Use of Semantic Relation Between Words in Text Clustering

Bhoopesh Choudhary
CSE Department
Indian Institute of Technology, Bombay
India.
bhoopesh@cse.iitb.ac.in

Pushpak Bhattacharyya*
CSE Department
Indian Institute of Technology, Bombay
India
pb@cse.iitb.ac.in

Abstract

In traditional document clustering methods, a document is considered a bag of words. The fact that the words may be semantically related- a crucial information for clustering- is not taken into account. The feature vector representing the document is constructed from the frequency count of document terms. To improve results, weights calculated from techniques like *Inverse Document Frequency (IDF)* and *Information Gain (IG)* are applied to the frequency count. These weights also are essentially statistical parameters and do not make use of any semantic information.

In this paper we describe a new method for generating feature vectors, using the semantic relations between the words in a sentence. The semantic relations are captured by the Universal Networking Language (UNL) which is a recently proposed semantic representation for sentences. UNL expresses a document in the form of a semantic graph, with nodes as disambiguated words and *semantic relations* between them as arcs. The method described in this paper takes the UNL graph and generates there from feature vectors representing the document. The clustering method applied to the feature vectors is the *Kohonen Self Organizing Maps (SOM)*. This is a neural network based technique which takes the vectors as inputs and forms a document map in which similar documents are mapped to the same or a nearby neurons. Experiments show that if we use the UNL method for feature vector generation, clustering tends to perform better than when the term frequency based method is used.

Keywords: Text clustering, Document vectors, Semantic net/graph, Universal Networking language, Self Organization Maps.

Approximate word count: 6,800.

1 Introduction

The World Wide Web is a vast resource of information and services that continues to grow rapidly. Powerful search engines have been developed to aid locating documents by category, contents, or subject. Relying on large indices to documents, search engines determine the URLs of documents satisfying a user's query. Often queries return inconsistent search results, with document referrals that meet the search criteria but are of no interest to the user. These problems arise largely due to the fact that no account is taken of the meaning of either the query or the documents.

While it may not be currently feasible to make use of the full meaning of a document, we can still extract semantic information from the properties of words, relations between words and/or the structure of a document. This information is then employed to classify and categorize the documents.

* Contacting author

Clustering has the advantage that a priori knowledge of categories is not required, and so the categorization process is unsupervised. The results of clustering could then be used to automatically formulate queries and search for other similar documents on the Web.

Automatic clustering techniques offer several advantages over a manual grouping process. Firstly, a clustering program can apply a specified objective criterion consistently to form groups. Human beings are excellent cluster seekers in two dimensions, but different individuals do not always identify the same cluster in the data. Secondly, a clustering algorithm can form the groups in a fraction of the time required by a manual grouping, particularly if a long list of descriptors or features are associated with each object. The speed, reliability, and consistency of a clustering algorithm are a good reason to use them.

There are many algorithms for automatic clustering like the *K Means algorithm* [Hartigan and Wong 1979], *Expectation Maximization* [Dempster et. al.1977] and *hierarchical clustering* [Jain and Dubes, 1988] which can be applied to a set of vectors to form the clusters. All statistical methods require vectors as input. So to cluster documents, it is required that these documents be represented as vectors. Traditionally the document is represented by the frequency of the words that make up the document. Different words are then given importance according to different criteria like *Inverse Document frequency* and *Information Gain*. These methods consider the document as a bag of words, and does not exploit the relations that may exist between the words. One of the shortcoming of these methods is due to *polysemy* or *homography* where a word has different meanings or meaning shades in different contexts (for example, the word *bank* in *He went to the bank to withdraw some money* and *The boat was beside the bank)*. It has been shown [Gonzalo et. al. 1998] that if we index words with their wordnet synset or sense then it improves the information retrieval performance. The frequency-based methods do not consider the structure of the sentences, which may also cause some problems (this has been discussed in section 2.3). Our work tries to improve the clustering accuracy by using the semantic information of the sentences representing the document. It uses the UNL representation of a document, which presents the information given in a document in the form of a semantic graph.

In section 2 we describe the methods for creation of document vector using the term frequency. Section 3 describes the Universal Networking Language, a semantic representation of documents, which depicts the document in the form of a graph. Section 4 describes the Self Organizing Map- a neural network model- which maps high dimensional data into a two dimensional map. Section 5 and 6 give the methods for document vector creation using the UNL graph link count and UNL relation label weightage scheme respectively. Section 7 is on the evaluation of the methods, describing the experimental setup. Section 8 discusses the results.

2 Document Representation using Term Frequency

The most common method for document representation considers the document as a bag of words and forms the document vector using the frequency count of each word in the document. The size of the vector is the total number of distinct words in the whole set of documents to be clustered. Some of the methods using the term count are described in the next subsection.

2.1 Vector Space Model

The basic method of representing a document is by considering it an element in a vector space. Each component of the vector is the frequency of occurrence of a word in the document. The size of the vector can be reduced by selecting a subset of most important words according to some criterion. It is, however, a difficult problem to find a suitable subset of words that still represents the essential characteristics of the documents. It is also important to remove the words which are not informative, hence most common words like *and, with, to etc.*, which are also known as *stop words*, are removed from the text while creating the vector.

