Clustering-based Stratified Seed Sampling for Semi-Supervised Relation Classification

Longhua Qian
Natural Language Processing Lab
School of Computer Science and Technology
Soochow University
1 Shizi Street, Suzhou, China 215006
qianlonghua@suda.edu.cn

Guodong Zhou
Natural Language Processing Lab
School of Computer Science and Technology
Soochow University
1 Shizi Street, Suzhou, China 215006
gdzhou@suda.edu.cn

Abstract

Seed sampling is critical in semi-supervised learning. This paper proposes a clustering-based stratified seed sampling approach to semi-supervised learning. First, various clustering algorithms are explored to partition the unlabeled instances into different strata with each stratum represented by a center. Then, diversity-motivated intra-stratum sampling is adopted to choose the center and additional instances from each stratum to form the unlabeled seed set for an oracle to annotate. Finally, the labeled seed set is fed into a bootstrapping procedure as the initial labeled data. We systematically evaluate our stratified bootstrapping approach in the semantic relation classification subtask of the ACE RDC (Relation Detection and Classification) task. In particular, we compare various clustering algorithms on the stratified bootstrapping performance. Experimental results on the ACE RDC 2004 corpus show that our clustering-based stratified bootstrapping approach achieves the best F1-score of 75.9 on the subtask of semantic relation classification, approaching the one with golden clustering.

1 Introduction

Semantic relation extraction aims to detect and classify semantic relationships between a pair of named entities occurring in a natural language text. Many machine learning approaches have been proposed to attack this problem, including supervised (Miller et al., 2000; Zelenko et al., 2003; Culotta and Soresen, 2004; Kambhatla, 2004; Zhao and Grishman, 2005; Zhou et al., 2005; Zhang et al., 2006; Zhou and Zhang, 2007; Zhou et al., 2007; Qian et al., 2008; Zhou et al., 2010), semi-supervised (Brin, 1998; Agichtein and Gravano, 2000; Zhang, 2004; Chen et al., 2006; Qian et al., 2009; Zhou et al., 2009), and unsupervised methods (Hasegawa et al., 2004; Zhang et al., 2005; Chen et al., 2005).

Current work on relation extraction mainly adopts supervised learning methods, since they achieve much better performance. However, they normally require a large number of manually labeled relation instances, whose acquisition is both time consuming and labor intensive. In contrast, unsupervised methods do not need any manually labeled instances. Nevertheless, it is difficult to assess their performance due to the lack of evaluation criteria. As something between them, semi-supervised learning has received more and more attention recently. With the plenitude of unlabeled natural language text at hand, semi-supervised learning can significantly reduce the need for labeled data with only limited sacrifice in performance. For example, Abney (2002) proposes a bootstrapping algorithm which chooses the unlabeled instances with the highest probability of being correctly labeled and add them in turn into the labeled training data iteratively.

This paper focuses on bootstrapping-based semi-supervised learning in relation extraction. Since the performance of bootstrapping depends much on the quality and quantity of the seed set and researchers tend to employ as few seeds as possible (e.g. 100 instances) to save time and labor, the quality of the seed set plays a critical role in bootstrapping. Furthermore, the imbalance of different classes and

Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 346–355,
MIT, Massachusetts, USA, 9-11 October 2010. ©2010 Association for Computational Linguistics

the inherent structural complexity of instances will severely weaken the strength of bootstrapping and semi-supervised learning as well. Therefore, it is critical for a bootstrapping procedure to select an appropriate seed set, which should be representative and diverse. However, most of current semi-supervised relation extraction systems (Zhang, 2004; Chen et al., 2006) use a random seed sampling strategy, which fails to fully exploit the affinity nature in the training data to derive the seed set. Alternatively, Zhou et al. (2009) bootstrap a set of weighted support vectors from both labeled and unlabeled data using SVM and feed these instances into semi-supervised relation extraction. However, their seed set is sequentially generated only to ensure that there are at least 5 instances for each relation class. Our previous work (Qian et al., 2009) attempts to solve this problem via a simple stratified sampling strategy for selecting the seed set. Experimentation on the ACE RDC 2004 corpus shows that the stratified sampling strategy achieves promising results for semi-supervised learning. Nevertheless, the success of the strategy relies on the assumption that the true distribution of all relation types is already known, which is impractical for real NLP applications.

