Community Detection for testing hypothesis of dispersion similarity

Vamsi krishna D and Vidhey Kumar PV

Abstract— **Here, we proposed a generic framework that aims at effectively identifying and characterizing the main specializations of the subfields of a scientific domain by leveraging paper contents. A scientific domain paper consists of subfields that can be further refined into specializations which emerge, evolve and consolidate, as reflected in particular in literature development, along a contents-based dimension where important problems are stated and addressed,and along a communal dimension where researchers collaborate and compete to solve those problems. More specifically, the latent knowledge structure of a domain is discovered and progressively refined along both the contents-based and communal dimensions.**

I. INTRODUCTION

Take a specific domain like Machine Learning in computer science field.It consists of a number of subfields : Supervised Learning, Unsupervised learning, Reinforcement Learning, Machine Translation etc. These subfields are themselves subdivided into specializations.For instance, Supervised Learning encompasses the specializations of Clustering, Classification, Regressions, Retrieval, BigData; Machine Translation encompasses the specializations of English to Chinese Machine Translation, English to French Machine Translation.

Specialization consists of a number of closely related studies that jointly endeavour to solve particular problems or develop certain techniques, based on the efforts of a community of researchers who closely work together, whether they collaborate or compete, whether they use each others work or oppose their frameworks or experiments.Therefore, a specialization comes to existence and develops along two dimensions: a contents-based dimension along which we find knowledge, problems, methods, etc.; a communal dimension along which we find citations, work-shops, etc.

In this project we mainly concentrated to divide research papers into some sets such that each set consisted of closely related research papers.In Section II we described about what dataset we took. In section III we described about pre-processing details, Section-IV contains feature extraction details, Section-V describes what algorithms used and brief explanation of that algorithm.In section VI, we describe the results we got by running that algorithm on given dataset by changing some parameters in algorithm.

II. DATASETS

The dataset considered here is set of 345 research papers from various domains in computer science field. In this project we mainly concentrated to divide research papers into some sets such that each set consisted of closely related research papers. Research Papers include Papers belonging to various fields such as: Natural Language Processing, Machine Learning, Computer Vision... etc.

III. PRE-PROCESSING DETAILS

Pre-Processing is essential for efficient Feature Extraction leading to non-redundant Feature Descriptors. The main steps involved in the Pre-Processing Stage of the Pipeline are :

A. Word Tokenization

Given a sentence a list of words are generated by considering space as delimiter. As many of the chemical entities consists of special characters, punctuation normal word tokenizers cannot give good results.

B. Stopwords Removal

Stop words refer to the most common words in a language. Although, there is no universal list for the Stop Words, but an exhaustive list has been used to ensure better accuracy and efficient feature extraction. They need to be removed because they are most likely to be not chemical names.

C. Tf-Idf Method

We used pdf2txt to convert given set of research papers to text files,then we removed all the stop words from data and then did stemming on the data.We then used word2vec library to convert given word to a vector.Then we assigned weights to each word using tf-idf method.For a given document we applied pos tagging and then took only proper-nouns (NNP) and key words.For these words we used tf-idf as weight and multiplied it with corresponding word-vector.This is taken as document vector.

For each document,we calculate document vector as follows:

$$Doc-vector = \sum_{for\ each\ NNP} tf-idf * word-vec$$

IV. FEATURE EXTRACTION

After the pre-processing stage, sufficient amount of redundancy gets removed from the patent content. The words now need to be converted to equivalent feature descriptors to ensure good classification and low enough to ensure that computation performed on them is tractable.

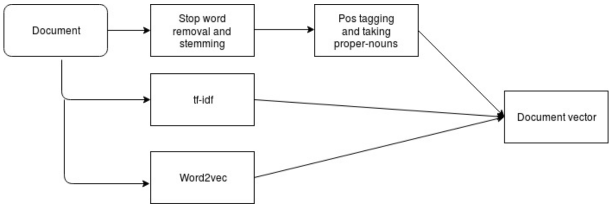
A. *Word and Pos tags*

The token itself (in lowercase) was added as a feature. POS tags were generated using pos-tag from nltk were also added as a feature.