2.2 Word Category Maps

In the *Self-organizing semantic map* [T. Kohonen, 1995] method the words are clustered onto neighboring grid points of a *Self Organizing Map*. Synonyms and closely related words are often mapped onto the same grid point or neighboring grid points. In this sense this clustering scheme is even more effective than the thesaurus method in which sets of synonyms are found manually. The input to the *self organizing map* consists of adjacent words in the text taken over a moving window. In making the word category map, all the words from all the documents are input interactively a sufficient number of times. After this, each grid point is labeled by all those words, the vector of which are mapped to that point. The grid points usually get multiple labels. To create a *Vector* for a document, the words of the document are scanned and counted at those grid points of the SOM that were labeled by that word.

The above two methods define a basic approach which can be used for representing a set of documents in the form of a vector. To improve the performance of clustering, the frequency count of different words are also weighted. A comparative evaluation of feature selection methods for text documents is done by Yang and Pedersen [Yang and Pedersen 1997]. A brief description of some of the methods is given below.

Inverse Document Frequency: Inverse Document Frequency for a given word is defined as the logarithm of the ratio between the total number of document in the corpus and the number of the documents which contain the word. The assumption behind this evaluation is that the word which occurs in a large number of documents does not help in the clustering of the document, while the word which occurs in few documents can be more informative in the clustering. The mathematical formula for inverse document frequency for a term *t* is:

$$IDF(t) = \log (n/n_t),$$

Where, *n* is the total number of documents in the corpus and n_t denotes the total number of documents which contain the term *t*.

Information Gain: Information gain is frequently employed as a term goodness criterion in the field of machine learning. It measures the number of bits of information obtained for category prediction. This is done by detecting the presence or absence of a term in a document. If c_i, $(i = 1..m)$ denotes the set of categories in the target space, then information gain of the term *t* is defined to be:

$$G(t) = -\sum_{i=1}^{m} P_r \log(P_r(c_i)) + P_r(t)\sum_{i=1}^{m} P_r(c_i \mid t)\log(P_r(c_i \mid t))$$

Given a training corpus, for each unique term we compute the information gain, and remove from the feature space those terms which have information gain less than some predetermined threshold.

Term Strength: Term strength method estimates the term importance based on how commonly a term is likely to appear in *closely-related* documents. It uses a training set of documents to derive document pairs whose similarity (measured using the cosine value of the two document vectors) is above a threshold. *Term strength* then is computed based on the estimated conditional probability that a term occurs in the second half of a pair of related documents given that it occurs in the first half. Let *x* and *y* be an arbitrary pair of distinct but related documents, and *t* be a term, then the strength of the term is defined to be:

$$s(t) = P_r(t \in y \mid t \in x)$$

Mutual Information: Mutual information is a criterion commonly used in statistical language modeling of word associations and related applications. If one considers the two way contingency table of a term *t* and a category *c*, where *A* is the number of times *t* and *c* co-occur, *B* is the number of times *t* occurs without *c*, *C* is number of times *c* occurs without *t*, and *N* is the total number of documents, then the mutual information criterion between *t* and *c* is defined to be:

$$I(t,c) = \log\left(\frac{P_r(t \wedge c)}{P_r(t) \times P_r(c)}\right)$$

and is estimated using

$$I(t,c) = \log\left(\frac{A \times N}{(A+C) \times (A+B)}\right)$$

I(t, c) has a value of *0* if *t* and *c* are independent.

2.3 Shortcomings of the Frequency Based Approach

If we consider the problem of document clustering, the representation of the document should be such that similar documents should also be similar in the representation. The representation of documents must reflect the knowledge meant to be conveyed by the documents. The above methods for representation of documents do not consider the semantic relations of the words. This may cause problems in many cases. Some combination of sentences, which have the same set of words having different meanings, should fall in different clusters. For example, if we consider the two sentences *John eats the apple standing beside the tree* and *The apple tree stands beside John's house,* they have the same set of words but talk about entirely different things. On the other hand there may be some sentences which have the same meaning but have been constructed from different sets of words. This case arises when synonymous words are used in the sentences. For example in the sentences, *John is an intelligent boy* and *John is a brilliant lad* mean more or less the same thing. There are some methods like Latent Semantic Indexing [Deerwester et. al. 1995] which try to solve it. The word category map method can also be used for the same purpose.

Another problem of frequency-based approach is that for a document even a word, which has a relatively lower frequency of occurrence in the document, can be more accurate in describing the document, whereas a word, which occurs more frequently, may have less importance. Frequency based methods do not take this into account. For solving the above problems we need to consider the semantic as well as the syntactic information present in the documents.

In this paper we describe a **new method for the creation of document vectors**. This approach uses the Universal Networking Language (UNL) representation of a document. The UNL represents the document in the form of a semantic graph with universal words (explained in the next section) as nodes and the semantic relation between them as links. Instead of considering the documents as a bag of words we use the information given by the UNL graph to construct the vector.

3 Universal Networking Language

Universal Networking Language (UNL) [Uchida, Zhu and Della 1995] is a semantic representation of a document, which expresses the document in the form of a graph. Information written in a natural language may be enconverted to UNL and the UNL can be deconverted into a target natural language. The UNL representation defines a semantic net [Woods 1985] like structure. The meaning is represented sentence by sentence in the form of a hyper graph having concepts as nodes and relations as directed arcs. Concepts are represented as character-strings called *Universal Words (UWs).*The knowledge within a document is represented in three dimensions:

