Graph Based Semi-Supervised Approach for Information Extraction

Hany Hassan   **Ahmed Hassan**   **Sara Noeman**

IBM Cairo Technology Development Center
Giza, Egypt
P.O. Box 166 Al-Ahram

hanyh@eg.ibm.com   hasanah@eg.ibm.com   noemans@eg.ibm.com

Abstract

Classification techniques deploy supervised labeled instances to train classifiers for various classification problems. However labeled instances are limited, expensive, and time consuming to obtain, due to the need of experienced human annotators. Meanwhile large amount of unlabeled data is usually easy to obtain. Semi-supervised learning addresses the problem of utilizing unlabeled data along with supervised labeled data, to build better classifiers. In this paper we introduce a semi-supervised approach based on mutual reinforcement in graphs to obtain more labeled data to enhance the classifier accuracy. The approach has been used to supplement a maximum entropy model for semi-supervised training of the ACE Relation Detection and Characterization (RDC) task. ACE RDC is considered a hard task in information extraction due to lack of large amounts of training data and inconsistencies in the available data. The proposed approach provides 10% relative improvement over the state of the art supervised baseline system.

1   Introduction

Classification techniques use labeled data to train classifiers for various classification problems. Yet they often face a shortage of labeled training data. Labeled instances are often difficult, expensive, and /or time consuming to obtain. Meanwhile large numbers of unlabeled instances are often available. Semi-supervised learning addresses the problem of how unlabeled data can be usefully employed, along with labeled data, to build better classifiers.

In this paper we propose a semi-supervised approach for acquiring more training instances similar to some labeled instances. The approach depends on constructing generalized extraction patterns, which could match many instances, and deploying graph based mutual reinforcement to weight the importance of these patterns. The mutual reinforcement is used to automatically identify the most informative patterns; where patterns that match many instances tend to be correct. Similarly, instances matched by many patterns also tend to be correct. The labeled instances should have more effect in the mutual reinforcement weighting process. The problem can therefore be seen as hubs (instances) and authorities (patterns) problem which can be solved using the Hypertext Induced Topic Selection (HITS) algorithm (Kleinberg, 1998).

HITS is an algorithmic formulation of the notion of authority in web pages link analysis, based on a relationship between a set of relevant "authoritative pages" and a set of "hub pages". The HITS algorithm benefits from the following observation: when a page (hub) links to another page (authority), the former confers authority over the latter.

By analogy to the authoritative web pages problem, we could represent the patterns as authorities and instances as hubs, and use mutual reinforcement between patterns and instances to weight the most authoritative patterns. Instances from unsu-

9

pervised data matched with the highly weighted patterns are then used in retraining the system.

The paper proceeds as follows: in Section 2 we discuss previous work followed by a brief definition of our general notation in Section 3. A detailed description of the proposed approach then follows in Section 4. Section 5 discusses the application of the proposed approach to the problem of detecting semantic relations from text. Section 6 discusses experimental results while the conclusion is presented in Section 7.

2   Previous Work

(Blum and Mitchell, 1998) proposed an approach based on co-training that uses unlabeled data in a particular setting. They exploit the fact that, for some problems, each example can be described by multiple representations. They develop a boosting scheme which exploits conditional independence between these representations.

(Blum and Chawla, 2001) proposed a general approach utilizing unlabeled data by constructing a graph on all the data points based on distance relationships among examples, and then to use the known labels to perform a graph partitioning using the minimum cut that agrees with the labeled data. (Zhu et al., 2003) extended this approach by proposing a cut based on the assumption that labels are generated according to a Markov Random Field on the graph , (Joachims, 2003) presented an algorithm based on spectral graph partitioning. (Blum et al., 2004) extended the min-cut approach by adding randomness to the graph structure, their algorithm addresses several shortcomings of the basic mincut approach, yet it may not help in cases where the graph does not have small cuts for a given classification problem.

3   Background

In graph theory, a graph is a set of objects called vertices joined by links called edges. A bipartite graph, also called a bigraph, is a special graph where the set of vertices can be divided into two disjoint sets with no two vertices of the same set sharing an edge.

