LRD: Latent Relation Discovery for Vector Space Expansion and Information Retrieval

Technical Report KMI-06-09
March, 2006

Alexandre Gonçalves, Jianhan Zhu, Dawei Song,
Victoria Uren, Roberto Pacheco

In Proc. of The Seventh International Conference on Web-Age Information Management (WAIM 2006), June, Hong Kong, China.

LRD: Latent Relation Discovery for Vector Space Expansion and Information Retrieval

Alexandre Gonçalves[1], Jianhan Zhu[2], Dawei Song[2],
Victoria Uren[2], Roberto Pacheco[1,3]

[1] Stela Institute, Florianópolis, Brazil
{a.l.goncalves, pacheco}@stela.org.br
[2] Knowledge Media Institute, The Open University, Milton Keynes, United Kingdom
{j.zhu, d.song, v.s.uren}@open.ac.uk
[3] Department of Computing and Statistics, Federal University of Santa Catarina,
Florianópolis, Brazil
{rpacheco}@inf.ufsc.br

Abstract. In this paper, we propose a text mining method called LRD (latent relation discovery), which extends the traditional vector space model of document representation in order to improve information retrieval (IR) on documents and document clustering. Our LRD method extracts terms and entities, such as person, organization, or project names, and discovers relationships between them by taking into account their co-occurrence in textual corpora. Given a target entity, LRD discovers other entities closely related to the target effectively and efficiently. With respect to such relatedness, a measure of relation strength between entities is defined. LRD uses relation strength to enhance the vector space model, and uses the enhanced vector space model for query based IR on documents and clustering documents in order to discover complex relationships among terms and entities. Our experiments on a standard dataset for query based IR shows that our LRD method performed significantly better than traditional vector space model and other five standard statistical methods for vector expansion.

1 Introduction

Textual corpora, such as web pages on a departmental website and blogs of a group of people, often mention named entities which are related to each other, and their relatedness is often shown by their co-occurrence in the same documents and their occurring close to each other in these documents, e.g., one document mentions Thomas works on project X in one sentence, and another document mentions Jack works on X in one paragraph. Given an entity, we can use either standard statistical measures such as mutual information [12] or our own CORDER method [11] to find related entities in a textual corpus. Given a document, suppose there are a number of entities originally occurring in the document, however, entities which are related to these original entities may not necessarily also occur in the document, e.g., Thomas and Jack both work on X but one document only mentions Thomas works on X.

Therefore, we propose to enhance the content description of a document with entities, which are not in the document but are closely related to existing entities in a document. By doing so, we can enrich what is missing but in fact very relevant to the document, e.g., since Thomas and Jack both work on X, we add Jack to one document which only mentions Thomas works on X.

In terms of information retrieval (IR), vector space models are traditionally used to index a document with terms and words occurring in the document for term-based querying and document clustering. Thus, we propose to enhance the vector of a document with entities and terms (CORDER and statistical methods are applied to terms in the same manner as entities) which are not in the document but are closely related to existing entities and terms in the document. Since humans' term-based queries are often an approximation of the kind of information they are looking for, these enhanced vectors can often lead to improved quality of returned documents, e.g., one document, which has Thomas and Jack as original dimensions and project X as an enhanced dimension, will match the query "X", and the user may find this document useful since it provides detailed information about Thomas and Jack, two members of X.

In this paper, we propose a text mining method called LRD (latent relation discovery) which can automatically process a textual corpus for unearthing relationships among entities and terms, and use these relationships to enhance traditional vector space model for IR and document clustering.

We propose a relevance measure for a pair of co-occurring entities by taking into account both their co-occurrence and distance. The relevance measure measures the degree of relatedness and is referred to as relation strength between them. Given a target entity, we aim to find its related entities and rank them by their relation strengths to the target entity.

LRD is based on our own CORDER algorithm [11]. LRD can be viewed as an unsupervised machine learning method, i.e., the method does not need either richly annotated corpora required by supervised learning methods or instances of relations as initial seeds for weakly supervised learning methods.