1. Word Knowledge is expressed by **Universal Words (UWs)**, which are language independent. These UWs are restricted using constructs, which describe the sense of the word in the current context. For example, *drink(icl>liquor)* signifies that in the current context *drink* is a noun, which is a type of *liquor*. Here, *icl* stands for inclusion. *icl* restriction forms an *is a* kind of relationship that is defined for semantic nets.
2. Conceptual Knowledge is captured by relating different universal words using the standard set of UNL **Relation Labels**. For example, *Humans are an intelligent species* is described as:

mod(species(icl>group),intelligent(icl>quality))
aoj(species(icl>group), human(icl>animal))

Here, *aoj* means agent with an attribute and *mod* restricts the scope of the entity specified as the first Universal word *(species(icl>biological taxonomical group))* (i.e., a restricted kind of *species* which is *intelligent).*

3. Speakers *view, aspect, tense of a verb, number of a noun etc.* are captured by UNL **Attributes**. For example, consider the sentence *please come here*. The UNL representation for this sentence is:

plc(come(icl>do).@present.@request.@entry, here(icl>relative place))

Here, *.@request* describes the speaker's intention when he says *please*, *.@present* means the *present tense* and *.@entry* is a special attribute indicating the predicate of the sentence from which the sentence generation can begin.

All relations in UNL are binary. A binary relationship between Universal words is defined by *rel(UW1, UW2)*. As an example, consider the sentence

John, who is the chairman of the company, has arranged a meeting at his residence

the UNL representation of the sentence is:

```
;======================== UNL ========================
;John who is the chairman of the company has arranged a meeting at his residence.
[S]
mod(chairman(icl>post):01.@present.@def,company(icl>institution):02.@def)
aoj(chairman(icl>post):01.@present.@def,  John(icl>person):00)
agt(arrange(icl>do):03.@entry.@present.@complete.@pred,John(icl>person):00)
pos(residence(icl>shelter):04,        John(icl>person):00)
obj(arrange(icl>do):03.@entry.@present.@complete.@pred,meeting(icl>conference):05.@indef)
plc(arrange(icl>do):03.@entry.@present.@complete.@pred,residence(icl>shelter):04)
[/S]
;=====================================================
```

The UNL graph for the sentence is given in figure 1.

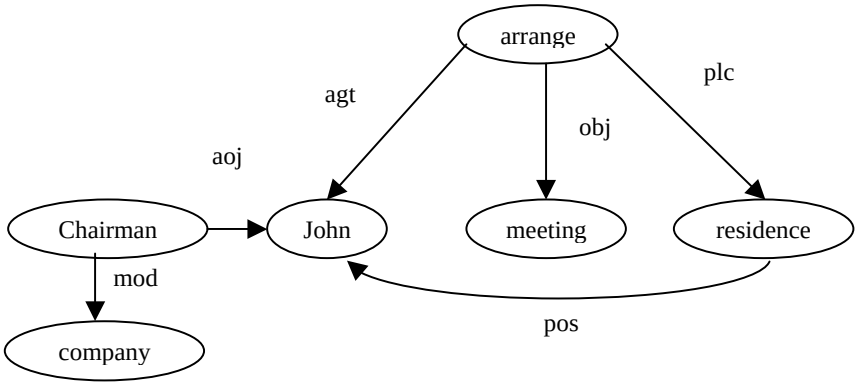


Figure 1: **An Example UNL Graph**

The main predicate or verb of the sentence is *arrange*, the agent (**agt**) of arrange is *John*, what *John* arranges (**obj**) is the meeting, the place where this event takes place (**plc**) is *John's residence* (**mod**) and finally the clausal qualifier for *John* is that he is the *chairman* (**aoj**) of the company (**mod** again).

4 Self Organizing Maps

The Self Organizing Maps (SOM) [T Kohonen 1995] is a general unsupervised learning method for ordering high dimensional data so that like inputs are in general mapped close to each other. It consists of a finite set of reference vectors that approximate the open set of input data. The main applications of the SOM are thus in the *visualization of complex data in a two-dimensional display* and *creation of abstractions* like in many clustering techniques. The SOM defines a mapping from the input data space R^n onto a two-dimensional array of nodes. With every node *i*, a parametric *reference vector* $M_i = <m_{i1}, m_{i2}, …., m_{in}> \in R^n$ is associated. The lattice type of the array can be defined to be rectangular, hexagonal, or even irregular. In the simplest case, an input vector $X_i = <x_{i1}, x_{i2}, …, x_{in}> \in R^n$ is connected to all neurons in parallel via variable scalar weights m_{ij}, which are in general different for different neurons. Let $x \in R^n$ be a stochastic data vector. One might then say that the SOM is a *nonlinear projection* of the probability density function *p(x)* of the high-dimensional input data vector *X* onto the two-dimensional display. Vector *x* may be compared with all m_i in any metric, in many practical applications, the smallest of the *Euclidean distance* $||x-m_i||$ can be made to define the *best-matching node*, signified by the subscript *c*:

$$c = arg\ min_i\ ||\ x - m_i\ ||$$

which means the same as

$$||x - m_c|| = min_i\ ||\ x - m_i\ ||$$

During *learning*, or the process in which the *non-linea*r projection is formed, those nodes that are *topographically close in the array up to a certain geometric distance* will activate each other to learn something from the main input X_i. This results in a local relaxation or *smoothing effect* on the weight vectors of the neurons in the neighborhood, which in continued learning, leads to *global ordering*. The learning law for the SOM is given by

