ENHANCING CLUSTER LABELLING USING WIKIPEDIA

ABDUR RAAFIU MOHAMED FAROOK
Contents

- Introduction
- Approach
- General Framework
  - Indexing
  - Clustering
  - Important Terms Extraction
  - Label Extraction
  - Candidate Label Evaluation
- Experiment
- Conclusion
Introduction

What is the need of Document Clustering?

- Organize data in manageable forms

How the Clusters should be?

- Documents with in cluster are as similar as possible
- Documents from different clusters should be dissimilar

And then Cluster Labeling

How it is done?

- Applying statistical techniques for feature selection
- “important” terms that best represent the cluster topic
Why there is a need of another system?

- Keywords or phrases fails to provide a meaningful label
- It represent different aspects of the topic underlying the cluster
- A good label may not occur directly in the text
Cluster labeling using JSD

<table>
<thead>
<tr>
<th>ODP category</th>
<th>Top-5 JSD important terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bowling</td>
<td>bowl, bowler, lane, bowl center, league</td>
</tr>
<tr>
<td>Buddhism</td>
<td>Buddhist, Buddhism, Buddha, Zen, dharma</td>
</tr>
<tr>
<td>Ice Hockey</td>
<td>hockey, nhl, hockey league, coach, head coach</td>
</tr>
<tr>
<td>Electronics</td>
<td>voltage, high voltage, circuit, laser, power supply</td>
</tr>
<tr>
<td>Tennis Players</td>
<td>Wimbledon, tennis, defeat, match today, Wta</td>
</tr>
<tr>
<td>Christianity</td>
<td>church, catholic, ministry, Christ, grace</td>
</tr>
</tbody>
</table>

ODP- Open Directory Project
JSD- Jensen-Shannon Divergence
Approach

1. Extracts the most important terms from the documents
2. Find relevant Wikipedia pages
3. The categories and titles (meta-data) are candidates and in addition all important terms from documents also candidates
4. Evaluation by several judges
5. Top ranked candidate as cluster labels
<table>
<thead>
<tr>
<th>ODP category</th>
<th>Top-5 JSD important terms</th>
<th>Top-5 Labels Using Wikipedia Enhancement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bowling</td>
<td>bowl, bowler, lane, bowl center, league</td>
<td>Bowls, Bowling, Bowling (cricket), Bowling organizations, Bowling competitions</td>
</tr>
<tr>
<td>Buddhism</td>
<td>Buddhist, Buddhism, Buddha, Zen, dharma</td>
<td>Buddhism, History of Buddhism, Buddhism by country, Tibetan Buddhism, Buddhists</td>
</tr>
<tr>
<td>Ice Hockey</td>
<td>hockey, nhl, hockey league, coach, head coach</td>
<td>Ice hockey, Ice hockey leagues, Hockey prospects, Canadian ice hockey coaches, National Hockey League</td>
</tr>
<tr>
<td>Electronics</td>
<td>voltage, high voltage, circuit, laser, power supply</td>
<td>Electronics, Power electronics, Diodes, Power supplies, Electronics terms</td>
</tr>
<tr>
<td>Tennis Players</td>
<td>Wimbledon, tennis, defeat, match today, Wta</td>
<td>Tennis Players, Tennis terminology, Tennis tournaments, 2002 in tennis, 2000 in tennis</td>
</tr>
<tr>
<td>Christianity</td>
<td>church, catholic, ministry, Christ, grace</td>
<td>Christianity, Christian denominations, Non-denominational Christianity, Christian theology, Christianity in Singapore</td>
</tr>
</tbody>
</table>
General Framework

1. Indexing

2. Clustering

3. Important Terms Extraction

4. Label Extraction

5. Candidate Evaluation

Documents

Clusters

Inverted index

Important Terms

Labels Extraction

Top k labels

Web

Wikipedia

The Free Encyclopedia
1. Indexing

- Parsed, tokenized and represented as term vectors
- Term weights are determined by tf-idf
- Indexed by generating a search index such that \( tf \) and \( idf \) value of each term \( t \) can be quickly accessed
2. Clustering

- It creates coherent clusters
- Given the input as collection of documents $D$, it returns a set of document clusters $C = \{C_1, C_2, \ldots, C_n\}$
- A cluster is represented by the centroid of the cluster’s documents
- The weights of the terms in centroid is slightly modified

$$w(t, C) = \text{ctf}(t, C) \cdot \text{cdf}(t, C) \cdot \text{idf}(t)$$

where

$$\text{ctf}(t, C) = \frac{1}{|C|} \sum_{d \in C} f(t, d)$$

$\text{cdf}(t, C) = \log(n(t, C) + 1)$

Where $n(t, C)$ is the document frequency of $t$ in $C$
3. Important terms extraction

- Given a cluster $C \in \mathcal{C}$; input

- And to find a list of terms $T(C) = (t_1, t_2, \ldots, t_k)$

- Term $T(C)$ is that which best separates the cluster’s documents from the entire collection

- Jensen-Shannon Divergence (JSD) is used to measure the distance between the cluster $C$ and the entire collection for a set of terms