LRD identifies entities which are relevant to a given target entity based on its co-occurrence and distance with other entities in a textual corpus. Given a document, entities, which are not in the document but are relevant to entities originally in the document, are used to enhance the vector representation of the document. The enhanced vector space model has led to improved IR on these documents and document clustering over the traditional vector space model. Since richer contexts are encoded in enhanced vectors, a document A, which is judged as not relevant to a query Q or another document B in the traditional vector space, however can be judged as relevant to the query Q or document B in the enhanced vector space. We have evaluated LRD in terms of F measure, a combination of precision and recall, in IR and compared with five other standard methods, and LRD has significantly outperformed all of them and the original vector space model.

The rest of the paper is organized as follows. We present related work in Section 2. Our LRD method is presented in Section 3. The experimental results are reported in Section 4. Finally, we conclude the paper and discuss future work in Section 5.

2 Related Work

Co-occurrence based methods have been widely applied, for instance, in the identification of collocations and information retrieval. Such methods aim to correlate textual structures in order to unearth latent relationships. One of these approaches is *Latent Semantic Indexing* (LSI) [3], which automatically discovers latent relationships among documents and terms through *Singular Vector Decomposition* (SVD). LSI has been applied mainly in the information retrieval area, and also used to discover highly related terms [4]. Furthermore, LSI can reduce the dimensionality without undermining precision in information retrieval systems. However the method is time-consuming when applied to a large corpus [7].

Other related methods which can be applied in this context are *t* test, chi-squared, *z* score, and mutual information (MI) [12]. Criticisms of these methods are that probabilities are assumed to be approximately normally distributed in *t* test, Z score is only applicable when the variance is known, *t* test and χ^2 test do not work well with low frequency events, and mutual information does not deal properly with data sparseness. Unlike these methods, LRD can deal with data sparseness and scales well to a large corpus since LRD treats each document as an atomic unit and any change requires only unitary reprocessing (the details of LRD is presented in Section 3).

In the line of document clustering, Hotho and Stumme [6] have made use of Formal Concept Analysis using background knowledge by mapping words to some concept in Wordnet in order to improve the clustering process. Also, Hotho et al. [5] proposed a model called COSA (Concept Selection and Aggregation) which uses ontologies for restricting the set of document features through aggregations. Another approach is based on analogy, aiming to produce, through alignment, pairs of definitions that share the same headword term, and promotes replacements in pairs without major changes in the meaning [1]. In previous work, we [9] presented a model that extracts relevant terms from researchers' curricula vitae integrated with ontology aiming to promote support to clustering. The problem with this approach lies in its ontology dependency. Our LRD method analyzes co-occurrences through textual corpora in order to establish the relation strength among entities which in turn improves IR and document clustering tasks.

In essence, our proposed LRD method is similar to those co-occurrence based approaches which aim to enhance context representation. However, by combining relation strength, which establishes latent relationships between entities, with the vector space model, our LRD method has achieved better results.

3 Proposed Approach

3.1 Overview

Our LRD model maps entities and their relationships extracted from documents. Entities are named entities extracted from documents using a *Named Entity Recognition* (NER) tool called ESpotter [10] and terms in the document. We calculate the relation strength between every pair of entities[1] by taking into account the pair's co-occurrences in these documents. We represent each document as a vector of entities, and construct an entity-by-document matrix. Given a document and its vector, the most relevant entities to those originally in the vector are identified to expand the document vector. We use these expanded vectors for query based information retrieval and document clustering.

3.2 Entity Extraction

Named Entity[2] Recognition (NER) is a well studied area [2]. We have used ESpotter [10], a NER system based on standard NER techniques and adapted to various domains on the Web by taking into account domain knowledge. ESpotter recognizes Named Entities (NEs) of various types. Users can configure ESpotter to recognize new types of entities using new lexicon entries and patterns. Domain knowledge, taken from sources such as ontologies, is represented as lexicon entries (e.g., the project names in an organization).

