Random Walks on Adjacency Graphs for Mining Lexical Relations from Big Text Data

Shan Jiang
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL, 61801 USA
sjiang18@illinois.edu

ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL, 61801 USA
czhai@cs.uiuc.edu

***Abstract*—Lexical relations, or semantic relations of words, are useful knowledge fundamental to all applications since they help to capture inherent semantic variations of vocabulary in human languages. Discovering such knowledge in a robust way from arbitrary text data is a significant challenge in big text data mining. In this paper, we propose a novel general probabilistic approach based on random walks on word adjacency graphs to systematically mine two fundamental and complementary lexical relations, i.e., paradigmatic and syntagmatic relations between words from arbitrary text data. We show that representing text data as an adjacency graph opens up many opportunities to define interesting random walks for mining lexical relation patterns, and propose specific random walk algorithms for mining paradigmatic and syntagmatic relations. Evaluation results on multiple corpora show that the proposed random walk-based algorithms can discover meaningful paradigmatic and syntagmatic relations of words from text data.**

I. Introduction

The dramatic growth of text data creates great opportunities for applying computational methods to mine "big text data" to discover all kinds of useful knowledge and support many data analytics applications. Unfortunately, text data are unstructured, and effective discovery of knowledge from text data requires the computer to understand natural languages, which is known to be an extremely difficult task. In this paper, we study how to mine two fundamental and complementary types of interesting semantic relations between words from arbitrary text data in a scalable way. The first is the relation between two words that tend to *occur in similar context*; such a relation connects distributionally similar words. The second is the relation between two words that tend to *co-occur* with each other together; such a relation connects statistically associated words. In semiotics, the first type of relation is called *paradigmatic relation*, and the second *syntagmatic relation*. Paradigmatic relation tells us how words are associated with one another as playing similar roles in terms of functional rule, thus often capturing synonym-like relations, while syntagmatic relation reveals how words can be combined with each other to complete the functional synthesis, thus often capturing topically associated words.

To illustrate these two relationships, consider two synonyms such as "car" and "vehicle", which is a good example of words that have a paradigmatic relation because they tend to occur in the same context. If we substitute one for the other in a sentence, we would still have a meaningful sentence, whereas two semantically associated words such as "car" and "drive" would have a syntagmatic relation because they tend to co-occur in the same sentence (note that we generally would not obtain a meaningful sentence by substituting "car" for "drive" or "drive" for "car").

Both paradigmatic and syntagmatic relations are very useful knowledge fundamental to various applications involving text processing, including, e.g., search engines, recommender systems, text classification, text summarization, and text analytics. For example, such relations can be directly useful in search engine applications to enrich the representation of a query or suggest related queries, and for capturing inexact matching of text for classification or clustering.

In this paper, we study how to mine large text data in an unsupervised way to discover paradigmatic and syntagmatic relations efficiently and effectively. We propose a novel general probabilistic approach based on random walks on word adjacency graphs to systematically mine these two fundamental and complementary lexical relations between words from arbitrary text data. Such a new approach has several important advantages: 1) It is completely general and unsupervised, thus it requires no/little human effort and can be applied to mine arbitrary text data in any natural language. 2) It is a principled probabilistic approach with a solid foundation based on random processes, thus the scores to quantify lexical relations are meaningful, and it is easy to adapt the approach to capture different types of semantic relations between words by simply changing the way a random walk is defined. 3) It is very efficient for discovering paradigmatic and syntagmatic relations, and thus can potentially scale up to mine very big text data. Updating the graph is very efficient as we only need to update the co-occurrence statistics, making such a method scale up well to handle the "never-ending growth" of big text data in the real world in an online manner. Evaluation results show that the proposed algorithms are effective for mining these two kinds of relational patterns and can discover quite meaningful lexical knowledge from large text data without any human effort. Due to the generality and scalability, the proposed algorithms can be potentially applied to large amounts of arbitrary text data in different natural languages for discovery of useful lexical knowledge.

II. Related Work

In the natural language processing community, many approaches have been proposed for acquiring lexical relations of words (see, e.g., the early work[7]). Many of these approaches rely on special linguistic rules, thus the lexical relations that can be acquired are quite limited, and these methods can only work for text in a language with manually created rules available. Statistical language models and neural network word embedding (e.g., [3], [1], [11]) have shown very promising results on word clustering and semantic representation of words, but these approaches are generally computationally very expensive. Earlier work has also made much effort to mine paradigmatic relation from text corpus [10], [14], [4]. In [4], Bullinaria and Levy present a method for semantic representation extraction from the word-word co-occurrence. They show that SVD can help to improve the performance significantly. Unfortunately, SVD is both space and time consuming, making it unable to handle very big text data.

The proposed sequence-based adjacency graph is closely related to the opinosis graph proposed in [6], where such a graph is used for summarization of opinions. However, our definition of the adjacency graph is more general as it allows edges to connect words that are not immediately adjacent to each other. Besides, we use such a graph for mining lexical knowledge which has not been attempted before.

Random walks on graph have been applied to many data mining problems, such as recommendation [2], [9], community detection [12], [15] and graph-based rating [13], [8]. We add to this pool of applications of random walks yet another new application to discover interesting lexical relations in text data.

III. