CS224N Final Report

Aman Neelappa
Deniz Kahramaner

Joint Entity-Event Coreference via Spectral Coclustering

Introduction

Event and Entity co-reference is an important problem in Natural Language Processing. Traditionally in literature these have been viewed as independent problems and various approaches to solving them have been proposed. Recently Lee et al [1], proposed a method of solving both these problems jointly.

Events and Entities within a text are often inter-related such that the knowledge of the co-references of one helps in, and sometimes is essentially for, finding coreference relations of the other. To illustrate this we use a couple of motivating examples from [1]

1. (a) **One of the key suspected Mafia bosses arrested yesterday** has hanged himself.
   (b) Police said **Lo Presti** had hanged himself.
   (c) His suicide appeared to be related to clan feuds.
2. (a) **The New Orleans Saints** placed **Reggie Bush** on the injured list on Wednesday.
   (b) **Saints** put **Bush** on I.R.

As the first example illustrates, it is not obvious that the noun phrases "One of the key suspected Mafia bosses arrested yesterday" and "Lo Presti" corefer unless we look at the associated verb phrases. Similarly, in the second example the verbs "placed" and "put" can be inferred to be corefferent by realizing that the argument noun phrases are coreferent.

Lee et al solve this problem by training a linear regressor that scores the merging of two clusters. The features of the linear regressor includes features that take care of the dependece on arguments in the way the above examples motivate. They then proceed to iteratively merge clusters and update the features as more and more such relations become apparent.

Dhillon[2] proposed co-clustering as an approach for simultaneously finding clusters of words and documents. The problem has striking similarities to the problem of the joint entity-event co-reference. He notes in his paper that a common theme of the algorithms is to cluster documents based on the word-distributions while word-clustering is determined by co-occurence in documents. He further says that dual clustering problem can be posed as the problem of finding the minimum cut vertex partitions in a bi-partite graph between documents and words.

Shi et al [3] extend the above approach to a a semi-supervised setting where some clustering over the vertices is already known from some other source and we wish to find the minimum cuts that do not violate (or violate the least) this clustering. Specifically they model these constraints as a trace minimization problem. Further they show that their algorithm has no significant overheads on top of traditional spectral co-clustering that does not use these constraints. In the next section we detail their approach and how we apply it to the problem of joint event-entity coreference.

Algorithm and Implementation

We extend the approach taken by Shi et al to the problem of Co-clustering events and entities. Here we describe the various algorithms used.

Algorithm : Cross-doc Joint Mention Coclustering

For each set of documents D

1. Find the initial set of clusters by running Deterministic Co-reference sieves on the data. Note that while Lee et al run only the sieves that apply to noun mentions, we run Sieves that find co-reference between both noun and verb mentions independently.
2. Split the mentions into two groups of Event Mentions and Entity mentions using some Partition function.
3. Construct a bipartite graph between the Entities and Mentions using some Linking function that scores the linkages between Entity-Mention pairs.
4. Construct a Constraint Matrix which is basically an adjacency matrix of the mentions as defined by the clusters found by the deterministic sieves.
5. Run the Spectral Constraint Modeling algorithm proposed by Lee et al to get the final clustering of Events and Entities.

We now discuss some of the components in more detail.

Clustering the documents

As [1] notes, it is important that for cross-document coreference resolution. Not only doe this reduce the search space, this provides a word sense disambiguation mechanism. We re-use the document clusters used by the authors of [1] that they were kind enough to provide.

Choice of Mentions

We further choose only those mentions for Joint co-clustering which are already a part of some cluster after the application of the deterministic sieves. This was done partly due to keep things simple and partly because we expected that most "important" mentions (those with multiple co-references) would surely be found to be co-referent with something after the sieves have been applied. Another important reason was that this might help us reduce the sparsity of our Constraint Matrix and bipartite graph. However we note that this restriction is one that can be easily overcome by incorporating mentions that are "singleton" even after the application of the deterministic sieves.