This paper presents a clustering-based stratified seed sampling approach for semi-supervised relation extraction, without the assumption on the true distribution of different relation types. The motivations behind our approach are that the unlabeled data can be partitioned into a number of strata using a clustering algorithm and that representative and diverse seeds can be derived from such strata in the framework of stratified sampling (Neyman, 1934) for an oracle to annotate. Particularly, we employ a diversity-motivated intra-stratum sampling scheme to pick a center and additional instances as seeds from each stratum. Experimental results show the effectiveness of the clustering-based stratified seed sampling for semi-supervised relation classification.

The rest of this paper is organized as follows. First an overview of the related work is given in Section 2. Then, Section 3 introduces the stratified bootstrapping framework including an intra-stratum sampling scheme while Section 4 describes various clustering algorithms. The experimental results on the ACE RDC 2004 corpus are reported in Section 5. Finally we conclude our work and indicate some future directions in Section 6.

2 Related Work

In semi-supervised learning for relation extraction, most of previous work construct the seed set either randomly (Zhang, 2004; Chen et al., 2006) or sequentially (Zhou et al., 2009). Qian et al. (2009) adopt a stratified sampling strategy to select the seed set. However, their method needs a stratification variable such as the known distribution of the relation types, while our method uses clustering to divide relation instances into different strata.

In the literature, clustering techniques have been employed in active learning to sample representative seeds in a certain extent (Nguyen and Smeulders, 2004; Tang et al., 2002; Shen et al., 2004). Our work is similar to the formal framework, as proposed in Nguyen and Smeulders (2004), in which K-medoids clustering is incorporated into active learning. The cluster centers are used to construct a classifier and which in turn propagates classification decision to other examples via a local noise model. Unlike their probabilistic models, we apply various clustering algorithms together with intra-stratum sampling to select a seed set in discriminative models like SVMs. In active learning for syntactic parsing, Tang et al. (2002) employ a sampling strategy of "most uncertain per cluster" to select representative examples and weight them using their cluster density, while we pick a few seeds (the number of the sampled seeds is proportional to the cluster density) from a cluster in addition to its center. Shen et al. (2004) combine multiple criteria to measure the informativeness, representativeness, and diversity of examples in active learning for named entity recognition. Unlike our sampling strategy of clustering for representativeness and stratified sampling for diversity, they either select cluster centroids or diverse examples from a prechosen set in terms of some combined metrics. To the best of our knowledge, this is the first work to address the issue of seed selection using clustering techniques for semi-supervised learning with discriminative models.

3 Stratified Bootstrapping Framework

The stratified bootstrapping framework consists of three major components: an underlying supervised learner and a bootstrapping algorithm on top of it as usual, plus a clustering-based stratified seed sampler.

3.1 Underlying Supervised Learner

Due to recent success in tree kernel-based relation extraction, this paper adopts a tree kernel-based method in the underlying supervised learner. Following the previous work in relation extraction (Zhang et al., 2006; Zhou et al., 2007; Qian et al., 2008), we use the standard convolution tree kernel (Collins and Duffy, 2001) to count the number of common sub-trees as the structural similarity between two parse trees. Besides, to properly represent a relation instance, this paper adopts the Unified Parse and Semantic Tree (UPST), as proposed in Qian et al. (2008). To our knowledge, the USPT has achieved the best performance in relation extraction so far on the ACE RDC 2004 corpus.

In particular, we use the SVM^{light}-TK[1] package as our classifier. Since the package is a binary classifier, we adapt it to the multi-class tasks of relation extraction by applying the *one vs. others* strategy, which builds K binary classifiers so as to separate one class from all others. The final classification decision of an instance is determined by the class that has the maximal SVM output margin.

3.2 Bootstrapping Algorithm

Following Zhang (2004), we have developed a baseline self-bootstrapping procedure, which keeps augmenting the labeled data by employing the models trained from previously available labeled data, as shown in Figure 1.