3.3 Relation Strength

Given a target entity (*E1*) which occurs in various documents, there are a number of entities which co-occur with it in these documents. We propose a latent relation discovery algorithm which ranks co-occurring NEs based on relation strength. Thus, NEs which have strong relations with a target NE can be identified. Our approach takes into account three aspects as follows:

Co-occurrence: Two entities are considered to co-occur if they appear in the same text fragment, which can be a document or a text window. For simplicity, in this section, we use document as the unit to count entity co-occurrence. The effect of different granularities of text fragments will be discussed later in Section 4. Generally, if one entity is closely related to another entity, they tend to co-occur often. To normalize the relatedness between two entities, *E1* and *E2*, the relative frequency [8] of co-occurrence is defined as follows.

$$\hat{p}(E1,E2)=\frac{Num(E1,E2)}{N} \tag{1}$$

where $Num(E1,E2)$ is the number of co-occurring documents for *E1* and *E2*, and N is the total number of documents in a corpus.

Distance. Two NEs which are closely related tend to occur close to each other. If two NEs, *E1* and *E2*, occur only once in a document, the distance between *E1* and *E2* is the difference between the word offsets of *E1* and *E2*. When *E1* or *E2* occur multiple times in the document, given *E1* as the target, the mean distance between *E1* and *E2* in the *i*th document, $m_i(E1, E2)$ is defined as follows.

[1] Entities refer to both named entities recognized by ESpotter and terms in the document.

[2] In this paper, named entities are proper names consisting of words or collocations extracted from documents and labeled as a particular class, i.e., person or organization.

$$m_i(E1,E2)=\frac{\sum_{j=1}^{f_i(E1)}\min(E1_j,E2)}{f_i(E1)} \tag{2}$$

where *f*$f_i(E1)$ is the number of occurrences of *E1* in the *i*th document, $min(E1_j, E2)$ is the minimum distance between the *j*th occurrence of *E1*, $E1_j$, and *E2*. Generally, $m_i(E1, E2)$ is not equal to $m_i(E2, E1)$.

Relation strength: Given an entity, *E1*, the relation strength between two entities *E1* and *E2* takes into account their co-occurrence, mean distance, and frequency in co-occurred documents as defined in Equation 3. The greater the mean distance is, the smaller the relation strength. Generally, the relation strength between *E1* and *E2* is asymmetric depending on whether *E1* or *E2* is the target.

$$R(E1,E2)=\hat{p}(E1,E2)\times\sum_i\left(\frac{f(Freq_i(E1))\times f(Freq_i(E2))}{m_i(E1,E2)}\right), \tag{3}$$

where $f(Freq_i(E1)) = tfidf_i(E1)$, $f(Freq_i(E2)) = tfidf_i(E2)$, and $Freq_i(E1)$ and $Freq_i(E2)$ are the numbers of occurrences of *E1* and *E2* in the *i*th document, respectively. The term frequency and inverted document frequency measure *tfidf* is defined as $tfidf_i(j) = tf_i(j) * \log_2\left(N/df_j\right)$, where $tf_i(j)= f_i(j)/ max(f_i(k))$ is the frequency $f_i(j)$ of entity *j* in the *i*th document normalized by the maximum frequency of any entity in the *i*th document, *N* is the number of documents in the corpus, and df_j is the number of documents that contain the entity *j*.

3.4 Vector Expansion

In vector composition, we intend to enhance the vector space by co-occurred entities. After entity extraction, we calculate the relation strength between every pair of entities using Equation 3. For example, in Table 1, an entity-by-document is constructed from 3 documents (D1, D2, and D3) and 7 entities.