The Hypertext Induced Topic Selection (HITS) algorithm is an algorithm for rating, and therefore ranking, web pages. The HITS algorithm makes use of the following observation: when a page (hub) links to another page (authority), the former confers authority over the latter. HITS uses two values for each page, the *"authority value"* and the *"hub value"*. "*Authority value*" and "*hub value*" are defined in terms of one another in a mutual recursion. An authority value is computed as the sum of the scaled hub values that point to that authority. A hub value is the sum of the scaled authority values of the authorities it points to.

A *template*, as we define for this work, is a sequence of generic forms that could generalize over the given training instance. An example template is:
COUNTRY   NOUN_PHRASE   PERSON VERB_PHRASE
This template could represent the sentence:
    "American vice President Al Gore visited ...".
This template is derived from the representation of the Named Entity tags, Part-of-Speech (POS) tags and semantic tags. The choice of the template representation here is for illustration purpose only; any combination of tags, representations and tagging styles might be used.

A *pattern* is more specific than a template. A pattern specifies the role played by the tags (first entity, second entity, or relation). An example of a pattern is:
COUNTRY(E2) NOUN_PHRASE(R) PERSON(E1) VERB_PHRASE
This pattern indicates that the word(s) with the tag *COUNTRY* in the sentence represents the second entity (Entity 2) in the relation, while the word(s) tagged *PERSON* represents the first entity (Entity 1) in this relation. Finally, the word(s) with the tag *NOUN_PHRASE* represents the relation between the two previous entities.

A *tuple*, in our notation during this paper, is the result of the application of a pattern to unstructured text. In the above example, one result of applying the pattern to some raw text is the following tuple:
Entity 1: Al Gore
Entity 2: United States
Relation: vice President

4   The Approach

The semi-supervised graph-based approach we propose depends on the construction of generalized extraction patterns that could match many training instances. The patterns are then weighted according to their importance by deploying graph based

mutual reinforcement techniques. Patterns derived from the supervised training instances should have a superior effect in the reinforcement weighting process. This duality in patterns and tuples relation could be stated that patterns could match different tuples, and tuples in turn could be matched by different patterns. The proposed approach is composed of two main steps namely, pattern extraction and pattern weighting or induction. Both steps are detailed in the next subsections.

4.1 Patterns Extraction

As shown in Figure 1, several syntactic, lexical, and semantic analyzers could be applied to the training instances. The resulting analyses could be employed in the construction of extraction patterns. Any extraction pattern could match different relations and hence could produce several tuples. As an example let's consider the pattern depicted in figure 1:
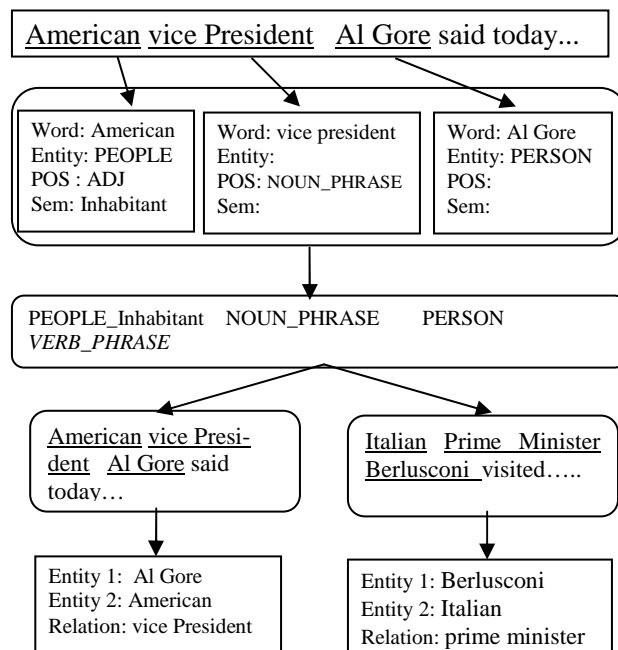
Figure 1: An example of a pattern and its possible tuples.