Since the SVM^{light}-TK package doesn't output any probability that it assigns to the class label on an instance, we devise a metric to measure the confidence with regard to the classifier's prediction. Given a sequence of output margins of all K binary classifiers at some iteration, denoted as $\{m_1, m_2, \ldots m_K\}$ with m_i the margin for the i-th classifier, we compute the margin gap between the largest and the mean of the others, i.e.

$$H = \max_{i=1}^{K} m_i - (\sum_{i=1}^{K} m_i - \max_{i=1}^{K} m_i)/(K-1) \tag{1}$$

Where K denotes the total number of relation classes, and m_i denotes the output margin of the i-th classifier. Intuitively, the bigger the H, the greater the difference between the maximal margin and all others, and thus the more reliably the classifier makes the prediction on the instance.

```
**Algorithm** self-bootstrapping
**Require**: labeled seed set L
**Require**: unlabeled data set U
**Require**: batch size S
**Repeat**
    Train a single classifier on L
    Run the classifier on U
    Find at most S instances in U that the classifier has
the highest prediction confidence
    Add them into L
**Until**: no data points available or the stoppage
condition is reached
```

Figure 1: Self-bootstrapping algorithm

3.3 Clustering-based Stratified Seed Sampler

Stratified sampling is a method of sampling in statistics, in which the members of a population are grouped into relatively homogeneous subgroups (i.e. strata) according to one certain property, and then a sample is selected from each stratum. This process of grouping is called stratification, and the property on which the stratification is performed is called the stratification variable. Previous work justifies theoretically and practically that stratified sampling is more appropriate than random sampling for general use (Neyman, 1934) as well as for relation extraction (Qian et al., 2009). However, the difficulty lies in how to find the appropriate stratification variable for complicated tasks, such as relation extraction.

The idea of clustering-based stratification circumvents this problem by clustering the unlabeled data into a number of strata without the need to explicitly specify a stratification variable. Figure 2 illustrates the clustering-based stratified seed sampling strategy employed in the bootstrapping procedure, where *RSET* denotes the whole unlabeled data, *SeedSET* the seed set to be labeled and $|RSET_i|$ the number of instances in the i-th cluster[2] $RSET_i$. Here, a relation instance is represented using USPT and the similarity between two instances is computed using the standard convolution tree

[1] http://ai-nlp.info.uniroma2.it/moschitti/

[2] Hereafter, when we refer to clusters from the viewpoint of stratified sampling, they are often called "strata".

kernel, as described in Section 3.1 (i.e., both the clustering and the classification adopt the same structural representation, since we want the representative seeds in the clustering space to be also representative in the classification space). After clustering, a certain number of instances from every stratum are sampled using an intra-stratum scheme (c.f. Subsection 3.4). Normally, this number is proportional to the size of that stratum in the whole data set. However, in case this number is 0 due to the rounding of real numbers, it is set to 1 to ensure the existence of at least one seed from that stratum. Furthermore, to ensure that the total number of instances being sampled equals the prescribed N_S, the number of seeds from dominant strata may be slightly adjusted accordingly. Finally, these instances form the unlabeled seed set for an oracle to annotate as the input to the underlying supervised learner in the bootstrapping procedure.

3.4 Intra-stratum sampling

Given the distribution of clusters, a simple way to select the most representative instances is to choose the center of each cluster with the cluster prior as the weight of the center (Tang et al., 2002; Nguyen and Smeulders, 2004). Nevertheless, for the complicated task of relation extraction on the ACE RDC corpora, which is highly skewed across different relation classes, only considering the center of each cluster would severely under-represent the high-density data. To overcome this problem, we adopt a sampling approach, in particular stratified sampling, which takes the size of each stratum into consideration.

Given the size of the seed set N_S and the number of strata K, a natural question will arise as how to select the remaining (N_S-K) seeds after we have extracted the K centers from the K strata. We view this problem as intra-stratum sampling, which is required to choose the remaining number of seeds from inside individual stratum (excluding the centers themselves).

At the first glance, sampling a certain number of seeds from one particular stratum (e.g., $RSET_i$), seems to be the same sampling problem as we have encountered before, which aims to select the most representative and diverse seeds. This will naturally lead to another application of a clustering algorithm to the stratification of the stratum $RSET_i$.

```
Require: $RSET = \{R_1, R_2, \ldots, R_N\}$, the set of unlabeled relation instances and $K$, the number of strata being clustered
Output: $SeedSET$ with the size of $N_S$ (100)
Procedure
    Initialize $SeedSET = NULL$
    Cluster $RSET$ into $K$ strata using a clustering algorithm and perform stratum pruning if necessary.
    Calculate the number of instances being sampled for each stratum $i=\{1,2,\ldots,K\}$
        $N_i = \frac{|RSET_i|}{N} * N_S$        (2)
        and adjust this number if necessary.
    Perform intra-strata sampling to form $SeedSET_i$ from each stratum $RSET_i$, by selecting the center $C_i$ and ($N_i$-1) additional instances
    Generate $SeedSET$ by summating $RSET_i$ from each stratum
```