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)]$$

where, $t = 0,1,2, \ldots$ is an integer, the discrete-time coordinate. The initial values $m_i(0)$ can be arbitrarily chosen. In the relaxation process, the function $h_{ci}(t)$ acts as the *neighborhood function*, a smoothing kernel defined over the lattice points. For convergence it is necessary that $h_{ci}(t)$ should go to zero when *t* goes to infinity. Usually

$$h_{ci}(t) = h(||\ r_c - r_i\ ||, t),$$

where, $r_c \in R^2$ and $r_i \in R^2$ are the location vector of nodes *c* and *i*, respectively, in the array. With increasing $||\ r_c - r_i||$, $h_{ci}(t)$ tends to *zero*. The average width and form of $h_{ci}(t)$ defines the *stiffness* of the *elastic surface* to be fitted to the data points. Two simple choices for $h_{ci}(t)$ occur frequently. The simpler of them refers to a *neighborhood set* of array points around node *c*. So $h_{ci}(t) = \alpha(t)$ if *i* is in the neighborhood else $h_{ci}(t) = 0$. Here $\alpha(t)$ denotes the *learning-rate factor* $(0 < \alpha(t) < 1)$. Both $\alpha(t)$ and the radius of *neighborhood set* are usually decreasing monotonically in time (during the ordering process). Another widely applied, smoother neighborhood kernel can be written in terms of the Gaussian function.

$$h_{ci}(t) = \alpha(t).\exp\left(-\frac{||\ r_c - r_i\ ||}{2\sigma^2(t)}\right)$$

where $\sigma^2(t)$ defines the width of the kernel which corresponds to the radius of the *neighborhood set* above.

By its very nature, SOM is a clustering algorithm. The basic idea is that the weights of a neuron get closer an closer in terms of Euclidian distance to an input vector or a set of vectors which form a cluster. *Thus the neuron through its associated weights becomes a representative for the cluster*. As can be seen from the above discussion of the basic ideas of the Kohonen's algorithm, it is essentially a distance minimization technique using the parameters, α the learning rate and $h_{ci}(t)$ the neighborhood function. In our problem, the documents are represented as real valued vectors constructed through various methods capturing frequency, referral importance and semantic relations. These are clustered through the self organization maps.

5 Document Vector Construction Using UNL Graph Links

In the UNL link method, instead of using the words as components for the document vector we use the Universal Words as the components of the vector. Since each UW is disambiguated, multiple words in the document get automatically differentiated, thereby producing correct frequency count. For example in the sentence,

The Commercial Bank is situated on the bank of the river

The word *bank* has two different senses, *viz., bank(icl>financial institute)* and *bank(mod>river).* Hence, the frequency count of 2 is wrong for this word. Since the UNL based method works with UWs, this mistake will not be committed. They find different places in the document vector. After this, each component of the document vector- which represents a different universal word (*i.e.*, a concept) is assigned the number of links incident on the node, considering the graph to be undirected. When a UW is not present in the UNL graph of the document then 0 is written in its position. The basic assumption behind this approach of counting the links is that *the more number of links to and from a universal word, the more is the importance of the word in the document.*

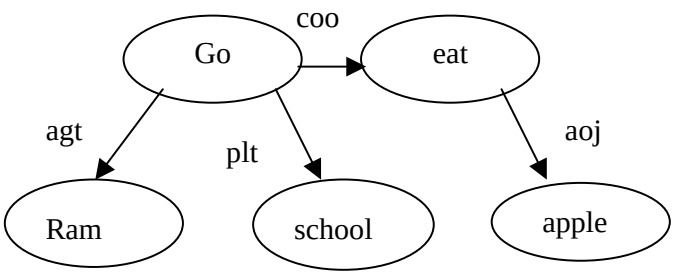


Figure 2: UNL graph of the sentence *Ram is going to the school eating an apple.*

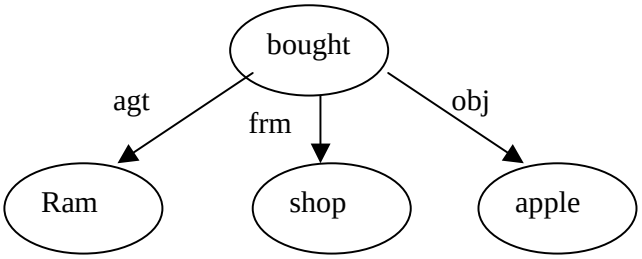


Figure 3: UNL graph of the sentence *Ram bought the apple from the shop*

For example consider the two sentences given in figures 2 and 3 as given documents (*Ram* is a typical Indian name). The vectors corresponding to the graphs are:

$$\mathbf{X_1 = <1, 3, 1, 2, 1, 0, 0>,}$$

and

$$\mathbf{X_2 = <1, 0, 0, 0, 1, 3, 1>}$$

Here, the words considered are *Ram, go, school, eat, apple, bought* and *shop* in that order. The numbers for each word in the vectors represent the number of the links that are incident on the word. For example, the first number in the vector $\mathbf{X_1}$ is 1, since the UW *Ram* is only connected to the UW *go* in the UNL graph of the first sentence. Similarly the last place, which indicates the number of links incident on the UW *shop*, is 0, as the UW is not present in the graph.

The process of construction of the document vector from the UNL representation is described:

1. Parse the UNL document to construct the UNL graph.
2. For each UW in the UNL graph count the links to other UWs from it.
3. Construct the feature vector by merging the counts got from step two.
4. Output the feature vector.