PEOPLE_Inhabitant(E2)   NOUN_PHRASE(R)
PERSON(E1) VERB_PHRASE
This pattern could extract the tuple:
Entity 1: Al Gore
Entity 2: American
Relation: vice President

Another tuple that could be extracted by the same pattern is:
Entity 1: Berlusconi
Entity 2: Italian
Relation: Prime Minister
On the other hand, many other patterns could extract the same information in the tuple from different contexts. It is worth mentioning that the proposed approach is general enough to accommodate any pattern design; the introduced pattern design is for illustration purposes only.

To further increase the number of patterns that could match a single tuple, the tuple space might be reduced i.e. by grouping tuples conveying the same information content together into a single tuple. This will be detailed further in the experimental setup section.

4.2 Pattern Induction

The inherent duality in the patterns and tuples relation suggests that the problem could be interpreted as a hub authority problem. This problem could be solved by applying the HITS algorithm to iteratively assign authority and hub scores to patterns and tuples respectively.
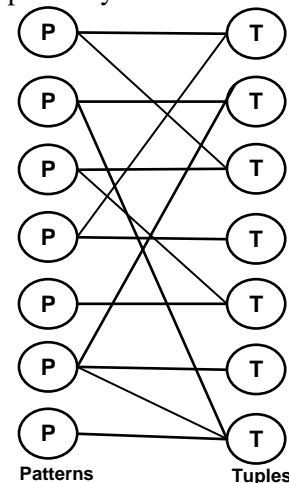
Figure 2: A bipartite graph representing patterns and tuples

Patterns and tuples are represented by a bipartite graph as illustrated in figure 2. Each pattern or tuple is represented by a node in the graph. Edges represent matching between patterns and tuples.

The pattern induction problem can be formulated as follows: Given a very large set of data *D* containing a large set of patterns *P* which match a

11

large set of tuples T, the problem is to identify $\widetilde{P}$, the set of patterns that match the set of the most correct tuples $\widetilde{T}$. The intuition is that the tuples matched by many different patterns tend to be correct and the patterns matching many different tuples tend to be good patterns. In other words; we want to choose, among the large space of patterns in the data, the most informative, highest confidence patterns that could identify correct tuples; i.e. choosing the most *"authoritative"* patterns in analogy with the hub authority problem. However, both $\widetilde{P}$ and $\widetilde{T}$ are unknown. The induction process proceeds as follows: each pattern p in P is associated with a numerical authority weight a_v which expresses how many tuples match that pattern. Similarly, each tuple t in T has a numerical hub weight h_t which expresses how many patterns were matched by this tuple. The weights are calculated iteratively as follows:

$$a^{(i+1)}(p) = \sum_{u=1}^{T(p)} \frac{h^{(i)}(u)}{H^{(i)}} \qquad (1)$$

$$h^{(i+1)}(t) = \sum_{u=1}^{P(t)} \frac{a^{(i)}(u)}{A^{(i)}} \qquad (2)$$

where $T(p)$ is the set of tuples matched by p, $P(t)$ is the set of patterns matching t, $a^{(i+1)}(p)$ is the authoritative weight of pattern p at iteration $(i+1)$, and $h^{(i+1)}(t)$ is the hub weight of tuple t at iteration $(i+1)$. $H(i)$ and $A(i)$ are normalization factors defined as:

$$H^{(i)} = \sum_{p=1}^{|P|} \sum_{u=1}^{T(p)} h^{(i)}(u) \qquad (3)$$

$$A^{(i)} = \sum_{v=1}^{|T|} \sum_{u=1}^{P(t)} a^{(i)}(u) \qquad (4)$$

Patterns with weights lower than a predefined threshold are rejected, and examples associated with highly ranked patterns are then used in unsupervised training.

It is worth mentioning that both *T* and *P* contain supervised and unsupervised examples, however the proposed method could assign weights to the correct examples (tuples and patterns) in a completely unsupervised setup. For semi-supervised data some supervised examples are provided, which are associated in turn with tuples and patterns.

We adopt the HITS extension introduced in (White and Smyth, 2003) to extend HITS with *Priors*. By analogy, we handle the supervised examples as priors to the HITS induction algorithm.