Figure 2: Clustering-based stratified seed sampling

Nevertheless, remember the fact that, this time for the stratum $RSET_i$, the center C_i has been chosen, so it may not be reasonable to extract additional centers in this way. Therefore, in order to avoid recursion and over-complexity, we employ a diversity-motivated intra-stratum sampling scheme (Shen et al., 2004), called KDN (K-diverse neighbors), which aims to maximize the training utility of all seeds from a stratum. The motivation is that we prefer the seeds with high variance to each other, thus avoiding repetitious seeds from a single stratum. The basic idea is to add a candidate instance to the seed set only if it is sufficiently different from any previously selected seeds, i.e., the similarity between the candidate instance and any of the current seeds is less than a threshold β. In this paper, the threshold β is set to the average pair-wise similarity between any two instances in a stratum.

4 Clustering Algorithms

This section describes several typical clustering algorithms in the literature, such as K-means, HAC, spectral clustering and affinity propagation, as well as their application in this paper.

4.1 K-medoids (KM)

As a simple yet effective clustering method, the K-means algorithm assigns each instance to the cluster whose center (also called centroid) is nearest. In particular, the center is the average of all the instances in the cluster, i.e., with its coordinates the arithmetic means for each dimension separately over all the instances in the cluster.

One problem with K-means is that it does not yield the same result with each run while the other problem is the requirement for the concept of a mean to be definable, which is unfortunately not available in our setting (we employ a parse tree representation for a relation instance). Hence, we adopt a variant of K-means, namely, K-medoids, where a medoid, rather than a centroid, is defined as a representative of a cluster. Besides, K-medoids has proved to be more robust to noise and outliers in comparison with K-means.

4.2 Hierarchical Agglomerative Clustering (HAC)

Different from K-medoids, hierarchical clustering creates a hierarchy of clusters which can be represented in a tree structure called a dendrogram. The root of the tree consists of a single cluster containing all objects, and the leaves correspond to individual object.

Typically, hierarchical agglomerative clustering (HAC) starts at the leaves and successively merges two clusters together as long as they have the shortest distance among all the pair-wise distances between any two clusters.

Given a specified number of clusters, the key problem is to determine where to cut the hierarchical tree into clusters. In this paper, we generate the final flat cluster structures greedily by maximizing the equal distribution of instances among different clusters.

4.3 Spectral Clustering (SC)

Spectral clustering has become more and more popular recently. Taking as input a similarity matrix between any two instances, spectral clustering makes use of the spectrum of the similarity matrix of the data to perform dimensionality reduction for clustering in fewer dimensions.

Compared to the "traditional algorithms" such as K-means or HAC, spectral clustering has many fundamental advantages. Results obtained by spectral clustering often outperform the traditional approaches. Furthermore, spectral clustering is very simple to implement and can be solved efficiently using standard linear algebra methods (von Luxburg, 2006).

4.4 Affinity Propagation (AP)

As a new emerging clustering algorithm, affinity propagation (AP) (Frey and Dueck, 2007) is basically an iterative message-passing procedure in which the instances being clustered compete to serve as cluster exemplars by exchanging two types of messages, namely, "responsibility" and "availability". After the procedure converges or has repeated a finite number of iterations, each cluster is represented by an exemplar. AP was reported to find clusters with much lower error than those found by other methods.

For our application, affinity propagation takes as input a similarity matrix, whose elements represent either the similarity between two different instances or the preference (a real number *p*) for an instance when two instances are the same. One problem with AP is that the number of clusters cannot be pre-defined, which is indirectly determined by the preference as well as the convergence procedure itself.

5 Experimentation

This section systematically evaluates the bootstrapping approach using clustering-based stratified seed sampling, in the relation classification (i.e., given the relationship already detected) subtask of relation extraction on the ACE RDC 2004 corpus.

