Graph Clustering on Crime Database

The present work describes application of graph clustering approaches on Crime Database. The Crime database is analyzed to identify the entity nodes, their attributes and the relationship node that joins the entity nodes along with the relationship node attributes.

The Crime Dataset used in the present work has been obtained from the web source www.iamnirbhaya.me. The Web Platform crowd source media reported crimes of violence against Women and Children in India with the following objectives

- Survivors can visually understand the extent and realise that they are not alone. it will motivate them to report what happened to them and seek justice.
- Common public, Government, NGOs, Funding Agencies, Media can visualise the extent of the reports and also download reports of all cases for research, campaign, advocacy and problem solving.

Today the platform hosts information about **10,000 Documented media reports** of violence against Women and children in India that have been crowd sourced by volunteers. Most important features of the Crime Dataset can be defined as

- News Collection from online News sites
- Crawled and then text formatted
- Crime Categories – Dowry Harassment, Murder, Child Abuse etc.
- State wise Collection of crime news for different crime categories on Women and Child

The task set for the present work has been identified as follows:

- Feature extraction from Crime News
- Organize each crime event as a nodes in a graph
- Application of Graph Clustering algorithm for community detection

Based on the preliminary study on the sentences of the crime news crawled from the abovementioned website the following design considerations were identified.

- Sentences with verbs related to crimes will be considered
- Entities refer to the victim and the perpetrator, one who commits the crime
- Relations between the two entities refer to the crime
- More attributes or features are available for relations and less so for the entities
- Both Entities and Relations are to be modelled as Graph Nodes in the Graph model

Regarding the implementation level, it is identified that Entities, Relations and Attributes of entities and relations can be modelled as the output of the Dependency parser. The Stanford Dependency parser is used for this task. The main verb or the root of the parse tree forms the relation. If the sentence is in active voice, the subject of the main verb identifies the perpetrator while the object of the main verb defines the victim. If the sentence is in passive voice, the object of the main verb identifies the perpetrator while the subject of the main verb defines the victim. Various dependency relations like nmod, advcl (with the verb) identify the crime features associated with the relation node. Similarly, dependency relations like amod, nmod with the subject and object identify the features associated with the perpetrator or the victim.

The following work plan has been taken up:

- Identification of information nodes for each crime event, considering that each sentence in the corpus may refer to a crime event.
- Node 1: crime and the relevant extracted features (Relation Node – R Node)
- Node 2: who has committed the crime and relevant extracted features (Entity Node - Perpetrator)
- Node 3: on whom the crime has been committed and relevant extracted features (Entity Node - Victim)
- The following two Edges have been considered for inclusion in the Graph database: Node 2 -> Node 1 and Node 3 -> Node 1
- All the R nodes are connected forming the crime graph

The tasks to be carried out have been organized as follows:

- Identification of some crime related verbs – attacked, murdered, killed etc.
- Application of Stanford Dependency Parser on each sentence that include the crime related verb
- Feature extraction of Node Metadata
- Graph Modelling in – Edge List Format
- Graph Clustering in SNAP tool using the Clauset-Newman-Moore Hierarchical Agglomeration Algorithm

Now, we consider some example sentences and the node metadata generated from the sentence.

Example Sentence 1: “An enraged Danniah attacked Ashok with an axe, inflicting grievous injuries on him.

Dependency Prase output:

- ((u'attacked', u'VBD'), u'nsubj', (u'Danniah', u'NNP'))
- ((u'Danniah', u'NNP'), u'det', (u'An', u'DT'))
- ((u'Danniah', u'NNP'), u'compound', (u'enraged', u'NNP'))
- ((u'attacked', u'VBD'), u'dobj', (u'Ashok', u'NNP'))
- ((u'Ashok', u'NNP'), u'nmod', (u'axe', u'NN'))
- ((u'axe', u'NN'), u'case', (u'with', u'IN'))
- ((u'axe', u'NN'), u'det', (u'an', u'DT'))
- ((u'attacked', u'VBD'), u'advcl', (u'inflicting', u'VBG'))
- ((u'inflicting', u'VBG'), u'dobj', (u'injuries', u'NNS'))
- ((u'injuries', u'NNS'), u'amod', (u'grievous', u'JJ'))
- ((u'inflicting', u'VBG'), u'nmod', (u'him', u'PRP'))
- ((u'him', u'PRP'), u'case', (u'on', u'IN'))

The node metadata generated from the above sentence is in the following format <FeatureName_FeatureValue> pairs.

- 1.R.label_crime.name_attacked.result_inflicting grievous injuries on him
- 2.N.label_who.name_An enraged Danniah
- 3.N.label_whom.name_Ashok with an axe

The first entry in the nsubj relation identifies the label feature of the R node which is crime as well as the name feature of the R node. Thus, in this case the following dependency relation

```
((u'attacked', u'VBD'), u'nsubj', (u'Danniah', u'NNP'))
```

generates the node metadata

1.R.label_crime.name_attacked.