Table 1. Example of a document-to-entity matrix and frequencies in the matrix are normalized using *tfidf* in the matrix on the right (D1-N, D2-N and D3-N)

Entities	D1	D2	D3	D1-N	D2-N	D3-N
E1	4	2	0	0.5850	0.5850	0
E2	2	0	3	0.2925	0	0.5850
E3	3	2	0	0.4387	0.5850	0
E4	1	1	0	0.1462	0.2925	0
E5	0	2	0	0	1.5850	0
E6	0	0	2	0	0	1.0566
E7	1	0	2	0.1462	0	0.3900

We create a table consisting of pairs of related entities. Each row in the table consists of a document ID, a source and a target entity co-occurring in the document with their frequencies, the frequency of their co-occurrences, and their intra-document distance. As an example, Table 2 shows pairs of related entities in document 1, their frequencies and the distance between them calculated using Equation 2 and the intra-document relation strength calculated using the second part of Equation 3 (i.e., $(f(Freq_i(E1)) \times f(Freq_i(E2)))/m_i(E1,E2)$). Similarly, we can get the table for co-occurred entities in document 2 and 3.

Table 2. Example of co-occurred entities in documents

Doc	Source Entity (SE)	SE tf	Target Entity (TE)	TE tf	Distance	Intra-doc relation strength
1	E1	4	E2	2	2.0000	0.4387
1	E1	4	E3	3	2.0731	0.4938
1	E1	4	E4	1	2.3634	0.3094
1	E1	4	E7	1	2.8540	0.2562
1	E2	2	E3	3	2.3412	0.3123
1	E2	2	E4	1	2.6887	0.1632
1	E2	2	E7	1	3.0805	0.1424
1	E3	3	E4	1	2.2642	0.2584
1	E3	3	E7	1	2.5654	0.2280
1	E4	1	E7	1	3.8074	0.0768

Table 3. Example of relation strengths between co-occurred entities

SE/TE	E1	E2	E3	E4	E5	E6	E7
E1	N/A	0.1462	0.7192	0.4788	0.2590	0	0.0854
E2	0.1136	N/A	0.1041	0.0544	0	0.2609	0.3290
E3	0.6528	0.0364	N/A	0.3898	0.3155	0	0.0760
E4	0.3675	0.1754	0.2568	N/A	0.1847	0	0.0256
E5	0.3687	0	0.4876	0.2512	N/A	0	0
E6	0	0.1856	0	0	0	N/A	0.1423
E7	0.1569	0.4587	0.1233	0.0489	0	0.1423	N/A

Given a pair of entities, we can calculate their relation strength shown in Table 3. For example, the relation strength between target entity (TE) E1 and source entity (SE) *E*3,is computed using Equation 3 as *R*(*E1*,*E3*)=2/3*(0.4938+0.5850)=0.7192. Relation strength is used to recompose the vector space. For example, in Table 1, the vector of document 1 does not contain *E*5 and *E*6. However, judging by relation strength, the most relevant entity to *E3* not in the vector of document 1 is *E5* and to *E2* not in the vector of document 1 is *E6*, with relation strength 0.3155 and 0.2609, respectively. Since *E3* and *E2* are dimensions in the vector of document 1, *E5* and *E6* are considered to be added to the vector of document 1. Generally, for each entity originally in a document vector as the target, we add each of the top *n* entities related to the target and not in the document vector (ranked by their relation strengths), E_{new}, to the document vector. The weight of E_{new}, $w(E_{new})$, is defined as follows.

$$w(E_{new}) = \sum_{i=1}^{num(E_{new},D)} R(E_{new}, E_i) \times w(E_i) \tag{4}$$

where $R(E_{new},E_i)$ is the relation strength between E_{new} and E_i, which is originally in the vector of document *D*, $w(E_i)$ is the weight of E_i in document *D*, and $num(E_{new},D)$ is the total number of entities originally in the document vector having E_{new} in the top *n* most relevant entities in terms of relation strength.

In Table 4, we set *n*=1. We add *E5* (No. 1 entity not in document vector (N1NDV) of SEs: E1, E3, E4) and *E6* (N1NDV of SE: E2) to document one, *E2* (N1NDV of SE: E1, E4) and *E7* (N1NDV of SE: E3) to document two, and *E3* (N1NDV of SE:

E1, E2, E7) to document three. For example, the weight of *E5* in *D1* is: 0.5850*0.2590+0.4387*0.3155+0.1462*0.1847 = 0.3169.