5.1 Experimental Setting

The ACE RDC 2004 corpus[3] is gathered from various newspapers, newswire and broadcasts. It contains 451 documents and 5702 positive relation instances of 7 relation types and 23 subtypes between 7 entity types. For easy reference with related work in the literature, evaluation is done on 347 documents (from *nwire* and *bnews* domains), which include 4305 relation instances. Table 1 lists the major relation types and subtypes, including their corresponding instance numbers and ratios in our evaluation set. One obvious observation from the table is that the numbers of different relation types is highly imbalanced. These 347 documents are then divided into 3 disjoint sets randomly, with

[3] http//www.ldc.upenn.edu/ Projects/ACE/

Types	Subtypes	#	%
PHYS	Located	738	17.1
	Near	87	2.0
	Part-Whole	378	8.8
PER-SOC	Business	173	4.0
	Family	121	2.8
	Other	55	1.3
EMP-ORG	Employ-Executive	489	11.4
	Employ-Staff	539	12.5
	Employ-Undeter.	78	1.8
	Member-of-Group	191	4.4
	Subsidiary	206	4.8
	Partner	12	0.3
	Other	80	1.9
ART	User-or-Owner	200	4.6
	Inventor-or-Man.	9	0.2
	Other	2	0.0
OTHER-AFF	Ethnic	39	0.9
	Ideology	48	1.1
	Other	54	1.3
GPE-AFF	Citizen-or-Resid.	273	6.3
	Based-In	215	5.0
	Other	39	0.9
DISC		279	6.5
Total		4305	100.0

Table 1: Relation types and their corresponding instance numbers and ratios in the ACE RDC 2004 corpus

10% of them (35 files, around 400 instances) held out as the test data set, 10% of them (35 files, around 400 instances) used as the development data set to fine-tune various settings and parameters, while the remaining 277 files (over 3400 instances) used as the training data set, from which the seed set will be sampled.

The corpus is parsed using Charniak's parser (Charniak, 2001) and relation instances are generated by extracting all pairs of entity mentions occurring in the same sentence with positive relationships. For easy comparison with related work, we only evaluate the relation classification task on the 7 major relation types of the ACE RDC 2004 corpus. For the SVM^{light}-TK classifier, the training parameters C (SVM) and λ (tree kernel) are fine-tuned to 2.4 and 0.4 respectively.

The performance is measured using the standard P/R/F1 (Precision/Recall/F1-measure). For each relation type, P is the ratio of the true relation instances in all the relation instances being identified, R is the ratio of the true relation instances being identified in all the true relation instances in the corpus, and F1 is the harmonic mean of P and R. The overall performance P/R/F1 is then calculated using the micro-average measure over all major class types.

5.2 Experimental Results

Comparison of various seed sampling strategies without intra-stratum sampling on the development data

Table 2 compares the performance of bootstrapping-based relation classification using various seed sampling strategies without intra-stratum sampling on the development data. Here, the size of the seed set *L* is set to 100, and the top 100 instances with the highest confidence (c.f. Formula 1) are augmented at each iteration. For sampling strategies marked with an asterisk, we performed 10 trials and calculated their averages. Since for these strategies the seed sets sampled from different trials may be quite different, their performance scores vary in a great degree accordingly. This experimental setting and notation are also used in all the subsequent experiments unless specified. Besides, two additional baseline sampling strategies are included for comparison: sequential sampling (SEQ), which selects a sequentially-occurring *L* instances as the seed set, and random sampling (RAND), which randomly selects *L* instances as the seed set.

Table 2 shows that

1) RAND outperforms SEQ by 1.2 units in F1-score. This is due to the fact that the seed set via RAND may better reflect the distribution of the whole training data than that via SEQ, nevertheless at the expense of collecting the whole training data in advance.
2) While HAC performs moderately better than RAND, it is surprising that both KM and AP perform even worse than SEQ, and that SC performs worse than RAND. Furthermore, all the four clustering-based seed sampling strategies achieve much smaller performance improvement in F1-score than RAND, among which KM performs worst with performance improvement of only 0.1 in F1-score.

