2009 Ninth IEEE International Conference on Data Mining

Unsupervised Relation Extraction by Massive Clustering

Edgar Gonzàlez, Jordi Turmo
TALP Research Center
Universitat Politècnica de Catalunya
Barcelona, Spain
{egonzalez,turmo}@lsi.upc.edu

***Abstract*—The goal of Information Extraction is to automatically generate structured pieces of information from the relevant information contained in text documents.**

Machine Learning techniques have been applied to reduce the cost of Information Extraction system adaptation. However, elements of human supervision strongly bias the learning process. Unsupervised learning approaches can avoid these biases.

In this paper, we propose an unsupervised approach to learning for Relation Detection, based on the use of massive clustering ensembles.

The results obtained on the ACE Relation Mention Detection task outperform in terms of F1 score by 5 points the state of the art of unsupervised techniques for this evaluation framework, in addition to being simpler and more flexible.

***Keywords*-Relation Detection, Unsupervised Methods, Ensemble Clustering**

I. Introduction

As the availability of large amounts of textual information is unlimited in practice, supervised processes for mining these data can become highly expensive for human experts. For this reason, unsupervised methods are a central topic for researchers on tasks related to text mining.

One of these tasks is Information Extraction (IE). The goal of IE is to automatically generate structured pieces of information from the relevant information contained in text documents. Part of this information will correspond to binary relations between entities present in the documents.

IE systems require a significant amount of specific linguistic knowledge, and the process of language or domain adaptation of IE systems can hence require significant human effort. In order to reduce this cost, researchers have been exploring Machine Learning techniques for over two decades. A plethora of adaptive IE systems have appeared, and the amount of required human supervision has been progressively reduced, leading to weakly supervised and unsupervised approaches. Most of these systems are able to benefit from the aforementioned increasing availability of huge collections of raw textual data. A detailed survey on adaptive IE can be found in Turmo et. al. [1].

However, even if reduced, elements of human supervision can strongly bias the learning process. To avoid this bias, unsupervised tools for an exploratory analysis of the data collections are needed. In this context, the utility of clustering techniques is well known [2].

In this paper, we propose a novel and flexible unsupervised approach to learning for Relation Detection, based on clustering, which reduces the elements of human supervision and simplifies the use of enriched feature sets with respect to other existing approaches. Our approach has been implemented and tested on the Automatic Content Extraction (ACE) Relation Mention Detection task, ACE being one of the most popular present-day IE evaluation frameworks [3]. The obtained results confirm the validity of our proposal.

The rest of the paper is organized as follows: Section II gives an overview of related work. Our approach is presented in Section III. Next Section IV gives an overview of the experiments carried out and their results. Last, Section V draws conclusions of our work.

II. Related Work

As defined in ACE 2004, the task of Relation Detection is that of, given a set of text documents where entities have been previously detected (by manual or automatic means), identifying the occurrences of relations between such entities (i.e. locating pairs of related entities in text). For instance, in the following passage:

> ***Thousands of people*** *were in* ***the streets*** *and in* ***the basilica*** *to pay tribute. Former president* ***Jimmy Carter*** *represented* ***the United States****.*

the entity **Thousands of people** is related to both **the streets** and **the basilica**, and **Jimmy Carter** is related to **the United States**.

Much research on the Relation Detection task has focused on the application of supervised Machine Learning methods [4], [5], [6]. Some research has been devoted to weakly supervised approaches either based on bootstrapping [7], [8] or on a user-provided description of the extraction task [9]. However, the fact that the output of these systems be sensible to the set of seed tuples or the description of the task means that user supervision introduces a strong learning bias, and remains a drawback of these approaches.