A prior probabilities vector $pr = \{pr_1, \ldots, pr_n\}$ is defined such that the probabilities sum to 1, where pr_v denotes the relative importance (or "prior bias") we attach to node *v*. A pattern P_i is assigned a prior $pr_i = 1/n$ if pattern P_i matches a supervised tuple, otherwise *pr_i is set to zero, n* is the total number of patterns that have a supervised match. We also define a *"back probability" β, $0 \le \beta \le 1$* which determines how often we bias the supervised nodes:

$$a^{(i+1)}(p) = (1-\beta)\left(\sum_{u=1}^{T(p)} \frac{h^{(i)}(u)}{H^{(i)}}\right) + \beta * pr_p \quad (5)$$

$$h^{(i+1)}(t) = (1-\beta)\left(\sum_{u=1}^{P(t)} \frac{a^{(i)}(u)}{A^{(i)}}\right) + \beta * pr_t \qquad (6)$$

where $T(p)$ is the set of tuples matched by p, $P(t)$ is the set of patterns matching *t*, and *H(i)* and *A(i)* are normalization factors defined as in equations (3) and (4)

Thus each node in the graph (pattern or tuple) has an associated prior weight depending on its supervised data. The induction process proceeds to iteratively assign weights to the patterns and tuples. In the current work we used $\beta = 0.5$.

5 Experimental Setup

5.1 ACE Relation Detection and Characterization

In this section, we describe Automatic Content Extraction (ACE). ACE is an evaluation conducted by NIST to measure Entity Detection and Tracking (EDT) and Relation Detection and Characterization (RDC). The EDT task is concerned with the detection of mentions of entities, and grouping them together by identifying their coreference. The RDC task detects relations between entities identified by the EDT task. We choose the RDC task to show the performance of the graph based semi-supervised information extraction approach we propose. To this end we need to introduce the notion of mentions and entities. Mentions are any instances of textual references to objects like peo-

ple, organizations, geo-political entities (countries, cities …etc), locations, or facilities. On the other hand, entities are objects containing all mentions to the same object.

Type	Subtype	Number of Instances
ART	User-Owner	331
	Inventor	
	Other	
DISC	DISC	143
EMP-ORG	Employ-Exec	1673
	Employ-Staff	
	Employ-Undetermined	
	Member-of-Group	
	Subsidiary	
	Other	
Other-AFF	Ethnic	153
	Ideology	
	Other	
GPE-AFF	Citizen-Resident	695
	Based-in	
	Other	
PER-SOC	Business	358
	Family	
	Other	
PHYS	Located	1411
	Near	
	Part-Whole	

Table 1. Types and subtypes of ACE relations

Table 1 lists the types and subtypes of relations for the ACE RDC task. Here, we present an example for those relations:

```
Spain's Interior Minister an-
nounced this evening the ar-
rest of separatist
organization Eta's presumed
leader Ignacio Garcia Ar-
regui. Arregui, who is con-
sidered to be the Eta
organization's top man, was
arrested at 17h45 Greenwich.
The Spanish judiciary sus-
pects Arregui of ordering a
failed attack on King Juan
Carlos in 1995.
```

In this fragment, all the underlined phrases are mentions to *Eta* organization, or to *"Garcia Arregui"*. There is a management relation between *leader* which references to *"Garcia Arregui"* and *Eta*.

5.2 Baseline System

The base line system uses a Maximum Entropy model that combines diverse lexical, syntactic and semantic features derived from text, like the system described in (Nanda, 2004). The system was trained on the ACE training data provided by LDC. The training set contained 145K words, and 4764 instances of relations, the number of instances corresponding to each relation is shown in Table 1.

The test set contained around 41K words, and 1097 instances of relations. The system was evaluated using standard ACE evaluation procedure. ACE evaluation procedure assigns the system an ACE value for each relation type and a total ACE value. The ACE value is a standard NIST metric for evaluating relation extraction. The reader is referred to the ACE web site (ACE, 2004) for more details.

