[RANDOM WALK TERM WEIGHTING FOR IMPROVED TEXT CLASSIFICATION]

[Zahra Fathollahi] [third Homework of Information Theory]

[Highlights]

1. New approach for estimating term weights in a document

2. New weighting scheme can be used to improve the accuracy of a text classifier

3. The method uses term co-occurrence as a measure of dependency between word features

4. A random walk model is applied on a graph encoding words and co-occurrence dependencies, resulting in scores that represent a quantification of how a particular word feature contributes to a given context

[Random Walk Algorithms]

there are several random walk algorithms that have been proposed in the past, we focus on only one such algorithm, namely PageRank.

Given a graph G = (V,E), let In(Va) be the set of vertices that point to vertex Va (predecessors), and Out(Va) be the set of vertices that vertex Va points to (successors). The PageRank score associated with the vertex Va is defined using a recursive function that integrates the scores of its predecessors:

$$S(V_a) = (1-d) + d * \sum_{V_b \in In(V_a)} \frac{S(V_b)}{|Out(V_b)|}. \qquad (1)$$

Particularly relevant for our work is the application of random walks to text processing, as done in the TextRank system [16]. TextRank has been successfully applied to three natural language processing tasks: document summarization, word sense disambiguation, and keyword extraction, with results competitive with those of state-of-the-art systems.

This approach follows similar steps as used in the TextRank keyword extraction application, which derives term weights using a graph representation that accounts for the co-occurrence dependencies between words in the text.

[Random Walks for Term Weighting]

Starting with a given document, we determine a ranking over the words in the document by using the following models.

[Random walk models]

they experimented with several variations of PageRank that incorporate additional information and variables into the traditional version shown in Eq. (1). We summarize the best PageRank-based term ranking models as follows:

rwo : It represents the original model, as described in Eq. (1) in which we use an undirected graph with a constant damping factor that adheres strictly to the traditional formula of PageRank.

rwe.idf : This model representsan undirected graph approach that uses the weighted edge version of PageRank with a variable damping factor. The edge weight is calculated by the following formula:

$$E_{V_1,V_2} = tf.idf_{V_1} * tf.idf_{V_2}. \quad (2)$$

where EV1,V2 is the edge connecting V1 to V2, and tf.idf represents the term frequency multiplied by the inverse document frequency.

The damping factor is expressed as a function of the incoming edges' weight, calculated as follows:

$$d_{E_{V_1,V_2}} = E_{V_1,V_2}/E_{max}. \quad (3)$$

where dEV1,V2 is the damping function and Emax represents the highest weight for an edge in the graph. The resulting node ranking formula is:

$$S'(V_a) = \frac{(1-d)}{|N|} + \sum_{V_b \in In(V_a)} C * \frac{d_{E_{V_b,V_a}} * S(V_b)}{|Out(V_b)|}. \quad (4)$$

where N represents the total number of nodes in the graph, d is the damping constant, and C is a scaling constant.

The model biases the random walker toward nodes with stronger (higher weight) edges, as opposed to nodes with weaker (lower weight) edges.

rwe.oc: This model is similar to the above approach, however the damping factor for an edge is estimated in terms of the bigram co-occurrence frequency of the two nodes connected by the edge, Eq. (5). For example, if the bigram "free software" occurred four times in a document, then the weight of the edge connecting "free" and "software" is four.

$$E_{V_1,V_2} = tf(_{V_1V_2}). \quad (5)$$

[An example]

To understand why the rw weights might be a good replacement for the traditional tf weights, consider the example in Fig. 2, which models a sample document. Starting with this text, a graph is constructed as follows. If a term has not been previously seen, then a node is added to the graph to represent this term. A term can only be represented by one node in the graph. An undirected edge is drawn between two nodes if they co-occur within a certain window size. Figure 3 shows the graph constructed for this text, assuming a window size of 2, corresponding to two consecutive terms in the text (e.g. London is linked to based)

London-based sugar operator Kaines Ltd confirmed it sold two cargoes of white sugar to India out of an estimated overall sales total of four or five cargoes in which other brokers participated. The sugar, for April/May and April/June shipment, was sold at between 214 and 218 dlrs a tonne cif, it said.

Fig. 2. Sample document.

After the graph is constructed, the random walk model is applied on the graph, resulting in a set of scores associated with the vertices (words). Table 1 shows the tf and rw weights. By analyzing the rw weights, we can observe a non-linear correlation with the tf weights, with an emphasis given to terms surrounding important key terms such as sugar or cargoes. This spatial locality has resulted in higher ranks for terms like operator compared to other terms like London.