The main benefit of this approach is that, it not only takes into account the frequency of the words in the document but also adds some extra information about the sentence structure by giving more weightage to the important words in the sentence. Consider the following sentences:

1. *Ram goes to the bank.*
2. *Shyam goes to the market.*

(*Ram* and *Shyam* are common Indian names)

When we compare the two sentences, their vectors using the UNL link method are **<1, 2, 1, 0, 0>** and **<0, 2, 0, 1, 1>** respectively. The words considered here are *Ram, goes, bank, Shyam,* and *market*. The cosine similarity of the sentences comes out to be 0.66. If we compare them using the term frequency method, the vectors are **<1, 1, 1, 0, 0>** and **<0, 1, 0, 1, 1>,** and the similarity comes out to be 0.33. Since both the sentences describe the event of somebody going somewhere and their similarity value should be high which the UNL link method achieves.

It is important to note that the UNL link approach does not lose any information given by the word frequency method, since the method implicitly incorporates the frequency of the UWs. *For any node in the graph there is at least one link incident on it.*

One other advantage of using UWs instead of simple words as the components of the document vector is that they also capture the meaning of the word according to its usage in the context. For example the word *crane* with the sense *bird* has the UW *crane(icl>bird)* which is different from the UW *crane(icl>machine)* for *crane* meaning a type of *machine*. This solves the problem of both occurrences of words being considered as the same. Thus the UNL link method will not give any similarity between the sentence *The crane was eating fish* and the sentence *The crane lifted the load* whereas the term frequency method will give some finite similarity to them.

6 Document Vector Construction Using UNL Relation Labels

The UNL link method does not consider the label of the links in the graph. Two different relations, for example, *agt* (agent) and *man* (manner) give the same importance to the UW, *i.e.,* one. This differentiation is necessary. Consider two examples:

1. *John saw Jack*
2. *Jack saw John*

Both term frequency and UNL link methods give similarity value of *1.0* for the documents. But the meanings of the documents are different and hence the similarity value should be less that *1.0.* This is achieved if the labels on the links are also taken into account.

To incorporate the relation information in the document vector we give different weights to different relations in the UNL graph. The weights can be given either manually according to the relation type or can be *learned* from the given set of example documents. The next paragraph describes the method by which we incorporate the relation labels in the document labels.

In this method, instead of a single dimensional vector we construct a two dimensional matrix *M* of dimension *n* x *n*, where *n* is the total number of UWs in the corpus encompassing all documents The element m_{ij} of the matrix denotes the value of the weight assigned to the label of the link connecting the UWs, UW_i and UW_j or a value of *0* if there is no link between the two UWs.

The matrix corresponding to the UNL graph in figure 2 is:

$$M_1 = \begin{bmatrix} Ram & go & school & eat & apple & bought & shop & \\ 0 & wt_{agt} & 0 & 0 & 0 & 0 & 0 & Ram \\ wt_{agt} & 0 & wt_{plt} & wt_{coo} & 0 & 0 & 0 & go \\ 0 & wt_{plt} & 0 & 0 & 0 & 0 & 0 & school \\ 0 & wt_{coo} & 0 & 0 & wt_{aoj} & 0 & 0 & eat \\ 0 & 0 & 0 & wt_{aoj} & 0 & 0 & 0 & apple \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & bought \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & shop \end{bmatrix}$$

and the matrix for the UNL graph in the figure 3 is:

$$M_2 = \begin{bmatrix} Ram & go & school & eat & apple & bought & shop & \\ 0 & 0 & 0 & 0 & 0 & wt_{agt} & 0 & Ram \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & go \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & school \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & eat \\ 0 & 0 & 0 & 0 & 0 & wt_{obj} & 0 & apple \\ wt_{agt} & 0 & 0 & 0 & wt_{obj} & 0 & wt_{plt} & bought \\ 0 & 0 & 0 & 0 & 0 & wt_{plt} & 0 & shop \end{bmatrix}$$

The matrix thus formed can be given to the SOM (which takes a single dimensional vector) in many ways, the simplest way is to represent the matrix by a single dimensional vector by considering the rows one by one. The *X* vector for the above matrix is

$$\mathbf{X_1 = < 0, wt_{agt}, 0, 0, 0, 0, 0, wt_{agt}, \ldots, 0, 0>}$$

The problem with this simple representation is that the vector length becomes very large. To make the feature vector we add up all the column of the matrix to form a single dimension vector of size equal to the number of distinct Universal words in the whole corpus.

6.1 Assigning Weights to Labels- Learning Based

When we use the weight matrix to form the representation, one of the problems we face is how to decide the value of weights for each of the different labels. One of the properties of the assigned weights should be that they are proportional to the information a particular relation gives to the Universal word which leads to effective clustering of the documents. The essential idea of the method, described below, is to *find how a particular relation when used alone, clusters the document.*

To find the weights of the relations we took a set of documents, which had already been clustered manually. For each relation label, we formed new graphs representing the documents by removing all the links whose label is other than the label of which we want to find the weight. After doing this we apply the UNL link method to the new graph representation of the documents and find the document vectors. In the next step, we cluster the documents using these vectors and find the accuracy of the clustering. The accuracy is defined here as the ratio between the number of correctly classified documents using this representation and total number of documents. The weight of the relation is then proportional to this accuracy (In the experiments we took the proportionality constant to be *1*). The assumption behind this approach is that higher *importance must be given to the relation, which has the capacity to classify more correctly, all by itself, i.e., to the discriminating power of the relation*.