Completely unsupervised approaches have appeared recently. Hassan et. al. [10] propose to obtain relation extraction patterns from POS n-grams in the corpus, and use an iterative procedure based on graph mutual reinforcement

1550-4786/09 $26.00 © 2009 IEEE
DOI 10.1109/ICDM.2009.81

to find the confidence of both the extraction patterns and the extracted tuples. Nevertheless, being n-gram-based, the approach suffers from a lack of flexibility: the patterns contain only POS and entity type tags, and the inclusion of additional information would lead to a combinatorial explosion in the number of n-grams, making the approach unfeasible.

Clustering techniques have already been used for unsupervised Relation Detection. In the approach of Hasegawa et. al. [11], for every pair of entities of two given types, their accumulated context is found as a bag of all the words appearing between the entities in all their co-occurrences in a corpus. The entity pairs are then clustered using cosine distance between their accumulated contexts. The obtained clusters contain pairs of entities linked by the same kind of relation.

Shinyama and Sekine [12] also propose a multi-level clustering approach for unsupervised Relation Detection. After web crawling, news articles are clustered to form basic clusters, which contain articles from different sources talking about the same news. From the entities in these clusters sets of basic patterns are extracted by considering all paths satisfying a set of constraints from a graph representation of the semantic structure of the sentences in which they occur. The mappings of entities between basic clusters are clustered according to their basic patterns, forming meta-clusters. Again, each meta-cluster contains pairs of entities linked by the same kind of relation.

However, on the contrary of our approach, neither of these clustering-based approaches produces models useful to find relations in data other than the training corpus. The new approach is described in the following section.

III. Approach

Our approach is based on the transformation of Relation Detection between entities of two given types, T_1 and T_2, into a binary classification problem: each pair of entities E_1 and E_2 of the proposed types co-occurring in the same sentence has to be classified as related or unrelated. To classify each pair, we use a two-step scoring-filtering architecture. We take into account the syntactic context of E_1 and E_2 to generate an instance in the form of a feature vector x. A **scorer** is used to calculate the score for this instance, $s(x)$, and a **filterer** assigns it to the related or unrelated class according to whether this score is above or below a relatedness threshold, th_{rel}, respectively.

We propose the use of clustering techniques to learn both the scorer and the filterer, under the following assumptions:

- A clustering provides a point of view of the instances it is grouping.
- The instances in a cluster point to sets of features that are often shared across contexts, and hence may indicate relatedness of entity pairs.
- Each cluster in a clustering has a certain reliability which may be estimated by means of a score.
- In consequence, a scorer can be built from a set of clusterings with their clusters scored: new instances can be assigned a score according to their similarity to each cluster and to the score of these clusters. Each clustering provides a different point of view in this combination.
- In an unsupervised learning approach, not all instances come from related pairs of entities. The most highly scored ones are the most likely to refer to related pairs.
- In consequence, a filterer can be built by determining a threshold value which separates the scores of the pairs likely to be related from those unlikely. New instances can be classified by comparing their score to this threshold value.

Next sections III-A and III-B give a more detailed description of the scorer and filterer learning procedures, respectively.

A. Scorer Learning

The process for the learning of the scorer starts from an unannotated learning corpus, and its goal is to obtain the set of clustering models, Θ_p and the scores, $z(c_{pq})$ for each one of their clusters, c_{pq}, that will make up the scorer.

The learning is performed as follows: after the corpus is pre-processed, all syntactic contexts in which a pair of entities of the given target types T_1 and T_2 occur are extracted and collected as instances. These instances are clustered to obtain a set of clustering models, and every cluster in each model is then scored using a scoring function. A more detailed description of each step follows.

1) Corpus Pre-Processing: The corpus is tokenized and POS-tagged. Entities are recognized, classified and, following Hassan et. al. [10], their heads are replaced with a single token with the entity type as POS tag.