Deterministic Sieves

We use the deterministic sieves that Lee et al use for their Baseline 1 results in [1]. The reason we choose this is because it performs Entity and Event clustering completely independent of each other. The Baseline results are shown in Table 1. As can be seen, it provides a rather good starting set of co-references which we wished to improve by co-clustering

Partition function

For simplicity, we use a naive partition function that classifies mentions which are verb phrases as Events and others as Entities. Note that while this is true for most cases, there are cases where this does not hold true. Such an example is shown in the sentence 1(c) where "suicide", though a noun, represents an Event in the real world. For our analysis we ignore such cases as they are few in number.

Linking function : As the examples given previously indicate the links between the Events and

Entities manifest themselved by means of the Arguments and Predicates of Verb and Nominal/Pronominal mentions. Keeping this in mind, we tried the following Linking functions

1. Simple Argument/Predicate link – The weight of the link is 1.0 if one of the mentions is an argument/predicate of the other.
2. Clustered Argument/Predicate link - The weight of the link between mention A and B is equal to the number of mentions in the (already known) cluster of A which are either arguments/predicates if mentions in the cluster of mention B.
3. Normalized Clustered Argument Predicate links – In this we normalize the above score by the product of the total number of mentions in the cluster of mention A and mention B.
4. Mutual Information based Argument Predicate Links – In this we further normalize the above with the number of of mentions in cluster of A and B with the total number of mentions in that category (Event/Entity)

After defining the above things, the we are now in a position to apply the Spectral Constraint Modeling algorithm which forms the crux of out approach.

Spectral Constraint Modeling(SCM) algorithm

Input: Bipartite edge weight matrix **E**; Constraint matrix **C**; Constraint confident parameter δ; Cluster number *k*
Output: Row partition matrix **X**r; Column partition matrix **X**c

1. Construct the diagonal matrices **Dr**, **Dc** where [**Dr**]ii = $\sum_j$ **E**ij and [**Dc**]ii = $\sum_j$**E**ji
2. Calculate **A** as defined below.
3. Perform Singular Value Decomposition on matrix **A**. Denote the left and right eigenvector of the 2nd to the (k+1)-th eigenvalues as **U** and **V** respectively.
4. Construct **Z**r = (**D**r - δ**Crr**)$^{-1/2}$**U** and **Zc** =(**Dc** – δ**Ccc**)$^{-1/2}$**V**. (**Ccc** and **Crr** defined below)
5. Run k-means on **Zr** to get the row partition matrix **Xr**; similarly get **Xc** from **Zc** (different number of row clusters and column clusters can be set in k-means).

Crr is the submatrix of the constraint matrix **C** which defines the adjacency graph of the objects corresponding to the rows of the bipartite graph **E**. Similarly **Crr** defines the adjaceny matrix of the columns of the bipratite graph. **A** is a matrix defined as

$$A = (\mathbf{D}r - \delta\mathbf{Crr} + \delta\mathbf{Sr})^{-1/2} (\mathbf{E} + \delta\mathbf{Crc}) (\mathbf{Dc} - \delta\mathbf{Ccc} + \delta\mathbf{Sc})^{-1/2}$$

where

$$[\mathbf{Sr}]ii = \sum_j [\mathbf{Crr}]_{ij} + \sum_t [\mathbf{Crc}]_{it}$$

$$[\mathbf{Sc}]ii = \sum_j [\mathbf{Ccc}]_{ij} + \sum_t [\mathbf{Ccr}]_{it}$$

and the matrices **Crc** and **Ccr** are the submatrices which defined the known adjacency relations between the objects in the rows and columns of the **E** matrix. In our case, these matrices are expected to be zero.

There is one further thing to note that while the algorithm given above takes in only one Cluster number, it can easily be modified to accomodate for different number of clusters for row and column objects. We provide two parameters k and l to the algorithm. We modify step 3 by defining matrix **V** as the columns 2 to *l+1* instead of *k +1*. Finally we ask k-means to find l clusters instead of k for **Zc.**

Results and Discussions

Dataset for joint co-reference is not easy to come by. We use the annotated dataset for this task created for [1] which is based on the dataset created for [4].