Sampling strategies	P(ΔP)	R(ΔR)	F1(ΔF1)
RAND*	69.1(3.1)	66.4(0.2)	67.8(2.0)
SEQ*	65.8(2.6)	68.0(0.1)	66.6(1.3)
KM*	62.0(0.9)	61.0(-0.5)	61.3(0.1)
HAC	**69.9(1.3)**	**70.4(0.4)**	**70.1(0.8)**
SC*	67.1(1.5)	68.1(0.0)	67.5(0.8)
AP	66.6(2.0)	66.2(0.1)	66.4(1.1)

Table 2: Comparison of various seed sampling strategies without intra-stratum sampling on the development data

3) All the performance improvements from bootstrapping largely come from the improvements in precision. While the bootstrapping procedure makes the SVM classifier more accurate, it lacks enough generalization ability.

To explain above special phenomena, we have a look at the clustering results. Our inspection reveals that most of them are severely imbalanced, i.e., some clusters are highly dense while others are extremely sparse. This indicates that merely selecting the centers from each cluster cannot properly represent the overall distribution. Moreover, the centers with high density lack the generalization ability due to its solitude in the cluster, leading to less performance enhancement than expected.

The only exception is HAC, which much outperforms RAND by 2.3 in F1-score, although HAC is usually not considered as an effective clustering algorithm. The reason may be that HAC creates a hierarchy of clusters in the top-down manner by cutting a cluster into two. Therefore, the centers in the two sibling clusters will be closer to each other than they are to the centers in other clusters. Besides, the final flat cluster structures given a special number of clusters are generated greedily from the cluster hierarchy by maximizing the equal distribution of instances among different clusters. In other words, when the cluster number reaches a certain threshold, the dense area will get more seeds represented in the seed set. As a consequence, the distribution of all the seeds sampled by HAC will approximate the distribution of the whole training data in some degree, while the seeds sampled by other clustering algorithm are kept as far as possible due to the objective of clustering and the lack of intra-stratum sampling.

These observations also justify the application of the stratified seed sampling to the bootstrapping procedure, which enforces the number of seeds sampled from a cluster to be proportional to its density, presumably approximated by its size in this paper.

Comparison of different cluster numbers with intra-stratum sampling on the development data

In order to fine-tune the optimal cluster numbers for seed sampling, we compare the performance of different numbers of clusters for each clustering algorithm on the development data set and report their F-scores in Table 3. For reference, we also list the F-score for golden clustering (GOLD), in which all instances are grouped in terms of their annotated ground relation major types (7), major types considering relation direction (13), subtypes (23), and subtypes considering direction (38). Besides, the performance of clustering-based semi-supervised relation classification is also measured over other typical cluster numbers (i.e., 1, 50, 60, 80, 100). Particularly, when the cluster number equals 1, it means that only diversity other than representativeness is considered in the seed sampling. Among these clustering algorithms, one of the distinct characteristics with the AP algorithm is that the number of clusters cannot be specified in advance, rather, it is determined by the pre-defined preference parameter (c.f. Subsection 4.4). Therefore, we should tune the preference parameter so as to get the pre-defined cluster number. However, sometimes we still couldn't get the exact number of clusters as we expected. In these cases, we use the approximate cluster numbers for AP instead.

Table 3 shows that

1) The performance for all the clustering algorithms varies in some degree with the number of clusters being grouped. Interestingly, the performance with only one cluster is better than those of clustering-based strategies with 100 clusters, at most cases. This implies that the diversity of the seeds is at least as important as their representativeness. And this could be further explained by our observation that, with the increase of cluster numbers, the clusters get smaller and denser while their centers also come closer to each other. Therefore, the representativeness and diversity as well as the distribution of the seeds sampled from them may vary accordingly, leading to different performance.

# of Clusters	GOLD	KM*	HAC	SC*	AP
1	-	68.7	68.7	-	-
7	**73.9**	70.3	**73.3**	**72.1**	-
13	70.2	68.9	70.3	67.3	-
23	64.9	**72.3**	72.9	68.9	71.1
38	60.8	69.9	71.6	68.0	**71.6**
50	-	68.5	69.9	68.5	70.4
60	-	66.3	68.5	68.6	69.7
80	-	64.2	65.9	68.0	68.1
100	-	61.3	70.1	67.5	66.4

Table 3: Performance in F1-score over different cluster numbers with intra-stratum sampling on the development data

2) Golden clustering achieves the best performance of 73.9 in F1-score when the cluster number is set to 7, significantly higher than the performance using other cluster numbers. Interestingly, this number coincides with the number of major relation types needed to be classified in our task. This is reasonable since the instances with the same relation type should be much more similar than those with different relation types and it is easy to discriminate the seed set of one relation type from that of other relation types.
3) Among the four clustering algorithms, HAC achieves best performance over most of cluster numbers. This further verifies the aforementioned analysis. That is, as a hierarchical clustering algorithm, HAC can sample seeds that better capture the distribution of the training data.
4) For KM, the best performance is achieved around the number of 23 while for both HAC and SC, the optimal cluster number is consistent with GOLD clustering, namely, 7. For AP, the optimal cluster number for AP is 38. This is largely due to that we fail to cluster the training data into about 7 and 13 groups no matter how we vary the preference parameter.