2) Instance Generation: Each pair of entities of the target types T_1 and T_2 co-occurring in the same sentence in the corpus is considered a learning instance. Features are generated from a set of feature patterns which try to capture the syntactic context of the pair. Each instance, x_i, is then a binary vector $(x_{i1} \dots x_{if})$, where x_{ij} tells whether the jth feature is active in the context of instance x_i. The inclusion of more or different kinds of syntactic information can be achieved by changing these feature patterns, a fact that gives flexibility to the model in a simple and efficient way. The feature patterns we used for our experiments are listed in Table I. We have only used distance and POS-based feature patterns as a first step, to allow a fair comparison to other approaches such as [10], which also use only distance and POS tag information (the former implicitly within the n-gram structure).

Given that most of the relations occur at a short distance, pairs of entities further than a threshold value, th_{dist} can

dist_%d	The distance between the words of the pair is %d		
lefty	The leftmost entity of the pair is of type T_1	righty	The rightmost entity of the pair is of type T_1
left_%d_%t	The word %d positions before the leftmost word of the pair has tag %t	lmid_%d_%t	The word %d positions after the leftmost word of the pair has tag %t
rmid_%d_%t	The word %d positions before the rightmost word of the pair has tag %t	right_%d_%t	The word %d positions after the rightmost word of the pair has tag %t

Table I
FEATURE PATTERNS USED

be discarded. Additionally, those features that are active in less instances than a certain threshold, th_{freq}, can be filtered out for efficiency. In our experiments we used a distance threshold th_{dist} of 8 tokens and a feature frequency threshold th_{freq} of 10 instances.

3) Instance Clustering: After generation, the instances are clustered to obtain the clustering models that make up the scorer.

Specifically, probabilistic generative clustering models are used, consisting of a mixture of k components. Each component is a sequence of Bernoulli distributions, one per feature, combined using the Naive Bayes assumption (independence of features given the component):

$$\begin{aligned} p(x_i) &= \sum_{q=1}^{k} p(c_q) \cdot p(x_i \mid c_q) = \sum_{q=1}^{k} \alpha_q \cdot p(x_i \mid c_q) \\ p(x_i \mid c_q) &= \prod_{j=1}^{f} (\vartheta_{qj})^{x_{ij}} \cdot (1 - \vartheta_{qj})^{1-x_{ij}} \end{aligned}$$

where f is the total number of features, α_q is the *a priori* probability of cluster c_q, and ϑ_{qj} is the probability of feature j to be active in an instance generated by cluster c_q. The values $(\alpha_q, \vartheta_{qj})$ are the parameters of the model, Θ, which have to be estimated from data.

This model family is similar to the one used by Nigam et. al. [13] for document classification, but replacing the Multinomial distribution by a Bernoulli distribution, more suitable for binary features as are the ones we are using.

The optimal parameters $\hat{\Theta} = (\hat{\alpha}_q, \hat{\vartheta}_{qj})$ for the model are obtained through Maximum a Posteriori estimation, using the Expectation-Maximization algorithm. As usual, Dirichlet distributions are used as priors for both α_q and ϑ_{qj}.

However, it is known that the models obtained by Expectation-Maximization are sensitive to the number of clusters and to the process of initialization. In order to overcome this limitation, we follow the *massive* approach described by Gonzalez and Turmo [14], who performed a comparison of different strategies of ensemble generation for clustering. A massive strategy, based on randomization of a single algorithm, was found to perform competitively with respect to other strategies and to individual clustering methods for the task of document clustering. Following this *massive* setting, we learn r different clustering models $\hat{\Theta}_p$, with the number of components k_p and the starting conditions for EM chosen at random. The value of k_p is restricted to lay between 2 and a certain k_{max}. Both r and k_{max} are parameters of our method, and we used a value of 100 for both of them in our experiments. These values will generate around 5000 clusters, an amount which gives a reasonable trade-off between computational cost and the ability to capture the different syntactic contexts in which relations may occur.

4) Cluster Scoring: As mentioned at the beginning of this Section III, the clusters in each probabilistic model obtained in the previous step contain syntactic contexts from both related and unrelated pairs of entities. To try to estimate the *quality* of each cluster, we define a cluster scoring function. In this score we take into account both the size of the cluster and the homogeneousness of the instances. Our hypothesis for this decision is that large clusters with instances similar to each other will point to sets of features that are often shared across contexts, and hence may contain related entity pairs.