The second entry in the nsubj relation identifies the name feature (Danniah) of the who node. Next, identify the relations in which ‘Danniah’ appear as the first entry and collect the words that appear as the second entry in those relations. Arrrange these words in the order in which they appear in the sentence to form the composite name feature of the who node. Thus, in the above example, the dependency relations

```
((u'Danniah', u'NNP'), u'det', (u'An', u'DT'))
((u'Danniah', u'NNP'), u'compound', (u'enraged', u'NNP'))
```

generate the who node metadata as

2.N.label_who.name_An enraged Danniah.

The second entry in the dobj relation is the name feature (Ashok) for the whom node. Next identify the relations in which Ashok appears. If the second entry in those relations appears in ‘case’ and ‘det’ relations, identify all these words, put them in the order in which they appear in the sentence and assign this word sequence as the composite name feature for the whom node. Thus, the following set of functional dependencies

```
((u'attacked', u'VBD'), u'dobj', (u'Ashok', u'NNP'))
((u'Ashok', u'NNP'), u'nmod', (u'axe', u'NN'))
((u'axe', u'NN'), u'case', (u'with', u'IN'))
((u'axe', u'NN'), u'det', (u'an', u'DT'))
```

generate the whom node metadata

2.N.label_whom.name_Ashok with an axe.

It is observed that due to error in parsing, ((u'attacked', u'VBD'), u'nmod', (u'axe', u'NN')) is the correct parse instead of ((u'Ashok', u'NNP'), u'nmod', (u'axe', u'NN')), the instrument feature (an axe) of the crime is missing.

The second entry (inflicting) in the advcl relation with the crime verb as the first entry identifies the result feature of the crime node. All the relations in which the word ‘inflicting’ occurs as the first entry are considered and words appearing as the second entry in all such relations are collected. If any such word appears in nmod relation, then the words with which this word forms ‘case’ or ‘det’ relations are considered and put in the same set. The words in the set are ordered in which they appear in the sentence forming the result feature of the crime node. Thus, the set of dependency relations

```
((u'attacked', u'VBD'), u'advcl', (u'inflicting', u'VBG'))
((u'inflicting', u'VBG'), u'dobj', (u'injuries', u'NNS'))
((u'injuries', u'NNS'), u'amod', (u'grievous', u'JJ'))
((u'inflicting', u'VBG'), u'nmod', (u'him', u'PRP'))
((u'him', u'PRP'), u'case', (u'on', u'IN'))
```

generate the crime node metadata as

1.R.label_crime.name_attacked.result_inflicting grievous injuries on him

The processing is over as all the relations have been scanned and processed.

Let us consider that the following 9 nodes have been identified after analyzing the following three sentences.

The three sentences are:

Sentence 1: An enraged Danniah attacked Ashok with an axe, inflicting grievous injuries on him.

Sentence 2: A farmer allegedly attacked his wife and her ‘friend’ with an axe at Hayathnagar on Sunday morning.

Sentence 3: A 30-year-old woman was attacked with a cleaning acid allegedly by her uncle in an incident at Katrapadu village in Pedanandipadu mandal.

- 1.R.label_crime.name_attacked.result_inflicting grievous injuries on him
- 2.N.label_who.name_An enraged Danniah
- 3.N.label_whom.name_Ashok with an axe
- 4.R.label_crime.name_attacked.instrument_an axe at Hayathnagar.time_Sunday morning
  5.N.label_who.name_A farmer
- 6.N.label_whom.name_his wife and her friend.
- 7.R.label_crime.name_attacked.instrument_a cleaning acid.location_ Pedanandipadu mandal
- 8.N.label_whom.name_A 30-year-old woman
- 9.N.label_who.name_allegedly by her uncle in an incident at Katrapadu village

The Graph is formed with the Edge formation strategy that the two N nodes from a sentence form edges with the R node from the same sentence and the R nodes are connected across sentences. Thus the following edge list is formed from the 9 nodes:

- Edge List
  - 1 2
  - 1 3
  - 4 5
  - 4 6
  - 7 8
  - 7 9
  - 1 4

    - 1 7
    - 4 7

The following output has been obtained after Graph Clustering has been done in SNAP tool using the Clauset_newman-Moore algorithm.

- # Input: edgelist.txt
- # Nodes: 9 Edges: 9
- # Algorithm: Clauset-Newman-Moore
- # Modularity: 0.333333
- # Communities: 3
- # NId   CommunityId
- 1   0
- 2   0
- 3   0
- 4   1
- 5   1
- 6   1
- 7   2
- 8   2
- 9   2

Next, we add the edge (10 11) indicating that the nodes 10 and 11 form an edge inserted into the graph. It has been observed that in some sentences the perpetrator information is not present but the victim information is there. When the clustering algorithm is run again on the new edge list, the newly added nodes are clustered in a separate cluster and the cluster modularity increases to 0.420000.