Final comparison of different clustering algorithms on the held-out test data

After the optimal cluster numbers are determined for each clustering algorithm, we apply these numbers on the held-out test data and report the performance results (P/R/F1 and their respective improvements) in Table 4. For easy reference, we also include the performance for GOLD, RAND, and SEQ sampling strategies.

Sampling strategies	P(ΔP)	R(ΔR)	F1(ΔF1)
GOLD	**79.5(7.8)**	72.7(2.1)	**76.0**(4.8)
RAND*	71.9(3.7)	69.7(0.1)	70.8(1.8)
SEQ*	71.9(2.6)	65.2(0.1)	69.3(1.3)
KM*	73.6(2.1)	72.3(0.3)	72.9(1.2)
HAC	79.0(10.2)	**73.0(1.1)**	**75.9(5.6)**
SC*	72.3(2.1)	72.1(0.4)	72.2(1.2)
AP	75.7(2.5)	72.0(0.4)	73.7(1.4)

Table 4: Performance of various clustering-based seed sampling strategies on the held-out test data with the optimal cluster number for each clustering algorithm

Table 4 shows that

1) Among all the clustering algorithms, HAC achieves the best F1-score of 75.9, significantly higher than RAND and SEQ by 5.1 and 6.6 respectively. The improvement comes not only from significant precision boost, but also from moderate recall increase. This further justifies the merits of HAC as a clustering algorithm for stratified seed sampling in semi-supervised relation classification.
2) HAC approaches the best F1-score of 76.0 for golden clustering. Obviously, this doesn't mean HAC performs as well as golden clustering in terms of clustering quality measures, rather it does imply that HAC achieves the performance improvement by making the seed set better represent the overall distribution over inherent structure of relation instances, while golden clustering accomplishes this using the distribution over relation types. Since the distribution over relation types doesn't always conform to that over instance structures, and for a statistical discriminative classifier, often the latter is more important than the former, it will be no surprise if HAC outperforms golden clustering in some real applications, e.g. clustering-based stratified sampling.

6 Conclusion and Future Work

This paper presents a stratified seed sampling strategy based on clustering algorithms for semi-supervised learning. Our strategy does not rely on any stratification variable to divide the training instances into a number of strata. Instead, the strata are formed via clustering, given a metric measuring the similarity between any two instances. Further, diversity-motivated intra-strata sampling is employed to sample additional instances from within each stratum besides its center. We compare the effect of various clustering algorithms on the performance of semi-supervised learning and find that HAC achieves the best performance since the distribution of its seed set better approximates that of the whole training data. Extensive evaluation on the ACE RDC 2004 benchmark corpus shows that our clustering-based stratified seed sampling strategy significantly improves the performance of semi-supervised relation classification.

We believe that our clustering-based stratified seed sampling strategy can not only be applied to other semi-supervised learning tasks, but also can be incorporated into active learning, where the instances to be labeled at each iteration as well as the seed set could be selected using clustering techniques, thus further reducing the amount of instances needed to be annotated.

For the future work, it is possible to adapt our one-level clustering-based sampling to the multi-level one, where for every stratum it is still possible to divide it into lower sub-strata for further stratified sampling in order to make the seeds better represent the true distribution of the data.

Acknowledgments

This research is supported by Projects 60873150, 60970056, and 90920004 under the National Natural Science Foundation of China.

References

S. Abney. 2002. Bootstrapping. *ACL-2002*.

E. Agichtein and L. Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM international Conference on Digital Libraries (ACMDL 2000)*.

S. Brin. 1998. Extracting patterns and relations from the world wide web. In *WebDB Workshop at 6th International Conference on Extending Database Technology (EDBT 98)*.