We start by finding two measures for each cluster: *size* and *homogeneousness*.

The *size* of every cluster, c_{pq}, is found as the sum of the posterior probabilities of each instance, x_i, to belong to it, $size(c_{pq}) = \sum_{x_i} p(c_{pq} \mid x_i)$.

However, when comparing sizes across different models we need a normalization factor, as clusters in models with less components will be larger in average than those in models with more components. Given that after model estimation some components often become irrelevant, specially for models with large number of components k_p, we can define the number of non-empty clusters k_p^{NE} in the model $\hat{\Theta}_p$ as the number of clusters whose size exceeds a certain threshold th_{empty}, $k_p^{NE} = \|c_{pq} \mid size(c_{pq}) > th_{empty}\|$. As the average size of instances in a cluster will be proportional to $1/k_p^{NE}$, product by k_p^{NE} will make cluster sizes comparable across models. For our experiments, the emptiness threshold th_{empty} has been set to 1.

The *homogeneousness* of the instances is estimated by means of a statistically motivated measure, based on the eigenvalue decomposition of the covariance matrix. Given that the principal components of the covariance matrix are the directions in which data variance occurs, and that its corresponding eigenvalues are a measure of the magnitude of this variance, we can take the sum of the eigenvalues of the covariance matrix as a measure of the variance, that is, the *heterogenousness* of the data.

To find this measure, firstly, the empirical feature expectation vector E_j^{pq} and covariance matrix $V_{jj'}^{pq}$ are found for each cluster c_{pq}:

$$\begin{aligned} E_j^{pq} &= \frac{\sum_{x_i} p(c_{pq} \mid x_i) \cdot x_{ij}}{\sum_{x_i} p(c_{pq} \mid x_i)} \\ V_{jj'}^{pq} &= \frac{\sum_{x_i} p(c_{pq} \mid x_i) \cdot (x_{ij} - E_j^{pq}) \cdot (x_{ij'} - E_{j'}^{pq})}{\sum_{x_i} p(c_{pq} \mid x_i)} \end{aligned}$$

The eigenvalue decomposition of the covariance matrix is found, and the sum of its eigenvalues is then taken, as mentioned, as a measure of the *heterogeneousness* of the elements within the cluster. We shall call this result the *radius* of the cluster.

From these two metrics, we can define different scores $z(c_{pq})$ for the clusters c_{pq}, trying to reward both large and homogeneous clusters. We considered three of them for our experiments: Normalized size (NSIZ, as $k_p^{NE} \cdot size(c_{pq})$), Inverse radius (RAD, as $1/radius(c_{pq})$), and Normalized density (NDNS), as the quotient of normalized size and radius, $k_p^{NE} \cdot size(c_{pq})/radius(c_{pq})$). Experiments on their suitability are described in Section IV-C.

B. Filterer Learning

The filterer implements a simple boundary classifier: those instances whose score exceeds a threshold value are considered related, whereas those below the threshold are considered unrelated. The filterer learning process consists hence in determining this threshold.

The threshold value is inferred from the distribution of scores in the training corpus. Hence, as a first step, all instances in the learning corpus are scored and ranked using the newly built scorer. The score $s(x_i)$ for an instance x_i is computed as the sum of the scores of each cluster, c_{pq}, in each model, Θ_p, weighted by the posterior probability of the instance to belong to the cluster:

$$s(x_i) = \sum_{\hat{\Theta}_p} \sum_{q=1}^{k_p} p(c_{pq} \mid x_i) \cdot z(c_{pq})$$

After scoring and ranking, the sequence of scores of the ranked instances is empirically found to follow the shape of a decreasing convex function (there is a small number of highly scored instances, and a large number of lowly scored ones). Our choice for the threshold value is the point of maximum compression of the instance set. We seek thus to select a threshold value such that it simultaneously *maximizes* the accumulated sum of scores of the selected instances and *minimizes* the number of selected instances. The point that fulfills these restrictions will be a maximum convexity point in the sequence.

