Web Page Clustering of Ambiguous Names

Submitted by : Satheeshkumar Karuppusamy
USC ID : 7349-7199-00

1. Introduction

Considering the enormous size of the World Wide Web, it is more common that there will be more than one real world entity mapping to each and every name. There will obviously be more than one resource in the web pointing to each and every ambiguated entities. Disambiguating those names and grouping the web resources according to the disambiguated entity name is an interesting task which makes use of Clustering and Topic modeling approaches in Natural Language Processing perspective. This particular report contains the details about the machine algorithms that have been applied to solve this web document clustering problem and the result of those experiments, that has been done with the SemEval-2007 dataset. Section2 describes the data Preparation tasks that have been done for applying NLP algorithms. Section3 describes the Experimentation with the set of machine learning algorithms that are being applied for this problem and their results on the given dataset. Section4 describes the best performing systems on different settings. Section5 concludes this report.

2. Data Preparation

The input document that have been provided for this task is the raw HTML dataset from SemEval-2007 task. Before applying NLP algorithms for these documents, it is important to remove formatting tags and make sure we have only the necessary text content. The tools that are being used to clean the HTML data and form the purified text files are as given below :

* Beautiful Soup
* Mozilla Bleach
* NLTK Sentence Segmenter
* Sentence Boundary Detector(Splitta 1.03)
* Stanford Tokenizer

2.1 Using Beautiful Soup and Mozilla Bleach for basic Text Content Extraction

Beautiful Soup is a python library which provides extensive functionalities for extracting the text content from HTML document by following standard Document Object Model structure. All the text content in the HTML document are extracted using predefined methods in Beautiful soup package. One issue with the extracted text content is that they still contain some java scripts and CSS style definitions which cannot be removed properly using Beautiful soup. Particularly for removing them, the partially purified text content is processing using Beautiful soup which gets almost purified text content.

2.1.1 Extracting Rich Features while purifying basic text information

The email addresses in a web document defines some important feature set. The name and domain part of the email address denotes some important uniqueness which can be exploited for clustering purpose. While preprocessing the data, if the token is a web email address, it is parsed as follows and the parsed tokens are added to the parsed content.

Example email address : **abby@climbing.com**

Parsed Output tokens :

* **abby**
* **climbing**

The above input email address is parsed into the tokens as given below and then added to the actual content of the web page which will act as unigram features in the feature space. Similarly content in the <title> tag are extracted and added to the feature space. Adding these kinds of features will add our feature set richer than what we had.

2.1.2 Sentence Segmentation Using NLTK Sentence Segmenter and Splitta 1.03

While calculating context dependent n-gram features, it is more important to set the context boundaries. Segmenting the entire corpus into individual sentences will make sure that those contexts are extracted properly. For this purpose the NLTK Sentence Segmentation and Splitta sentence segmentation tools are tried. NLTK Sentence segmentation performed badly when we had non ASCII characters in our corpus. But removing the non ascii characters and performing sentence segmentation using NLTK resulted in getting better segmentation than using Splitta.

2.1.3 Tokenization using Stanford Tokenizer

The sentence segmented input corpus is then fed as input to the Stanford word tokenizer which tokenizes each words in the sentences of the corpus properly which can then be used for finding n gram features. The output from the tokenizer is a well-cleaned set of word tokenized sentence segmented text files with abundant feature sets including title and parsed email information from every web page.