E. Charniak. 2001. Intermediate-head Parsing for Language Models. *ACL-2001*: 116-123.

M. Collins and N. Duffy. 2001. Convolution Kernels for Natural Language. *NIPS 2001*: 625-632.

J.X. Chen, D.H. Ji, C.L. Tan, and Z.Y. Niu. 2005. Unsupervised Feature Selection for Relation Extraction. *CIKM-2005*: 411-418.

J.X. Chen, D.H. Ji, and C. L. Tan. 2006. Relation Extraction using Label Propagation Based Semi supervised Learning. *ACL/COLING-2006*: 129-136.

A. Culotta and J. Sorensen. 2004. Dependency tree kernels for relation extraction. *ACL-2004*: 423-439.

B.J. Frey and D. Dueck. 2007. Clustering by Passing Messages between Data Points. *Science*, 315: 972-976.

T. Hasegawa, S. Sekine, and R. Grishman. 2004. Discovering Relations among Named Entities from Large Corpora. *ACL-2004*.

N. Kambhatla. 2004. Combining lexical, syntactic and semantic features with Maximum Entropy models for extracting relations. *ACL-2004(posters)*: 178-181.

S. Miller, H. Fox, L. Ramshaw, and R. Weischedel. 2000. A novel use of statistical parsing to extract information from text. In *Proceedings of the 6th Applied Natural Language Processing Conference*.

J. Neyman. 1934. On the Two Different Aspects of the Representative Method: The Method of Stratified Sampling and the Method of Purposive Selection. *Journal of the Royal Statistical Society*, 97(4): 558-625.

H.T. Nguyen and A. Smeulders. 2004. Active Learning Using Pre-clustering, *ICML*-2004.

L.H. Qian, G.D. Zhou, Q.M. Zhu, and P.D. Qian. 2008. Exploiting constituent dependencies for tree kernel-based semantic relation extraction. *COLING-2008*: 697-704.

L.H. Qian, G.D. Zhou, F. Kong, and Q.M. Zhu. 2009. Semi-Supervised Learning for Semantic Relation Classification using Stratified Sampling Strategy. *EMNLP-2009*: 1437-1445.

D. Shen, J. Zhang, J. Su, G. Zhou and C. Tan. 2004. Multi-criteria-based active learning for named entity recognition. *ACL*-2004.

M. Tang, X. Luo and S. Roukos. 2002. Active Learning for Statistical Natural Language Parsing. *ACL*-2002.

U. von Luxburg. 2006. A tutorial on spectral clustering. Technical report, Max Planck Institute for Biological Cybernetics.

D. Zelenko, C. Aone, and A. Richardella. 2003. Kernel Methods for Relation Extraction. *Journal of Machine Learning Research*, (2): 1083-1106.

M. Zhang, J. Su, D. M. Wang, G. D. Zhou, and C. L. Tan. 2005. Discovering Relations between Named Entities from a Large Raw Corpus Using Tree Similarity-Based Clustering. *IJCNLP-2005*: 378-389.

M. Zhang, J. Zhang, J. Su, and G.D. Zhou. 2006. A Composite Kernel to Extract Relations between Entities with both Flat and Structured Features. *ACL/COLING-2006*: 825-832.

Z. Zhang. 2004. Weakly-supervised relation classification for Information Extraction. *CIKM-2004*.

S.B. Zhao and R. Grishman. 2005. Extracting relations with integrated information using kernel methods. *ACL-2005*: 419-426.

G.D. Zhou, J. Su, J. Zhang, and M. Zhang. 2005. Exploring various knowledge in relation extraction. *ACL-2005*: 427-434.

G.D. Zhou, L.H. Qian, and J.X. Fan. 2010. Tree kernel-based semantic relation extraction with rich syntactic and semantic information. *Information Sciences*, (179): 1785-1791.

G.D. Zhou, L.H. Qian, and Q.M. Zhu. 2009. Label propagation via bootstrapped support vectors for semantic relation extraction between named entities. *Computer Speech and Language*, 23(4): 464-478.

G.D. Zhou and M. Zhang. 2007. Extraction relation information from text documents by exploring various types of knowledge. *Information Processing and Management*, (42):969-982.

G.D. Zhou, M. Zhang, D.H. Ji, and Q.M. Zhu. 2007. Tree Kernel-based Relation Extraction with Context-Sensitive Structured Parse Tree Information. *EMNLP/CoNLL-2007*: 728-736.