For example, if we consider the relation *agt*, the modified graph for the UNL representation shown in figures 2 and 3 are as shown in figures 4 and 5 respectively, and the vector formed from these graphs are,

$$\mathbf{X_1=<1, 1, 0, 0, 0, 0, 0>,}$$

and,

$$\mathbf{X_2=<1, 0, 0, 0, 0, 1, 0>,}$$

Here the words taken are *Ram, go, school, eat, apple, bought* and *shop.*

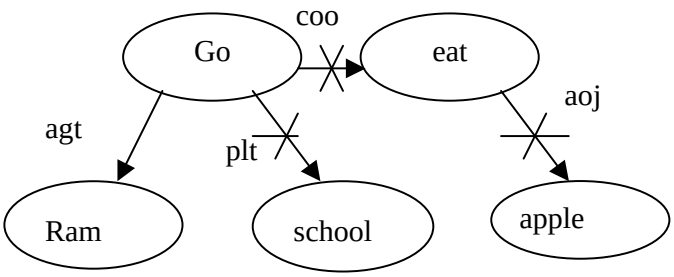


Figure 4: UNL graph of *Ram is going to the school eating an apple* considering only *agt*

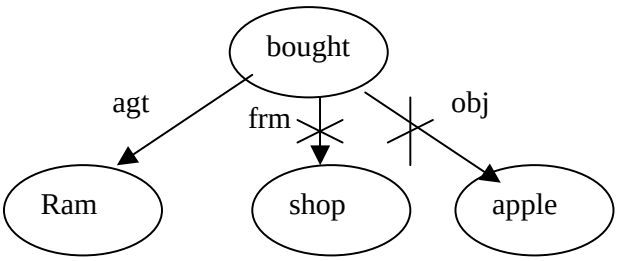


Figure 5: UNL graph of *Ram bought the apple from the shop* considering only *agt*

The steps for finding the weight values for the different relations are:

1. For each relation do
    a. Construct a new UNL graph for the documents by removing all links with labels other than the current relation label.
    b. Construct the feature vector using the UNL link method on the new graphs.
    c. Cluster the documents and find the accuracy of the clustering.
2. Assign the weight for the relation labels using the formula.

    Wt(rel) = Accuracy of the clustering using *rel*

    Where accuracy of clustering is given as

$$\frac{m}{n}$$

    m denoting the number of documents correctly classified and *n* denotes the total number of documents.

The table bellow enumerates the weights for the *37* UNL relations used in the corpus.

Label	Weight	Label	Weight
Agt	0.787143	or	0.682857
and	0.907143	per	0.584286
aoj	0.930000	plc	0.728095
bas	0.668572	plt	0.636364
ben	0.584286	pof	0.636364
cnt	0.783809	pos	0.584286
cob	0.636364	ptn	0.636364
con	0.732857	pur	0.635238
coo	0.699524	qua	0.584286
dur	0.662857	rsn	0.636364
fmt	0.668572	scn	0.778095
frm	0.654286	seq	0.604286
gol	0.699524	src	0.636364
lpl	0.584286	tim	0.617619
man	0.584286	tmf	0.636364
met	0.736191	tmt	0.636364
mod	0.710476	to	0.636364
nam	0.631905	via	0.571429
obj	0.825238		

Table 1: Weight values for different relation labels in the Corpus.

These results were found out by averaging 5 iteration of weight calculation. Each iteration had different subsets of the whole corpus.

7 Evaluation

Vectors of documents were created using the term frequency, the UNL link and the UNL relational label methods. Then they were clustered using the Self Organizing Maps. The neurons were labeled using the majority approach, *i.e.*, if most of the documents assigned to a neuron belong to the cluster *C*, then the label of the neuron is designated as *C*. After the self organization process, the neurons get labeled and we know the classes of the documents. Then comparing the actual classes with the SOM

found classes we can obtain the number of documents correctly clustered. The accuracy of clustering is given by,

$$Accuracy = \frac{Number\ of\ documents\ correctly\ clustered}{Total\ number\ of\ documents.}$$

7.1 Experimental Setup

Total number of documents: 26
Total number of clusters: 3
Documents in cluster 1: 14
Documents in cluster 2: 8
Documents in cluster 3: 4

One sample document from each cluster is given bellow:

Sample 1: Sample document from the first cluster.

Knowledge of other cultures is essential for establishing a constructive dialogue between different communities. This knowledge implies reflection about the common ground between all individuals as well as the qualities that differentiate them. Therefore, the only way to achieve a meaningful dialogue is through the acceptance of the identities of others, with their particularities, yet without renouncing one's own. Based on this premise, the Universal Forum of Cultures will offer an opportunity to celebrate the elements that differentiate us and to confront the bigotry, intolerance and mistrust that threaten to turn these differences into sources of conflict. The Forum strives to foster the kind of understanding and respect capable of increasing both our appreciation of our human environment and our ability to work together to make the world a better place.

Sample 2: Sample document from the second cluster.

The Council acts on behalf of the PPC in the interval of Plenipotentiary Conferences. It considers broad telecommunications policy issues, prepares reports on the policy and strategic plan for ITU, exercises financial control, ensures coordination of the network of ITU, and approves its biennial budgets. In addition, the Council is responsible for ensuring the smooth day-day running of ITU, coordinating work programmes, and controlling finances and expenditures. It also takes all the steps to facilitate the implementation of the provisions stemming out of treaties, conventions and other regulation-settings approved by the Plenipotentiary Conference and other conferences.