As an efficient approximate way to calculate this point, we propose to consider the normalized plot of score against rank, with both axes normalized to the range $[0, 1]$, and take the instance which is closest to the origin as cut-off point.

$$\begin{aligned} th_{rel} &= s(x_{i_{rel}}) \\ i_{rel} &= \arg\min_i \sqrt{\hat{s}(x_i)^2 + (i/\max i)^2} \\ \hat{s}(x_i) &= \frac{s(x_i) - \min s(x_i)}{\max s(x_i) - \min s(x_i)} \end{aligned}$$

The score of this instance is then taken as relatedness threshold th_{rel}.

IV. EXPERIMENTATION

To evaluate the validity of our approach, we applied it in an actual relation extraction task, the Relation Detection and Recognition (RDR) task of the ACE evaluation, whose details are given in Section IV-A. We compared our method with two other approaches.

The first one is an implementation of the method of Hassan et. al. [10]. We chose this method because it learns a set of patterns that can be applied on a test corpus different from the training one, and thus allows for evaluation in Relation Extraction tasks. Although other methods such as Hasegawa et. al. [11] or Shinyama and Sekine [12] are also clustering-based and would offer chances for a comparison, they extract tables of related entities from their training corpora and hence do not allow a direct comparison on a different test corpus.

The second is a version of our own scoring-filtering approach using a single clustering model, whose number of clusters and starting point were determined using the Akaike Information Criterion. Comparison to this method will allow us to validate the effectivity of the massive combination.

The experimental data, setting and results are detailed in the following sections.

A. Evaluation Data

As learning corpus we used the year 2000 subset of the Associated Press section of the AQUAINT Corpus. The considered data set contains almost 29 million words from newswire data. We will refer to this corpus as APW.

As mentioned, as test corpus we used data from the Relation Detection and Recognition task of the ACE evaluation. Specifically, we used the training data of ACE evaluations for years 2003, 2004 and 2008. The corpus adds up to over half million words, in which 98,009 entities and 18,322 binary relations between them are annotated. Given that we are evaluating the task of Relation Detection, information relevant to Relation Recognition such as relation types was discarded. Moreover, we approach the task at mention level,

		Rec	Prc	F1
Grams-Ub	-	43.5	65.6	**51.0**
Single	NSiz	52.8	54.3	**52.3**
Single	Rad	52.1	54.2	**50.3**
Single	NDns	53.4	54.1	**52.5**
Mass	NSiz	59.5	53.7	**55.8**
Mass	Rad	62.8	51.7	**56.0**
Mass	NDns	59.1	54.2	**55.9**

Table II
Average Results in the Ace Corpus

as the issues of Relation and Entity coreference are not taken into account, so the task is strictly Relation Mention Detection in ACE terminology.

The gold entities were kept for the test in Ace, whereas entities in Apw were automatically recognized using the BIOS suite[1], trained on Ace. A total of 4,544,830 entities were recognized.

B. Experimental Setup

We considered 11 entity type pairs among the most frequently related in the Ace corpus for evaluation, including the two entity type pairs that Hassan et. al. [10] used for evaluation in their paper, Gpe-Per and Org-Per.

The usual metrics of Precision, Recall and F1 measure on the detected relations are used to evaluate the performance of the proposed approaches.

For all approaches subject to random initialization, five runs were performed for each experiments, and the presented results are the average of the results across all runs.

C. Results

Table II presents the average values for Recall, Precision and F1 for the tested approaches. Results for the approach of Hassan et. al. [10] are listed as Grams-UB; results for our scoring-filtering approach with a single clustering are listed as Single; and finally results for the full massive scoring-filtering approach presented in Section III are listed as Mass. Additionally, the results for Single and Mass are detailed by the cluster scoring function used.