5.3 Pattern Construction

We used the baseline system described in the previous section to label a large amount of unsupervised data. The data comes from LDC English Gigaword corpus, Agence France Press English Service (AFE). The data contains around 3M words, from which 80K instances of relations have been extracted.

We start by extracting a set of patterns that represent the supervised and unsupervised data. We consider each relation type separately and extract a pattern for each instance in the selected relation. The pattern we used consists of a mix between the part of speech (POS) tags and the mention tags for the words in the training instance. We use the mention tag, if it exists; otherwise we use the part of speech tag. An example of a pattern is:

```
Text: Eta's presumed leader
Arregui …
Pos: NNP POS JJ NN NNP
Mention: ORG 0 0 0 PERSON
Pattern: ORG(E2) POS JJ NN(R)
PERSON(E1)
```

5.4 Tuples Clustering

As discussed in the previous section, the tuple space should be reduced to allow more matching between pattern-tuple pairs. This space reduction could be accomplished by seeking a tuple similarity measure, and constructing a weighted undirected graph of tuples. Two tuples are linked with an edge if their similarity measure exceeds a certain threshold. Graph clustering algorithms could be deployed to partition the graph into a set of homogeneous communities or clusters. To reduce the space of tuples, we seek a matching criterion that group similar tuples together. Using WordNet, we can measure the semantic similarity or relatedness between a pair of concepts (or word senses), and by extension, between a pair of sentences. We use the similarity measure described in (Wu and Palmer, 1994) which finds the path length to the root node from the least common subsumer (LCS) of the two word senses which is the most specific word sense they share as an ancestor. The similarity score of two tuples, S_T, is calculated as follows:.

$$S_T = \sqrt{S_{E1}^2 + S_{E2}^2} \qquad (9)$$

where S_{E1}, and S_{E2} are the similarity scores of the first entities in the two tuples, and their second entities respectively.

The tuple matching procedure assigns a similarity measure to each pair of tuples in the dataset. Using this measure we can construct an undirected graph G. The vertices of G are the tuples. Two vertices are connected with an edge if the similarity measure between their underlying tuples exceeds a certain threshold. It was noticed that the constructed graph consists of a set of semi isolated groups as shown in figure 3. Those groups have a very large number of inter-group edges and meanwhile a rather small number of intra-group edges. This implies that using a graph clustering algorithm would eliminate those weak intra-group edges and produce separate groups or clusters representing similar tuples. We used Markov Cluster Algorithm (MCL) for graph clustering (Dongen, 2000). MCL is a fast and scalable unsupervised cluster algorithm for graphs based on simulation of stochastic flow.

A bipartite graph of patterns and tuple clusters is constructed. Weights are assigned to patterns and tuple clusters by iteratively applying the HITS with Priors' algorithm. Instances associated with highly ranked patterns are then added to the training data and the model is retrained. Samples of some highly ranked patterns and corresponding matching text are introduced in Table 2.
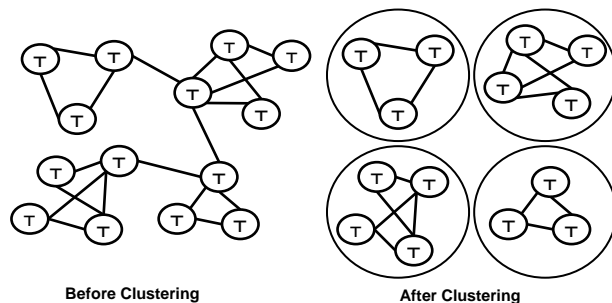

Before Clustering   **After Clustering**

Figure 3: Applying Clustering Algorithms to Tuple graph

Pattern	Matches
GPE PERSON PERSON PERSON	Zimbabwean President Robert Mugabe
GPE POS PERSON PERSON	Zimbabwe 's President Robert Mugabe
GPE JJ PERSON	American diplomatic personnel
PERSON IN JJ GPE	candidates for local government
ORGANIZATION PERSON	Airways spokesman
ORGANIZATION PERSON	Ajax players
PERSON IN DT JJ ORGANIZATION	chairman of the opposition parties
ORGANIZATION PERSON	parties chairmans

Table 2: Examples of patterns with high weights

6 Results and Discussion

We train several models like the one described in section 5.2 on different training data sets. In all experiments, we use both the LDC ACE training data and the labeled unsupervised data induced with the graph based approach we propose. We use the ACE evaluation procedure and ACE test corpus, provided by LDC, to evaluate all models.

