
is mediated by	16	93.8
was observed by	16	50.0
is an important	15	66.6
increased expression of	15	60.0
related to the	15	93.3
protein production as well as	15	**33.3**
dependent on	14	85.7
Median precision: 86.35		

(Davidov et al., 2007) introduce the use of term frequency patterns for relationship discovery. However, they search for a specific type of relationship; namely, attributes common to all entities of a particular type (for example, all countries have the attribute *capital*), and use a special purpose set of filters rather than entity co-occurrence and clustering. Our work can be seen as a generalization of theirs to relationships of any kind, and we extend the use of frequency patterns to finding general n-gram entities rather than single word entities.

(Madkour et al., 2007) give an excellent overview of biomedical NER and RE. They propose a statistical system for RE, but rely on NER, POS tagging, and the creation of a dictionary for each domain of application. Also, they do not cluster relationships into semantically related groups.

6 Conclusion

Our work makes a series of important improvements to the state-of-the-art in relationship discovery. First, by incorporating entity discovery into the relationship discovery pipeline, our method does not require distinct training phases to accommodate different entity types, relations, or discourse types. Second, p-median clustering effectively uncovers the base form of relations present in the corpus, addressing an important limitation in usability. In terms of specific hypotheses, we have tested and confirmed that co-occurrence can be a good indicator of the presence of a relationship but the size of a cluster is not necessarily a good indicator of the importance or strength of the discovered relationship. Furthermore, we have shown that longer RPs with more context give higher precision (at the cost of reduced coverage). Finally, the integration of ideas in our approach—unsupervisedness, efficiency, flexibility (in application), and specificity—is novel in itself.

In future work, we seek to expand upon our RD methods in three directions. First, we would like to generalize the scope of our discovery pipeline beyond binary relations and with richer considerations of context, even across sentences. Second, we hope to achieve greater tunability of performance, to account for additional discovery metrics besides precision. Finally, we intend to induce entire concept maps from text using the discovered relations to boostrap an RE phase, where the underlying problem is not just of inferring multiple types of relations, but to have sufficient co-ordination among the discovered relations to ensure connectedness among the resulting concepts.

While our method requires no supervision in the form of manually annotated entities or relations, the effectiveness of the system relies on the careful tuning of a number of parameters. Nevertheless, the results reported in Section 4.2 suggest that the two parameters that most significantly affect performance exhibit predictable precision/recall behavior. Of the parameters not considered in Section 4.2, we would like to further investigate the benefits of chunking entities on the resulting base relations, experimenting with different measures of collocation.

Acknowledgements

We would like to thank our anonymous reviewers for their thought-provoking questions. This work was supported in part by the Institute for Critical Technology and Applied Science (ICTAS), Virginia Tech.

References

Dmitry Davidov and Ari Rappoport. 2006. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In *ACL '06: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 297–304, Morristown, NJ, USA. Association for Computational Linguistics.

Dmitry Davidov, Ari Rappoport, and Moshe Koppel. 2007. Fully unsupervised discovery of concept-specific relationships by web mining. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 232–239, Prague, Czech Republic, June. Association for Computational Linguistics.

Beate Dorow, Dominic Widdows, Katarina Ling, Jean-Pierre Eckmann, Danilo Sergi, and Elisha Moses. 2005. Using curvature and markov clustering in graphs for lexical acquisition and word sense discrimination. In *MEANING 05: 2nd workshop organized by the MEANING Project*, Trento, Italy, February.

Gunes Erkan, Arzucan Ozgur, and Dragomir R. Radev. 2007. Semi-supervised classification for extracting protein interaction sentences using dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 228–237.

Brendan J. Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science*, 315:972–976.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 415, Morristown, NJ, USA. Association for Computational Linguistics.

Amgad Madkour, Kareem Darwish, Hany Hassan, Ahmed Hassan, and Ossama Emam. 2007. Bionoculars: Extracting protein-protein interactions from biomedical text. In *Biological, translational, and clinical language processing*, pages 89–96, Prague, Czech Republic, June. Association for Computational Linguistics.

T. Mitsumori, M. Murata, Y. Fukuda, K. Doi, and H. Doi. 2006. Extracting protein-protein interaction information from biomedical text with svm. *IEICE Transactions on Information and Systems*, 89(8):2464–2466.

Benjamin Rosenfeld and Ronen Feldman. 2006. High-performance unsupervised relation extraction from large corpora. In *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, pages 1032–1037, Washington, DC, USA. IEEE Computer Society.

Benjamin Rosenfeld and Ronen Feldman. 2007. Clustering for unsupervised relation identification. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 411–418, New York, NY, USA. ACM.

Sayori Shimohata, Toshiyuki Sugio, and Junji Nagata. 1997. Retrieving collocations by co-occurrences and word order constraints. In *In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 476–481.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 304–311, New York City, USA, June. Association for Computational Linguistics.

A. Yakushiji, Y. Miyao, Y. Tateisi, and J. Tsujii. 2005. Biomedical information extraction with predicate-argument structure patterns. In *Proceedings of the eleventh annual meeting of the association for natural language processing*, pages 93–96.