Table 4. Example of an entity-by-document matrix enhanced by related entities

Entities	D1	D2	D3
E1	0.5850	0.5850	0.1462
E2	0.2925	0.1368	0.5850
E3	0.4387	0.5850	0.2143
E4	0.1462	0.2925	0
E5	0.3169	1.5850	0
E6	0.0763	0	1.0566
E7	0.1462	0.0445	0.3900

3.5 Query-based Information Retrieval and Document Clustering

We calculate a cosine coefficient between the expanded vector of each document and the vector of a term-based query and use the cosine coefficients to rank documents with respect to the query. We setup a threshold on the cosine coefficient to trade precision against recall in retrieving these documents.

We apply a clustering algorithm to generate patterns for in-depth analysis of how documents and entities are inter-connected. Unlike the traditional *k*-means algorithm which is based on the parameter *k* (number of clusters), we use an approach based on a radius parameter (*r*) to control the cluster formation.

The algorithm starts with selecting a vector (either randomly or the one most separated from the others) and forms the first cluster. By repeating the process, the next vector is selected and compared with the first cluster by applying the cosine measure defined as follows.

$$\cos\theta = \frac{\sum_{i=1}^{n}\left(t_i * q_i\right)}{\sqrt{\sum_{k=1}^{n}\left(t_k\right)^2} * \sqrt{\sum_{j=1}^{n}\left(q_j\right)^2}}, \tag{5}$$

where t_i and t_k are the normalized frequencies of the *i*th and *k*th entities in the vector *t*, and q_i and q_j are the normalized frequencies of the *i*th and *j*th entities in the vector *q*.

If the similarity between a vector and a cluster centroid subtracted from 1 is greater than the *r* parameter, the vector forms a new cluster. Otherwise, it is assigned to the cluster and we recalculate the centroids of the clusters. Experiments using a range of values of *r* from 0.2 to 0.7 were carried out and the best results were achieved with *r* = 0.3. During the next iterations, if the vector moves from one cluster to another, the centroid updating is carried out in both the new cluster to which the vector has been added and the old cluster from which the vector has been removed.

The clustering process stops when it reaches convergence, which is determined by the total average difference between the current and previous epoch. Our experiments on different datasets have shown that epochs between 2 and 10 are required. After the clustering, we get clusters consisting of vectors and cluster centroid average.

4 Empirical Evaluation

We have evaluated our proposed relation strength model (LRD) in term of *F* measure, which combines precision and recall, by comparing with five standard statistical methods (LSI and four other methods based on a relation strength model for vector expansion) in information retrieval. In order to automate the evaluation process, the Glasgow Information Retrieval benchmark dataset called CISI[3] containing 1,460 documents and 112 queries has been used. Terms in the documents and entities extracted from documents using ESpotter are used during the correlation and vector expansion processes.

4.1 Relation Strength Models

We have compared LRD with four standard statistical methods in relation strength calculation. These relation strengths are used for vector expansion. The four methods, i.e., mutual information (MI), improved MI, phi-squared, and Z score are presented as follows.

Mutual Information (MI) compares the probability of two entities, *x* and *y* or any other linguistic unit, such as named entities, appearing together against the probability that they appear independently. The higher the MI value, the greater the degree of relevance between two entities. MI is defined as follows.

$$I(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)}, \tag{6}$$

where $P(x,y)$ is the probability that *x* and *y* co-occur in a text fragment (which can be a document, or a text window), and $P(x)$ and $P(y)$ are the probabilities that *x* and *y* occur individually.

We have also applied Vechtomova et al.'s improved MI (VMI) method [13]. The standard MI is symmetrical, i.e. $I(x,y) = I(y,x)$, as joint probabilities are symmetrical, $P(x,y) = P(y,x)$. Unlike traditional MI, VMI is asymmetrical. An average window size calculated from all windows around term *x* is used to estimate the probability of occurrence of *y* in the windows around *x*. VMI is defined as follows.

$$I_v(x,y) = \log_2 \frac{P_v(x,y)}{P(x)P(y)} = \log_2 \frac{\frac{f(x,y)}{Nv_x}}{\frac{f(x)f(y)}{N^2}}, \tag{7}$$

where $f(x,y)$ is the joint frequency of *x* and *y* in the corpus, $f(x)$ and $f(y)$ are frequencies of independent occurrence of *x* and *y* in the corpus, v_x is the average window size around *x* in the corpus, and *N* is the corpus size.