3. Experimentation with Different Machine Learning Algorithms

The different clustering and topic modeling algorithms that are being tried for this particular assignment are as given below :

Clustering Algorithms
* **K-means Clustering Algorithm**
* **EM Algorithm**
* **Hierarchical Clustering**

Topic Modeling Algorithm
* **Latent Dirichlet Allocation**

The following toolkits are used for performing the experiments :
* Weka Toolkit
* Mallet Toolkit

3.1 Clustering Algorithms

3.1.1 Feature Sets used

The basic feature set used for this task are as described below :

* **Unigrams**
* **Bigrams**
* **TF_IDF**

The given text files under a particular entity are converted into a single arff file with the content of the text document is represented as every data point in the arff file. The arff document is fed into the weka toolkit and the StringToVector filter is applied. Instead of just storing the counts of the words in the documents, TF-IDF is computed for every unigram and bigram terms in the document. Required StringToVector filter properties like lowercase only words, tf-idf and n gram features are applied to the loaded arff files and the following clustering algorithms are applied. All the clustering algorithms are tested under two following scenarios :

* **Treating the discarded files as an additional Cluster (Scenario1)**

In this scenario, discarded files are not pre-calculated. All the input files about particular entity is fed as input to the clustering algorithm and the documents are clustered based on the parameters of the algorithm.

* **Pre-processing Discarded Files (Scenario2)**

In this scenario, file is marked as discarded if it does not contain the any of the token that are present in the entity names. For example, a document under the entity "Abby Watkins" does not contain both the words "Abby" or "Watkins", then the document is added to the discarded list

Surprisingly, both of the above scenarios fetches almost the same accuracy of clustering. This means the documents which doesnt contain the entity names are grouped themselves under the same cluster by our clustering algorithms.

3.1.2 K-means Clustering Algorithm

Applying K-means algorithm for the above two scenarios fetches almost the same clustering accuracies. The default settings for the K-means algorithm are initially tried with the setting given below :

Seed value : 10
Number of Iterations : 500
Number of Clusters : #Varied for each entity

Same initial seed value is chosen for every entities and the accuracy is calculated. Since the clustering accuracy of k-means algorithm mainly depends on the initial centroid value of each clusters, the initial centroid value is modified to different value and the accuracy of clustering is calculated. The different seed values chosen are

Seed values : 50 and 10

It has been observed that setting seed value to 50 performs poor clustering than setting the seed value to 25. The clustering performed by k-means algorithms are fed into the evaluator and the following output has been observed for different k seed values

Topic	F-Score when Seed value = 10	F-Score when Seed Value = 50
Abby Watkins	0.67	0.47
Cathie Ely	1.00	1.00
Dan Rhone	1.00	0.67
Jane Hunter	0.35	0.40
Michael Howard	0.46	0.46
Thomas Baker	0.67	0.67
Tim Whisler	0.56	0.57
Average	**0.67**	**0.61**

3.1.3 Hierarchical Clustering

Single Linkage Hierarchical Clustering has also been applied to the dataset and the performance is analyzed. The performance of Single Linkage Clustering almost reaches the baseline performance and it is faster than the expected runtime of O(n^3) since our dataset is comparatively smaller than the usual size of dataset which are normally used for clustering purposes. Given below are the F-scores when applying Hierarchical clustering algorithm with respect to two scenarios.

Topic	F-Score(Single Linkage Clustering)
Abby Watkins	0.73
Cathie Ely	1.00
Dan Rhone	0.67
Jane Hunter	0.39
Michael Howard	0.49
Thomas Baker	0.69
Tim Whisler	0.55
Average	**0.65**

3.1.4 EM Algorithm

EM algorithm is also tried for the dataset under different settings. Unlike other clustering algorithms, the performance of the EM algorithm is measured in terms of improvement of the log likelihood of the

data being observed. The optimization curve of the EM algorithm often ends up in local maximum likelihood points instead of reaching the global maximum likelihood point. This can be avoided by restarting the EM algorithm with different initial settings. This approach has been tried in my experiment. The different seed values like 50 and 100 are tried and the results are furnished as given below :

Topic	F-Score when Seed value = 50	F-Score when Seed Value = 100
Abby Watkins	0.76	0.73
Cathie Ely	1.00	1.00
Dan Rhone	0.67	0.67
Jane Hunter	0.51	0.44
Michael Howard	0.44	0.50
Thomas Baker	0.67	0.70
Tim Whisler	0.56	0.55
Average	**0.66**	**0.65**