Sample 3: Sample document from the third cluster

The Fulton County Grand Jury said on Friday that an investigation of Atlanta's recent primary election produced no evidence that any irregularities took place. The jury further said in term end presentments that the City Executive Committee which had over-all charge of the election, deserves the praise and thanks of the City of Atlanta for the manner in which the election was conducted The September October term jury had been charged by Fulton Superior Court Judge Durwood Pye to investigate reports of possible irregularities in the hard fought primary which was won by Mayor nominate Ivan Allen Jr.

UNL Expressions for the first sentences from the samples are shown below:

Knowledge of other cultures is essential for establishing a constructive dialogue between different communities.

aoj(essential(mod<thing).@entry, knowledge.@def)
mod(knowledge.@def, culture(icl>abstract thing).@pl)
mod(culture(icl>abstract thing).@pl,other(mod<thing))
pur(essential(mod<thing).@entry, establish(icl>do))
obj(establish(icl>do),dialogue(icl>event))
mod(dialogue(icl>event),constructive(mod<thing))
scn(establish(icl>do),between(icl>how))
obj(between(icl>how),community(icl>group).@pl)
mod(community(icl>group).@pl, different(icl>various))

Council acts on behalf of the PPC in the interval of Plenipotentiary Conferences.

agt(act(icl>do).@entry, Council(pof>International Telecommunication Union))
man(act(icl>do).@entry, on behalf of(icl>how))
obj(on behalf of(icl>how), PPC(icl>Plenipotentiary Conference).@def)
tim(act(icl>do).@entry, Plenipotentiary Conference.@pl)
mod(Plenipotentiary Conference.@pl, interval(icl>period))

The Fulton County Grand Jury said on friday an investigation of Atlanta's recent primary election produced no evidence that any irregularities took place.

obj(say(icl>do):0C.@entry.@past.@pred, :02)
aoj(say(icl>do):0C.@entry.@past.@pred, Fulton_County_Grand_Jury(icl>group):04.@def)
tim(say(icl>do):0C.@entry.@past.@pred, friday(icl>daytime):0D)
aoj:02(produced(icl>happen):2A.@entry.@past.@pred, investigation(icl>inquiry):1.@indef)
obj:02(produced(icl>happen):2A.@entry.@past.@pred, evidence(icl>information):3D)
aoj:02(no(icl>nary):38, evidence(icl>information):3D)
aoj:02(:01, evidence(icl>information):3D)
aoj:01(took_place(icl>happen):48.@entry.@past.@pred, irregularity(icl>misbehavior):3A)
aoj:01(any:3, irregularity(icl>misbehavior):3A)
mod:02(investigation(icl>inquiry):1.@indef, primary_election(icl>election):2B)
pos:02(primary_election(icl>election):2B, atlanta(icl>state capital):1O)
aoj:02(recent(icl>past):26, primary_election(icl>election):2B)

The UNL expressions were parsed to form the UNL graphs which gave the link counts of nodes. The vectors formed by the three methods are shown below.

Vectors by TF method.	
Dimension :	35
Sentence 1 :	0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 1 0 1 0 1 1 1 1 0 0 1
Sentence 2 :	1 0 0 0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1
Sentence 3 :	0 0 1 1 0 1 0 0 1 1 1 1 1 0 1 0 0 0 1 0 1 1 1 1 0 1 0 0 0 0 0 0 1 0 1

Vectors by UNL link Method:	
Dimension :	34
Sentence 1 :	0 2 2 1 2 0 2 0 1 0 0 1 0 0 0 2 0 0 0 0 2 0 1 0 0 0 0 2 1 0 0 0 0 3
Sentence 2 :	0 0 0 0 0 2 0 0 0 0 0 0 0 0 2 0 0 0 1 0 0 0 0 0 1 0 0 0 0 3 1 0 0 0
Sentence 3 :	1 0 0 0 0 0 0 2 0 3 1 0 1 1 0 0 2 1 0 3 0 1 0 1 0 3 1 0 0 0 0 1 2 0

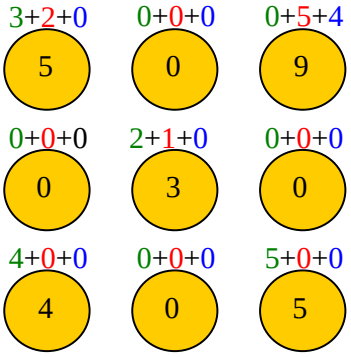
Vectors by UNL Relation Method:

Dimension :34
Sentence 1 :0 1.494 1.420 0.783 1.535 0 1.565 0 0.710 0 0 0.710 0 0 0 1.535 0 0 0 0 1.640 0 0.710 0 0 0 0 1.603 0.710476 0 0 0 0 2.238
Sentence 2 :0 0 0 0 0 1.409 0 0 0 0 0 0 0 0 1.328 0 0 0 0.825 0 0 0 0 0 0.710 0 0 0 0 1.989 0.787 0 0 0
Sentence 3 : 0.617 0 0 0 0 0 0 1.755 0 2.224 0.825 0 0.93 0.584 0 0 1.86 0.93 0 2.685 0 0.93 0 0.93 0 2.372 0.93 0 0 0 0 0.93 1.640 0

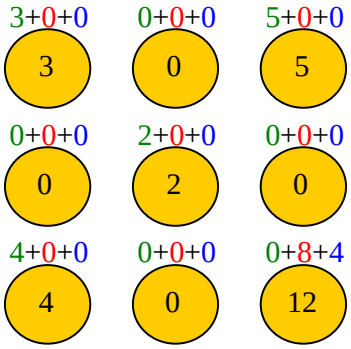
The Clustering Step:

The dimension of the vector created by TF method for the whole of the twenty-six documents was *1025* and the dimensions of the vectors created by the UNL methods were *1255*. The vectors were then input to a Self Organizing Map of 9 neurons organized as a 3 x 3 grid. The SOM was initialized randomly and was trained with the vectors in 2 steps- first for organizing the map (*10,000* iteration) and then for fine-tuning (*1,000,000* iterations) it. The value for α and h_c were *0.5* and *2.0* respectively in the organization step and *0.1* and *2.0* respectively in the fine-tuning step.