Phi-squared (ϕ^2) makes use of a contingency table as follows:

[3] http://www.dcs.gla.ac.uk/idom/ir_resources/test_collections/cisi/

	w_2	$\overline{w}_2$
w_1	a	b
$\overline{w}_1$	c	d

where cell *a* indicates the number of times entities w_1 and w_2 co-occur in a window. Cell *b* indicates the number of times w_1 occurs but w_2 does not. Cell *c* indicates the number of times w_2 occurs but w_1 does not. Finally, cell *d* indicates the number of times neither entity occurs, that is, $d = \frac{N}{S} - a - b - c,$ where *N* is the size of the corpus and *S* the size of the text window. ϕ^2 measure between w_1 and w_2 is defined as:

$$\phi^2 = \frac{(ad - bc)^2}{(a+b)(a+c)(b+d)(c+d)}, \tag{8}$$

where $0 \le \phi^2 \le 1$. Unlike MI which typically favors entities with low frequency, ϕ^2 can be used as an alternative, since it tends to favor high frequency ones.

Z score has been used by Vechtomova et al. [13] for query expansion. Z score is defined as follows.

$$Z(x,y) = \frac{O-E}{\sqrt{E}} = \frac{f(x,y) - \frac{v_x f(x) f(y)}{N}}{\sqrt{\frac{v_x f(x) f(y)}{N}}}, \tag{9}$$

where *f*(*x*,*y*) is the joint frequency of *x* and *y* in the corpus, *f*(*x*) and *f*(*y*) are frequencies of independent occurrence of *x* and *y* in the corpus, v_x is the average window size around *x* in the corpus and *N* is the corpus size.

4.2 Experimental Results of Vector Space Model for Information Retrieval

By expanding document vectors and applying different relation strength methods we intend to establish a way to automatically evaluate our proposed method. In this sense we have compared LRD, Phi-squared, MI, VMI and *Z* score in order to find out entities and terms closely related to the original entities and terms in the vector. For each method, the original vector is expanded using the method by taking into account different text windows and expansion factors.

Given a document vector, it is expanded using the method presented in Section 3 with different text windows and *n* factors (the *n* most related entities to each original entity in a document vector, which do no occur in an original vector as dimensions, are added to the vector). We have used the text window of 20, 50, 100 and 200, and the whole document (i.e., two entities are considered as co-occurring as long as they occur in a same document). For the *n* factor, values of 1, 5, 10, 20, 30, 40 and 50 most relevant entities are used to expand the vector space. The same vector expansion process using each of the relation strength methods is applied to the corpus using different text window and *n* factor.

We have applied each relation strength method to the 1,460 documents in the CISI dataset. The constructed vector space by each method using different text window and *n* factor is used for information retrieval. We randomly selected 20 queries from the 112 queries in CISI. Given a query, we calculate a cosine coefficient between the vector of each document and the vector of the query to rank these documents against the query.

Given a query, we set a threshold on the cosine coefficient and only documents having cosine coefficient with the query above the threshold are taken into account in our precision and recall calculation. Given a query, the precision (*P*) of our answer is the number of relevant documents returned divided by the total number of returned documents, and recall (*R*) is the number of relevant documents returned divided by the total number of relevant documents as the gold standard in CISI. We define the *F* measure as $F = \frac{2 \times P \times R}{P + R}$. In our experiments, we set the cosine similarity threshold as 0.54 which maximizes F measure on most queries. Given a relation strength method with different window size and *n* factor, we average the *F* measure for each of the 20 answers to get the total F measure and the results are shown in Table 5.

Table 5. The average F measure for LSI and 5 methods with seven expansion factor (*n*) values and five text window settings, the highest *F* measure for each window setting is in bold and shaded cell.