V. ALGORITHMS

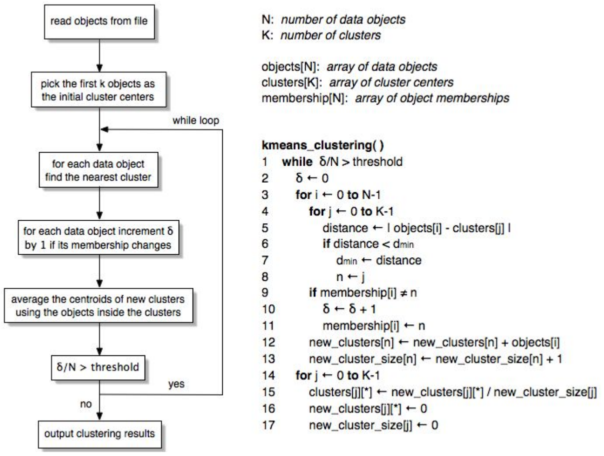
j We used k-means algorithm to cluster the data.We changed number of clusters and ran this algorithm on that dataset and observed the results.

K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other.


Document
Stop word removal and stemming
Pos tagging and taking proper-nouns
tf-idf
Word2vec
Document vector

The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycenter of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center. A loop has been generated. As a result of this loop we may notice that the k centers change their location step by step until no more changes are done or in other words centers do not move any more. Finally, this algorithm aims at minimizing an objective function know as squared error function which is sum of squares of euclidian distances of a point from its cluster center.

$$\text{objective function} \leftarrow J = \sum_{j=1}^{\overset{\text{number of clusters}}{k}} \sum_{i=1}^{\overset{\text{number of cases}}{n}} \underbrace{\left\| \overset{\text{case } i}{x_i^{(j)}} - \overset{\text{centroid for cluster } j}{c_j} \right\|^2}_{\text{Distance function}}$$


read objects from file
pick the first k objects as the initial cluster centers
while loop
for each data object find the nearest cluster
for each data object increment δ by 1 if its membership changes
average the centroids of new clusters using the objects inside the clusters
δ/N > threshold
yes
no
output clustering results

```
N:  number of data objects
K:  number of clusters

objects[N]:  array of data objects
clusters[K]:  array of cluster centers
membership[N]:  array of object memberships

kmeans_clustering( )
while  δ/N > threshold
    δ ← 0
    for i ← 0 to N-1
        for j ← 0 to K-1
            distance ← | objects[i] - clusters[j] |
            if distance < d_min
                d_min ← distance
                n ← j
        if membership[i] ≠ n
            δ ← δ + 1
            membership[i] ← n
        new_clusters[n] ← new_clusters[n] + objects[i]
        new_cluster_size[n] ← new_cluster_size[n] + 1
    for j ← 0 to K-1
        clusters[j][*] ← new_clusters[j][*] / new_cluster_size[j]
        new_clusters[j][*] ← 0
        new_cluster_size[j] ← 0
```

VI. RESULTS

TABLE I
OBSERVATIONS

No of Clusters	no of documents in each cluster
5	36, 76, 75, 64, 96
7	34, 49, 69, 75, 44, 37, 41
9	9, 32, 8, 47, 14, 44, 49, 68, 76
11	21, 18, 31, 34, 29, 31, 11, 47, 32, 46, 47
13	10, 18, 12, 21, 41, 12, 24, 31, 52, 33, 13, 35, 45
15	11, 7, 18, 21, 10, 21, 7, 24, 37, 30, 33, 26, 33, 53, 16

VII. CONCLUSION

We can group each research-paper belonging to a certain field into a definite cluster based on the specialization in subfield or in any fields