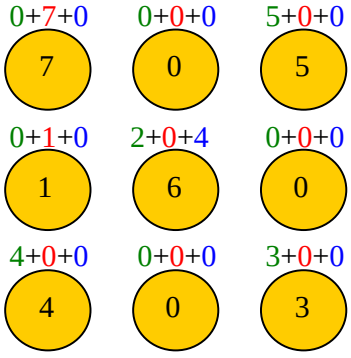
The output of the SOM corresponding to the TF, UNL link and UNL relation method are shown in figures 6(a), 6(b) and 6(c) respectively. The *nine* circles in the figures denote the nine neurons of the 3 x 3 SOM. The number inside the circle denotes the number of documents that were assigned to the neuron *after the self organization process.* The numbers above the circles (n_1 + n_2+ n_3) represent the number of documents of class 1, 2 and 3 respectively assigned to that neuron. For example 3+*2+0* above the first circle in figure 6(a) indicates that 3 documents belonging to the first cluster, 2 documents belonging to second cluster and no documents from the third cluster were mapped to that neuron.



(a) **TF method**



(b) **UNL Links method**



(c) **UNL Relation method**

Figure 6: The different Self Organizing Maps

8 Discussion of Results

We denote the neurons by the tuple *(row number, column number)* with *row number* increasing from bottom to top and the *column number* increasing from left to right. As seen in figure 6(a), using the term frequency method the documents of clusters 1 are distributed to neurons (1,1), (1,3), (2,2) and (3,1), while those of cluster 2 are given to (3,1), (2,2) and (3,3). The documents of cluster 3 go to (3,3) only. By the majority rule the labels of neurons are as follows:

Neuron	Label
(1,1)	Cluster 1
(1,2)	Unlabeled
(1,3)	Cluster 1
(2,1)	Unlabeled
(2,2)	Cluster 1
(2,3)	Unlabeled
(3,1)	Cluster 1
(3,2)	Unlabeled
(3,3)	Cluster 2

Now it is apparent that 2+1 documents of cluster 2 and all 4 documents of cluster 3 are wrongly mapped. Hence the accuracy is 19/26 which is 0.730769.

When we consider the UNL link method, figure 6(b) shows that only the 4 documents of cluster 3 are wrongly mapped to the neuron for cluster 2 at (1,3). All 8 documents of cluster 2 are together. The documents of cluster 1- which is big- is distributed to 4 neurons, probably because of intra document differences in spite of being from the same cluster. The accuracy here is seen to be 22/26 which is 0.846154.

Coming to the last method of UNL relation labels, figure 6(c) shows that the distribution of cluster 1 documents are same as before. However, cluster 2 documents stand independently in two neurons. But the good thing is that the cluster 3 now has got an independent neuron label. The number of wrongly clustered documents is only 2 giving, thus, an accuracy of 24/26 which is 0.923077. All the accuracy values are tabulated in table 2.

Method	Accuracy
Term Frequency	*0.730769*
UNL Link	*0.846154*
UNL Relation	*0.923077*

Table 2: Accuracy of different methods

9 Conclusion

We have proposed a new method for text clustering. This method uses the semantic information present in the form of relations between words in sentences. Thus the approach is different from traditional methods of clustering which consider the document as a bag of words. As shown in the experiments, this approach performs better than the methods based on only frequency. While we have used Self Organizing Maps as the clustering algorithm, any other clustering algorithm which takes a vector as input can be adopted.

The problem of text clustering is a very important one and poses challenges to the information retrieval community. All possible help in the form of lexical, syntactic and semantic knowledge should be taken for addressing this task which finds application in better web searching, question answering and indexing.

References

A.P. Dempster, N.M. Laird, and D.B. Rubin. *Maximum Likelihood from Incomplete Data via the EM Algorithm*. Journal of the Royal Statistical Society, Series B (Methodological) , 39(1):1--38, 1977.

Julio Gonzalo, Felisa Verdejo, Irina Chugur, Juan Cigarran *Indexing with WordNet synsets can improve Text Retrieval*, Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet for NLP, Montreal. 1998.

J. A. Hartigan and M. A. Wong. *A k-means clustering algorithm.* Applied Statistics, 28:100--108, 1979.

A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs NJ, U.S.A., 1988.

Kohonen T. *Self-organizing Maps*, Series in Information Sciences, vol. 30, Springer, 1995.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman *Indexing by Latent Semantic Analysis.* Journal of the American Society of Information Science 1990.

Yang Y., Pedersen J.O. *A Comparative Study on Feature Selection in Text Categorization*, Proc. of the 14th International Conference on Machine Learning ICML 1997.

Uchida H., Zhu M., Della Senta T. *UNL: A Gift for a Millennium.* The United Nations University, 1995 *http://www.unl.ias.unu.edu/publications/gm/index.html*

Woods William A. *What's in a Link: Foundation for Semantic Networks.* in Readings in Knowledge Representation, R.J. Brachman and H.J.Levesque (ed.), Morgan Kaufmann Publishers, 1985.