F-measure (%)		*n*=1	5	10	20	30	40	50
No win-dow	LRD	16.5	18.4	18.1	18.7	**19.3**	18.9	17.8
	Phi-squared	12.9	11.5	10.8	10.8	10.9	11.3	12.1
	MI	9.3	6.6	6.3	5.9	5.8	5.7	5.9
	VMI	10.5	6.6	6.1	5.8	5.8	5.9	5.9
	Z Score	14.7	12.8	13.0	11.8	11.0	10.0	10.0
	LSI	15.3	16.1	15.7	16.1	16.6	15.2	14.9
Size =20	LRD	16.9	18.3	18.7	18.9	**19.0**	18.7	18.6
	Phi-squared	15.3	12.9	13.1	10.7	12.0	11.7	11.1
	MI	10.3	6.6	6.2	5.9	5.8	6.0	5.9
	VMI	10.6	6.7	6.2	5.9	5.8	6.0	5.9
	Z Score	14.4	14.3	14.4	14.9	14.5	15.6	15.3
Size =50	LRD	16.8	18.4	18.2	18.8	**18.9**	18.9	18.7
	Phi-squared	13.3	10.6	9.7	9.6	9.3	10.8	9.9
	MI	10.3	6.7	6.2	5.9	5.8	5.9	5.9
	VMI	10.5	6.7	6.1	5.8	5.8	6.0	5.9
	Z Score	14.5	14.5	14.0	15.6	15.2	17.1	16.1
Size =100	LRD	16.5	18.4	18.1	18.6	**19.2**	18.8	17.7
	Phi-squared	13.8	10.4	7.9	8.1	7.1	6.6	6.3
	MI	9.4	6.7	6.3	5.9	5.8	5.7	5.9
	VMI	9.6	6.6	6.1	5.8	5.8	5.9	5.9
	Z Score	14.5	14.6	13.9	15.2	15.4	17.4	16.8
Size =200	LRD	16.5	18.3	18.1	18.7	**19.2**	18.8	17.8
	Phi-squared	14.2	13.0	12.0	8.7	7.7	7.6	7.2
	MI	9.3	6.6	6.2	5.9	5.8	5.7	5.9
	VMI	10.5	6.6	6.1	5.8	5.8	5.9	5.9
	Z Score	14.5	14.6	13.9	15.2	15.4	16.7	16.6

The average *F* measure for the original vector space model, i.e., without vector expansion and text window, is 9.2% and provides a baseline for our comparison. As shown in Table 5, LSI, which is only evaluated based on the use of whole documents rather than text windows, is also included. For LSI and the other five methods which work on different window settings, LRD consistently performs the best. The highest *F* measure is 19.3% using LRD with no window and *n*=30. The second best performing method is LSI with highest *F* measure 16.6% with *n*=30. The third best performing method is Z score with highest F measure 17.4% with window size 100 and *n*=40. MI and VMI have similar performance.

In terms of the influence of *n* factor on these methods, when *n* factor increases, the *F* measure of LSI keeps roughly the same. For a given window setting, when *n* factor increases, the *F* measures of LRD and *Z* score increase (for the *F* measure of LRD, the biggest increases is from *n*=1 to *n*=5 and the increase from *n*=5 becomes very small even some small decreases), the *F* measure of phi-squared, MI and VMI decrease. Since the baseline is 9.2%, we can see that vector expansion with LRD, LSI, phi-squared and *Z* score have a positive effect on information retrieval and with the other methods have a negative effect on information retrieval. LRD has achieved the best performance for all window sizes when *n* = 30, and larger *n* values will not bring benefit to the performance and on the contrary bring computational cost.

In terms of the effect of window size on average *F* measure, LRD and Z score are not very sensitive to the varying window sizes, and consistently perform better than the other methods. Phi-squared, MI, and VMI achieve better *F* measures with smaller window sizes than those with larger window sizes.