3.2. Latent Dirichlet Allocation(Clustering by Topic Modeling) using Mallet

A generative topic modeling algorithm called Latent Dirichlet Allocation has been applied to uncover the underlying latent topic document relation and cluster the documents into different clusters based on their topic distribution. Based on the probability of each topic associated with the current document, the document is given a topic id to which it is strongly associated. That topic id is chosen as the cluster id for that document. The number of topics chosen is set to the value of number of clusters that we intend to divide the document collection. This algorithm is implemented in a toolkit called which has been exploited for our experimentation purpose. This algorithm works failry better in classifying documents based on the topic distribution. Observe that the documents under 'Jane Hunter' entity has been more accurately classified only in LDA when compared to clustering methods which shows documents under that entity have different topics. The documents which have more probability of similar topics are later clustered. The results for the current dataset when we apply LDA is as given below :

Topic	F-Score
Abby Watkins	0.42
Cathie Ely	1.00
Dan Rhone	0.67
Jane Hunter	0.70
Michael Howard	0.51
Thomas Baker	0.75
Tim Whisler	0.54
Average	**0.66**

4.Performance Analysis

4.1 Hierarchical Clustering

Documents 42 and 112 under "Abby Watkins" deals with the some accidental tragedy that occurred while climbing the mountains which has been grouped correctly. But propagating the general idea of "Abby Watkins" being a climber is not propagated while merging with the other clusters that are formed. This makes this system to perform poorer than k-means when clustering several documents to form a cluster.

4.2 K-means

K-means algorithm, as it has been already stated, depends entirely on choosing the initial centroid values of the cluster. K-means usually calculates the euclidean distance between the centroid of each cluster along with the data points and cluster them such that the total distance from the centroid is reduced in each step. Choosing a better starting points achieves better accuracy of clustering the documents and it performs better than Hierarchical clustering algorithm.

4.3 EM Algorithm

EM algorithm often ends up in reaching local maximum and thus it performs worser than k-means with imperfect initial parameter values. Restarting k-means algorithm with random parameter values is often a better way of choosing best parameters for clustering the documents. But it usually takes much time than the above algorithms. With respect to running time, EM optimizes all the parameter values at each iteration such that the likelihood is increased which makes it to consume more runtime than the above algorithms.

4.4 Latent Dirichlet Allocation

LDA technique is the best algorithm in terms of time and efficiency trade-off. This algorithm unveils the topic distribution within each document and finds the probability of association with each document and a topic. The topic with which a document is most associated is chosen as the cluster of that document. This algorithm performs fairly better with lesser runtime.

5. Description of Best Performing System

K-means with the seed value = 10 is the best performing systems among the systems that has been experimented which has the F-Score of 0.67. EM with the seed value 50 and LDA performs equally well with the F-score of 0.67.

Conclusion

Clustering algorithms like k-means, Hierarchical clustering and EM algorithm and topic modeling algorithms like Latent Dirichlet Allocation has been successfully and experimented with the web document dataset and the performance of those algorithms under different settings are studied and experimented successfully. It is clearly observed there is a trade off between the runtime of each algorithm and the performance of the better performance system. It is also observed that enrichment of

feature space and a better preprocessing is very much necessary to get better results.

Reference

[1] Ying Chen, James Martin, CU-COMSEM: Exploring Rich Features for Unsupervised Web Personal Name Disambiguation
[2] Zornitsa Kozareva, Sonia Vazquez and Andres Montoyo, UA-ZSA: Web Page Clustering on the basis of Name Disambiguation
[3] http://mallet.cs.umass.edu/api/cc/mallet/topics/LDA.html
[4] http://weka.wikispaces.com/Using+cluster+algorithms
[5] David M. Blei, Andrew NG, Michael I. Jordan, Latent Dirichlet Allocation