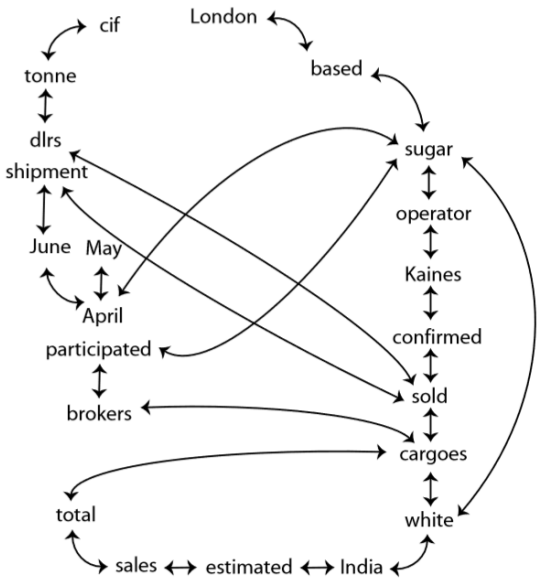

Fig. 3. Sample graph.

Table 1. tf & rw scores for a sample text.

Term	rw	tf	Term	rw	tf
sugar	16.88	3	participated	3.87	1
sold	14.15	2	april	3.87	2
based	7.39	1	india	1.00	1
confirmed	6.90	1	estimated	1.00	1
Kaines	6.81	1	sales	1.00	1
operator	6.76	1	total	1.00	1
London	4.14	1	brokers	1.00	1
cargoes	4.01	2	may	1.00	1
shipment	4.01	1	june	1.00	1
dlrs	4.01	1	tonne	1.00	1
white	3.87	1	cif	1.00	1

[Random Walk Algorithms]

Accuracy for different random walk models

	Tf	**Rw0**	Rweoc	Rwtfidf
NB LSpam	0.9785	0.8409	0.9090	0.9090
20NG	0.7284			
SVM LSpam	0.9772	0.9545	0.9545	0.9545
20NG	0.6456			
Rocchio LSpam	0.76	0.75	0.75	0.7272
20ng	0.7275			

Accuracy for the rwe.oc random walk model for different window sizes.

	Tf	**Rw2**	**Rw4**	Rw6	Rw8
NB LSpam	0.9785	0.9090	0.9090	0.9090	0.9090
20NG	0.7284				
SVM LSpam	0.9772	0.9545	1	1	1
20NG	0.6456				
Rocchio LSpam	0.76	0.75	0.75	0.75	0.75
20ng	0.7275				

Linspam (RWeidf)

Max Features: 7000 Window size: 2 Steps: 10

SVM Accuracy : 0.9545454545454546

SVM f1 micro : 0.9545454545454546

SVM f1 macro : 0.9545454545454545

NB Accuracy : 0.9090909090909091

NB f1 micro : 0.9090909090909091

NB f1 macro : 0.9089026915113871

clf f1 micro : 0.75

clf f1 macro : 0.7435082140964493

clf accuracy : 0.75

execute time: 1050.6639

rweoc

Max Features: 7000 Window size: 2 Steps: 10

SVM Accuracy : 0.9545454545454546

SVM f1 micro : 0.9545454545454546

SVM f1 macro : 0.9545454545454545

NB Accuracy : 0.9090909090909091

NB f1 micro : 0.9090909090909091

NB f1 macro : 0.9089026915113871

clf accuracy : 0.75

clf f1 micro : 0.75

clf f1 macro : 0.7435082140964493

execute time: 961.0126

Max Features: 7000 Window size: 4 Steps: 10

SVM Accuracy : 1.0

SVM f1 micro : 1.0

SVM f1 macro : 1.0

NB Accuracy : 0.9090909090909091

NB f1 micro : 0.9090909090909091

NB f1 macro : 0.9089026915113871

clf accuracy : 0.75

clf f1 micro : 0.75

clf f1 macro : 0.7435082140964493

execute time: 1072.5759

% RWeoc featureing : 97.73

Max Features: 7000 Window size: 6 Steps: 10

SVM Accuracy : 1.0

SVM f1 micro : 1.0

SVM f1 macro : 1.0

NB Accuracy : 0.9090909090909091

NB f1 micro : 0.9090909090909091

NB f1 macro : 0.9089026915113871

clf accuracy : 0.75

clf f1 micro : 0.75

clf f1 macro : 0.7435082140964493

execute time: 865.0672

rw0

SVM Accuracy : 0.9545454545454546

SVM f1 micro : 0.9545454545454546

SVM f1 macro : 0.9545454545454545

NB Accuracy : 0.8409090909090909

NB f1 micro : 0.8409090909090909

NB f1 macro : 0.8388278388278387

clf accuracy : 0.75

clf f1 micro : 0.75

clf f1 macro : 0.7435082140964493

execute time: 823.0385

20news

tf

```
Max Features: 7000   Window size: 2   Steps: 10

SVM Accuracy :  0.6456452469463622
SVM f1 micro :  0.6456452469463622
SVM f1 macro :  0.640391584858808
NB Accuracy :  0.7284917684545937
NB f1 micro :  0.7284917684545937
NB f1 macro :  0.7210369648953442
clf accuracy :  0.72755979819437068
clf f1 micro :  0.75755979819437068
clf f1 macro :  0.7512302844876394
execute time: 601.2344
```