5 Conclusions and Future Work

We present a co-occurrence based approach, namely LRD (Latent relation Discovery), which associates entities using relation strengths among them. We propose to use inter-entity relation strength to enhance the traditional vector representation of documents in order to provide additional meaning and improve query based information retrieval on these documents. Our initial experiments using the CISI dataset have shown that LRD can dramatically improve the *F* measure of information retrieval over the traditional vector space model, and significantly outperformed five standard methods for vector expansion. Our experiments on the CISI dataset show that LRD's running time increases linearly with the size of documents and the number of documents it examines. It can incrementally evaluate existing relations and establish new relations by taking into account new documents. Thus, LRD can scale well to a large corpus.

Our future work is five-fold. First, we are working on refining the LRD model in order to improve the metrics used to establish the relation strengths between entities and improve the clustering method. Second, we propose clustering documents based on the enhanced vector space models produced by our LRD method, however, the evaluation and interpretation of these clusters is neither an easy nor an intuitive task. We are carrying out work on using various techniques to evaluate these clusters. Our work underway is the visualization of these clusters in order to show complex patterns of inter-connected entities in clustered documents for easy comparison between these clusters produced by different vector space models. Third, we are evaluating our enhanced vector space models for information retrieval and clustering on large scale TREC collections such as TIPSTER. Fourth, dimensionality reduction is another direction and needs to be studied in order to improve the performance of our method. Finally, entities and their relations constitute a social network of communities of practice. We are working on using the social network to help analyze and understand the behavior of these communities.

References

1. Castillo, G., Sierra, G., and McNaught, J. An improved algorithm for semantic clustering. In Proc. of 1st International Symposium on Information and Communication Technologies, ACM International Conference Proceeding Series, Dublin, Ireland, 2003, 304-309.
2. Cunningham, H. GATE: a General Architecture for Text Engineering. Computers and the Humanities, vol. 36, issue 2, 2002, 223-254.
3. Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. Indexing by latent semantic analysis. Journal of the American Society of Information Science, vol. 41, issue 6, 1990, 391-407.
4. Ding, C. H. Q. A probabilistic model for dimensionality reduction in information retrieval and filtering. In Proc. of the 1st SIAM Computational Information Retrieval Workshop, Raleigh, NC, 2000.
5. Hotho, A., Maedche, A., and Staab, S. Text clustering based on good aggregations. In Proc. of the 2001 IEEE International Conference on Data Mining, IEEE Computer Society, San Jose, CA, 2001, 607-608.
6. Hotho, A., and Stumme, G. Conceptual clustering of text clusters. In Proc. of the Fachgruppentreffen Maschinelles Lernen (FGML), Hannover, Germany, 2002, 37-45.
7. Ikehara, S., Murakami, J., Kimoto, Y., and Araki, T. Vector space model based on semantic attributes of words. In Proc. of the Pacific Association for Computational Linguistics (PACLING), Kitakyushu, Japan, 2001.
8. Resnik, P. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. Journal of Artificial Intelligence Research: An International Electronic and Print Journal, vol. 11, 1999, 95-130.
9. Gonçalves, A., Uren, V., Kern, V. and Pacheco, R. Mining Knowledge from Textual Databases: An Approach using Ontology-based Context Vectors. In Proc. of the International Conference on Artificial Intelligence and Applications (AIA 2005), Innsbruck, Austria, 2005, 66-71.
10. Zhu, J., Uren, V., and Motta, E. ESpotter: Adaptive Named Entity Recognition for Web Browsing. In Proc. of the 3rd Conference on Professional Knowledge Management (WM 2005), pp.518-529, Springer LNAI, 2005.
11. Zhu, J., Gonçalves, A., Uren, V., Motta, E., and Pacheco, R. (2005). Mining Web Data for Competency Management. In Proc. of Web Intelligence (WI 2005), France, 2005, pp. 94-100, IEEE Computer Society.
12. Church, K., and Hanks, P. Word association norms, mutual information, and lexicography. Computational Linguistics, vol. 16, issue 1, 1990, 22-29.
13. Vechtomova, O., Robertson, S., and Jones, S. Query expansion with long-span collocates. Information Retrieval. vol. 6, issue 2, 2003, 251-273.