Given that the authors of Hassan et. al. [10] do not provide a criterion to determine the optimal number of patterns to be taken, the results shown for this method are those giving the maximum F1 measure, that is, its upper bound. This gives Grams-UB an advantage with respect to Single and Mass, which has to be kept in mind when interpreting these results.

However, it can be seen from the results that, in average, the behaviour in terms of F1 measure of Mass is better than that of Single, which in turn is better than that of Grams-UB.

	Grams-UB			Mass-NSiz		
	Rec	Prc	F1	Rec	Prc	F1
Fac-Gpe	55.2	68.8	61.3	54.4	73.0	**62.3**
Fac-Loc	27.1	60.9	37.5	61.5	61.0	**61.3**
Fac-Per	23.3	51.8	32.1	37.1	42.7	**39.7**
Gpe-Loc	54.8	73.9	62.9	72.4	59.7	**65.4**
Gpe-Org	73.5	60.7	**66.5**	72.8	60.9	66.3
Gpe-Per	51.6	72.4	**60.2**	60.1	56.6	58.3
Gpe-Veh	51.0	67.5	**58.1**	75.1	46.9	57.8
Loc-Per	27.6	52.3	36.1	44.8	38.1	**41.2**
Org-Per	70.3	53.9	**61.0**	67.8	55.1	60.8
Org-Veh	46.8	89.8	61.5	71.1	61.8	**65.8**
Per-Veh	24.4	57.5	34.3	45.1	36.0	**40.0**

Table III
Results detailed by Pair on the Ace Corpus

We think that these are excellent results, given that we have been able to build an unsupervised Relation Detection system which gives a 4 point increase in F1 with respect to the upper bound of a state-of-the-art approach. Additionally, as mentioned in Section III-A2, our model is more flexible and allows for easier integration of richer information.

It can also be seen that the Mass method tends to produce results with a slight bias for Recall, the results for Single are quite balanced, and Grams-UB clearly favours Precision over Recall.

With respect to the cluster scoring functions, the behaviour of the three is quite similar. Only Rad when applied within Single gives lower F1 values than the other two, and with Mass the results are all within 0.1 points of each other. Given that there is no relevant difference in the performance of three functions, and that the calculation of the radius of the clusters involved for Rad and NDns has a non-neglectable computational cost, we decide to choose the cheaper NSiz for the rest of the comparisons.

Table III contains a comparison of the performance of Grams-UB and Mass with NSiz across the different entity type pairs. As it can be seen, method Mass gives better F1 measures for most of the considered pairs of entity types. For pairs Gpe-Org, Gpe-Per, Gpe-Veh and Org-Per, Grams-UB is better, but in no case by more than 2 points. On the contrary, in the case of Fac-Loc, Grams-UB is some dramatic 24 points below Mass. It can also be observed how, as mentioned before, Grams-UB tends to favour Precision over Recall, whereas Mass behaves the opposite way. There are exceptions, however, in both cases.

D. Results on Filterer Learning

The filterer learning procedure described in Section III-B can be performed on the corpus on which the scorer was learnt or can use a different one. We also ran a series of experiments performing this learning process on the test data. Table IV contains the F1 values obtained by applying the Mass method with the NSiz cluster score using the original

[1] http://www.surdeanu.name/mihai/bios/

	MASS-NSIZ		
	TRAIN	TEST	BEST
FAC-GPE	62.3	**66.8**	67.6
FAC-LOC	**61.3**	61.2	62.6
FAC-PER	39.7	**42.3**	43.3
GPE-LOC	**65.4**	63.5	67.2
GPE-ORG	**66.3**	61.1	72.3
GPE-PER	**58.3**	56.8	59.7
GPE-VEH	**57.8**	56.1	62.3
LOC-PER	41.2	**42.1**	43.9
ORG-PER	**60.8**	60.3	62.8
ORG-VEH	**65.8**	61.9	69.8
PER-VEH	40.0	**42.5**	42.8

Table IV
F1 SCORES ON THE ACE CORPUS

(TRAIN) and modified (TEST) filterer learning procedures, detailed by pair. In addition, the maximum achievable results (BEST) are also listed.