- Each term is scored according to their JSD distance

- The top scored terms are selected as Cluster important terms
4. Label extraction

- Given the important terms T(C)
- And to extract candidate labels for cluster C

Two types
  i. Use directly top-k important terms
  ii. Use this top-k important terms to execute a query $q$ against Wikipedia index

The result is a list of documents $D(q)$

Documents title and categories are considered as potential candidate cluster labels $L(C)$
5. Candidate label evaluation

- Done by several judges
- Given the input for judges are L(C) and T(C)
- Two judges
  1. MI judge
  2. SP judge
- The scores of all judges are then aggregated and the label with highest score returned
MI (Mutual Information) judge

- It scores each candidate by the average pointwise mutual information (PMI) with respect to a given external textual corpus.
- The average PMI reflects the semantic distance of the label from the cluster content.
- Labels closer to the cluster content are preferred.
MI (Mutual Information) judge

- Given the input is $L(C)$, $T(C)$ and a corpus (Google n-grams)
- Given a candidate label $l \in L(C)$, the following score is assigned to $l$:

$$\text{MI}(l, T(C)) = \sum_{t \in T(C)} \text{PMI}(l, t|\text{corpus}) \times \omega(t)$$

Where $\omega(t)$ denotes the relative importance of important term $t$ in $T(C)$.

- The PMI between two terms is measured by:

$$\text{PMI}(l, t|\text{corpus}) = \log \left( \frac{Pr(l, t|\text{corpus})}{Pr(l|\text{corpus}) \times Pr(t|\text{corpus})} \right)$$

- The probability of a term is approximated by the maximum likelihood estimation

$$Pr(x|\text{corpus}) = \frac{\#(x|\text{corpus})}{\#(\text{corpus})}$$
SP(Score Propagation) judge

- It scores each candidate label with respect to the scores of the documents in the result set associated with that label.

  Given \( l \in \mathcal{L}(C) \), the score propagation from \( D(q) \) to \( l \), weight for \( l \) is represented as,

  \[
  \omega(l) = \sum_{d \in D(q): l \in d} \frac{score(d)}{n(d)}
  \]

  \( n(d) \) - number of candidate labels extracted from document \( d \)

- Scoring of label keywords

  \[
  \omega(kw) = \sum_{l \in \mathcal{L}(C): kw \in l} \omega(l)
  \]

  Each candidate label is scored by the average score from its keywords

  \[
  SP(l|D(q)) = \frac{1}{n(l)} \sum_{kw \in l} \omega(kw)
  \]

  \( n(l) \) - number of \( l \)’s unique keywords
The final stage is to aggregate the scores from the different judges for each label.

Each candidate label is scored using a linear combination of the judge ($J_1, ... J_m$) scores:

$$\text{score}(l|C) = \sum_{i=1}^{m} \beta_i J_i(l|C)$$

Where $\sum_i \beta_i = 1$

Finally the set of top-k scored candidates are recommended for cluster labeling.
Data Collection

I. 20 News Groups (20NG) data collection
   - Newsgroup documents that were manually classified into 20 different categories
   - Each category includes 1,000 documents, so totally 20,000 documents

II. Open Directory Project (ODP)
   - Randomly selected 100 different categories from the ODP hierarchy
   - Example categories include, among others, sub-categories of the top level ODP categories such as Ceramic Art and Pottery
   - From each category randomly selected up to 100 documents, so totally 10,000 documents
Evaluation and Experimental setup

- Given a collection of clusters, and the parameter k
- The system proposes up to k labels for each cluster

**Evaluation of system’s performance:**

- Two methods were used

  I. **Match@K**
     The relative number of clusters for which at least one of the top-k labels is correct.

  II. **Mean Reciprocal Rank (MRR@K)**
     Given an ordered list of k proposed labels for a cluster, the reciprocal rank is the inverse of the rank of the first correct label, or zero if no label in the list is correct. The MRR@K is the average of the reciprocal ranks of all clusters.
Here four different feature selection methods also compared
There are two significant parameters that can affect the quality of Wikipedia’s labels:

I. The number of important terms that are used to query Wikipedia
II. The number of top scored results from which candidate labels are extracted
Observations for all judges shows, as $k$ increases (i.e., more cluster labels are proposed) the MRR score increases.

Overall, among the four different judges, the SP(rank) judge performs the best.
The Effect of Clusters’ Coherency on Label Quality

- A cluster of documents given to the labelling component is usually the corresponding result of the clustering algorithm used by the system.

\[ \text{coherency}(C) = \frac{\sum_{i=1}^{n} \frac{|C_i|}{|D|} \text{sim}_{in}(C_i)}{\text{sim}_{out}(C)} \]

Testing on a noisy cluster

- For a noise level \( p(\text{in}[0,1]) \) of clusters, each document in one cluster have probability \( p \) to swap with document in other cluster.
Conclusion

Advantages

- Cluster labeling can be enhanced by utilizing the Wikipedia knowledge-base
- A detailed evaluation is done all the phase of the Framework
- Evaluation results demonstrates the proposed system is robust and resiliency to noise

Disadvantages

- The topics which are not covered by Wikipedia may affect the system performance
Thank you!