We incrementally added labeled unsupervised data to the training data to determine the amount of data after which degradation in the system performance occurs. We sought this degradation point separately for each relation type. Figure 4 shows the effect of adding labeled unsupervised data on

the ACE value for each relation separately. We notice from figure 4 and table 1 that relations with a small number of training instances had a higher gain in performance compared to relations with a large number of training instances. This implies that the proposed approach achieves significant improvement when the number of labeled training instances is small but representative.
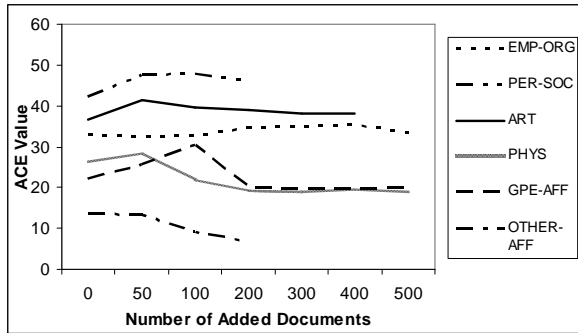

Figure 4: The effect of adding labeled unsupervised data on the ACE value for each relation. The average number of relations per document is 4.

From figure 4, we determined the number of training instances resulting in the maximum boost in performance for each relation. We added the training instances corresponding to the maximum boost in performance for all relations to the supervised training data and trained a new model on them. Figure 5 compares the ACE values for each relation in the base line model and the final model

The total system ACE value has been improved by 10% over the supervised baseline system. All relation types, except the DSC relation, had significant improvement ranging from 7% to 30% over the baseline supervised system. The DISC relation type had a small degradation; noting that it already has a low ACE value with the baseline system. We think this is due to the fact that the DISC relation has few and inconsistent examples in the supervised data set.

To assess the usefulness of the smoothing method employing WordNet distance, we repeated the experiment on EMP-ORG relation without it. We found out that it contributed to almost 30% of the total achieved improvement. We also repeated the experiment but with considering hub scores instead of authority scores. We added the examples associated with highly ranked tuples to the training set. We noticed that using hub scores yielded very

little variation in the ACE value (i.e. 0.1 point for EMP-ORG relation).

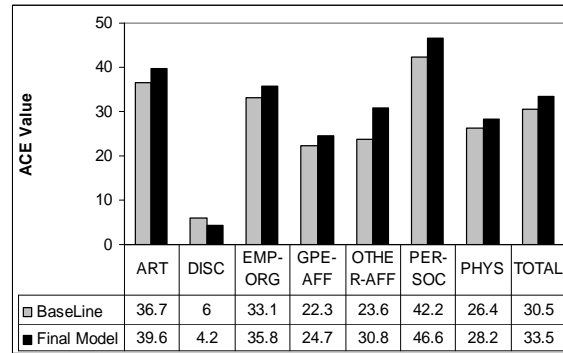|	ART	DISC	EMP-ORG	GPE-AFF	OTHER-AFF	PER-SOC	PHYS	TOTAL
BaseLine	36.7	6	33.1	22.3	23.6	42.2	26.4	30.5
Final Model	39.6	4.2	35.8	24.7	30.8	46.6	28.2	33.5

Figure 5: A comparison of base line ACE values, and final ACE values for each relation.