The results of running are algorithm are presented in Table 1. As can be seen, our model performed significantly worse than the Baseline. Specifically, though the recall score improved slightly, the precision score went down substantially.

	MUC			**B-Cubed**			**BLANC**		
Baseline 1 = B1	R	P	F1	R	P	F1	R	P	F1
Entity Event Both	49.41 56.69 51.5	76.84 72 80.28	60.15 63.43 62.75	42.02 53.09 42.73	84.92 82.95 86.85	56.22 64.74 57.28	61.15 64.4 61.88	85.4 89.18 87.83	66.82 70.89 67.99
B1 + Spectral	R	P	F1	R	P	F1	R	P	F1
Entity Event Both	55.01 63.25 57.37	58.01 29.24 66.26	56.47 39.99 61.5	53.23 76.79 50.7	47.12 27.36 50.46	59.99 40.35 50.58	66.27 66.6 66.79	57.29 52.56 58.47	59.93 53.86 61.19

Looking into the source of the errors, we found that the major source of the error was that fact that the matrix **A** computed in step 2. of the SCM is very sparse. As a result this the matrices **U** and **V** have elements that have values often in several negative powers of magnitude. This makes it impossible for an algorithm like k means that embeds these points in the metric space to be able to find meaningful clusters and it ends up clustering a number of them in the same cluster.

We initially thought we could work around it by using a lenient linking function. However we soon realized that the base relations between mentions are so few and far between that any such strategy will end up creating too many bipartite links. We also tried to smooth the function by allowing at least one match per mention pair. But that did not seem to have much effect on the cluster quality.

Another thing we tried to solve this problem was to change the **U** and **V** matrices to a logarithmic scale. The hope was this will allow for a more reasonable computation of distances to allow for k-means to find some meaningul patterns. We were again met with disappointment as the sparsity cause the log values to turn negative infinity or NotANumber.

We then tried to run this algorithm *over clusters* instead of *over mentions*. That is, in step.2 of the algorithm we created lists of the know entity and event clusters. The partition function in this case classifies a cluster as an event cluster if the first mention of the cluster is a verb. Otherwise the cluster is put in the Entity cluster list. One problem with this approach was that we do not have any prior knowlege of clustering over the clusters. Hence the C matrix essentially becomes zero. However we hoped that the Bipartite edge weight matrix **E** would be a lot less sparse. And as the co-clustering operation depends more crucially on this infomation flow between entities and events, we were hoping to get better results. However the matrices were still largely sparse. It was then we realized how minimal and localized the information flow across entities and events is.

One approach that we tried to then implement but could not due to the lack of time was to perform co-clustering at the level of individual documents instead of the document cluster. The motivation for this is that for the Events and Entities within a document, the information flow is expected to be less sparse. Hence more meaningful patterns can possibly be discerned and the information from

these patterns can then be applied across documents.

We were eventually not able to work around this problem of sparsity. This prevented us from meaningfully carrying out any of the experiments we had planned on the effect of the model selection parameter δ, finding the optimal number of clusters, comparing the various linking functions and trying out a better partition function.

Conclusion and Future Work

Despite the initial promise that the idea held, it seems now to us that performing Joint coreference of Entities and Events using co-clustering is a rather challenging task. Inspite of the disappointments in our efforts, we believe there may be ways to work around the problem of sparsity. We would in particular like to try develop methods to discern the information flow across events and entities more locally, maybe within a document and try to come up with smart ways to apply these to the complete set.

Acknowlegments

We would like to thank Spence Green for the initial guidance he provided. We are also deeply grateful to Heeyoung Lee for the constant help he provided in terms of advice,providing us the dataset and for lettting us reuse his code.

References

1. Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint Entity and Event Coreference across Documents
2. I.S. Dhilllon. 2001. Co-clustering documents and words using bipartite spectral graph partitioning
3. Xiaoxiao Shi, Wei fan, and Philip S. Yu. 2010. Efficient Semi-supervised Co-clustering with Constraints
4. Cosmin Bejan and Sanda Harabagiu. 2010, Unsupervised Event Coreference Resolution with Rich Linguisitic features.