The results in the table show that, despite in some cases, such as for FAC-GPE and FAC-PER, the change of the TRAIN data by the TEST ones can improve the performance of the system, in most cases the F1 values do not change considerably, and in some cases, such as for GPE-ORG and ORG-VEH, they can impair by more than 4 points.

The results also show that the simple threshold determination criterion proposed in Section III-B works reasonably well for most of the cases, with TRAIN falling within 2 points of the BEST achievable F1 measure. However, it is also true that in some cases, such as for FAC-GPE or FAC-PER, the result is 5 points below the best one.

V. CONCLUSIONS

This paper proposes a new unsupervised approach to learning for relation extraction, using probabilistic clustering models.

We have compared the proposed approach to a state-of-the-art unsupervised system on data from the ACE Relation Mention Detection task. Our approach obtains a F1 measure of 55.7, more than 4 points above the upper bound of 51.0 attainable by the other system, with both using only POS information. Besides, it is more flexible and allows the inclusion of richer features.

Additionally, we have shown that learning using a massive combination of clusterings improves the performance of the scorer, with respect to a learner based on a single clustering model and a model selection criterion. We have also proposed several cluster score functions, and we have proved that the method is robust to its choice.

We think that our approach can be considered a powerful learning method for relation extraction, given its simplicity, flexibility, efficiency, non-supervision and improved performance with respect to the state-of-the-art.

ACKNOWLEDGMENTS

Work partially funded by the KNOW2, TIN2009-14715-C04-04, and the TEXT-MESS, TIN2006-15265-C06, projects.

REFERENCES

[1] J. Turmo, A. Ageno, and N. Català, "Adaptive information extraction," *ACM Computing Surveys*, vol. 38, pp. 1–47, 2006.

[2] E. Dimitriadou, "Exploratory data analysis and applications," Ph.D. dissertation, Technische Universität Wien, 2003.

[3] "Automatic Content Extraction (ACE) Evaluation," 2009, http://www.itl.nist.gov/iad/mig/tests/ace/.

[4] N. Kambhatla, "Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2004.

[5] S. Zhao and R. Grishman, "Extracting relations with integrated information using kernel methods," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005, pp. 419–426.

[6] M. Surdeanu and M. Ciaramita, "Robust information extraction with perceptrons," in *NIST Automatic Content Extraction Workshop (ACE)*, 2007.

[7] S. Brin, "Extracting patterns and relations from the World-Wide Web," in *International Workshop on the Web and Databases (WebDB)*, 1998.

[8] E. Agichtein and L. Gravano, "Snowball: Extracting relations from large plain-text collections," in *ACM Conference on Digital Libraries (DL)*, 2000, pp. 85–94.

[9] S. Sekine, "On-demand information extraction," in *International Meeting of the Committee on Computational Linguistics and the Association for Computational Linguistics (COLING-ACL)*, 2006, pp. 731–738.

[10] H. Hassan, A. Hassan, and O. Emam, "Unsupervised information extraction approach using graph mutual reinforcement," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2006, pp. 501–508.

[11] T. Hasegawa, S. Sekine, and R. Grishman, "Discovering relations among named entities from large corpora," in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2004.

[12] Y. Shinyama and S. Sekine, "Preemptive information extraction using unrestricted relation discovery," in *Human Language Technology Conference and North American chapter of the Association of Computational Linguistics Anual Meeting (HLT-NAACL)*, 2006, pp. 304–311.

[13] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Machine Learning*, vol. 39, no. 2/3, pp. 1–34, 2000.

[14] E. Gonzàlez and J. Turmo, "Comparing non-parametric ensemble methods for document clustering," in *Natural Language and Information Systems (NLDB)*, 2008, pp. 245–256.