To evaluate the quality and representativeness of the labeled unsupervised data, acquired using the proposed approach, we study the effect of replacing supervised data with unsupervised data while holding the amount of training data fixed. Several systems have been built using mixture of the supervised and the unsupervised data. In Figure 6, the dotted line shows the degradation in the system performance when using a reduced amount of supervised training data only, while the solid line shows the effect of replacing supervised training data with unsupervised labeled data on the system performance. We notice from Figure 6 that the unsupervised data could replace more than 50% of the supervised data without any degradation in the system performance. This is an indication that the induced unsupervised data is good for training the classifier.
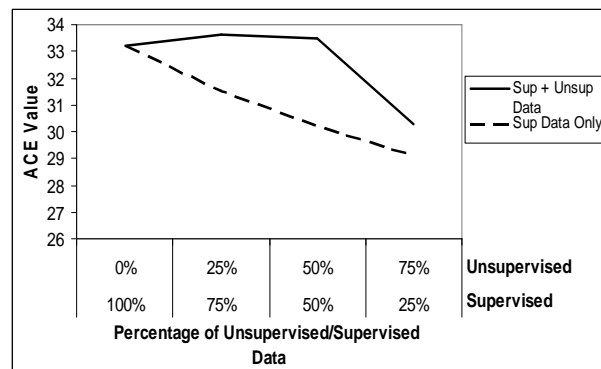

Figure 6: The effect of removing portions of the supervised data on the ACE value. And the effect

of replacing portions of the supervised data with labeled training data.

7 Conclusion

We introduce a general framework for semi-supervised learning based on mutual reinforcement in graphs. We construct generalized extraction patterns and deploy graph based mutual reinforcement to automatically identify the most informative patterns. We provide motivation for our approach from a graph theory and graph link analysis perspective.

We present experimental results supporting the applicability of the proposed approach to ACE Relation Detection and Characterization (RDC) task, demonstrating its applicability to hard information extraction problems. Our approach achieves a significant improvement over the base line supervised system especially when the number of labeled instances is small.

8 Acknowledgements

References

ACE. 2004. The NIST ACE evaluation website. http://www.nist.gov/speech/tests/ace/

Avrim Blum, and Tom Mitchell. 1998. *Combining Labeled and Unlabeled data with Co-training*. Proceedings of the 11th Annual Conference on Computational Learning Theory.

Avrim Blum and Shuchi Chawla. 2001. *Learning From Labeled and Unlabeled Data Using Graph Mincuts*. Proceedings of International Conference on Machine Learning (ICML).

Avrim Blum, John Lafferty, Mugizi Rwebangira, and Rajashekar Reddy. 2004. *Semi-supervised Learning Using Randomized Mincuts*. Proceedings of the International Conference on Machine Learning (ICML).

Stijn van Dongen. 2000. *A Cluster Algorithm for Graphs*. Technical Report INS-R0010, National Research Institute for Mathematics and Computer Science in the Netherlands.

Stijn van Dongen. 2000. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht

Radu Florian, Hany Hassan, Hongyan Jing, Nanda Kambhatla, Xiaqiang Luo, Nicolas Nicolov, and Salim Roukos. 2004. *A Statistical Model for multilingual entity detection and tracking*. Proceedings of the Human Language Technologies Conference (HLT-NAACL'04).

Dayne Freitag, and Nicholas Kushmerick. 2000. *Boosted wrapper induction*. The 14th European Conference on Artificial Intelligence Workshop on Machine Learning for Information Extraction

Taher Haveliwala. 2002. *Topic-sensitive PageRank*. Proceedings of the 11th International World Wide Web Conference

Thorsten Joachims. 2003. *Transductive Learning via Spectral Graph Partitioning*. Proceedings of the International Conference on Machine Learning (ICML).

John Kleinberg. 1998. *Authoritative Sources in a Hyperlinked Environment*. Proceedings of the. 9th ACM-SIAM Symposium on Discrete Algorithms.

Nanda Kambhatla. 2004. *Combining Lexical, Syntactic, and Semantic Features with Maximum Entropy Models for Information Extraction*. Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi, 2004, *WordNet::Similarity - Measuring the Relatedness of Concepts*. Proceedings of Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-2004)

Scott White, and Padhraic Smyth. 2003. *Algorithms for Discoveing Relative Importance in Graphs*. Proceedings of Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Zhibiao Wu, and Martha Palmer. 1994. *Verb semantics and lexical selection*. Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. 2003. *Semi-supervised Learning using Gaussian Fields and Harmonic Functions*. Proceedings of the 20th International Conference on Machine Learning.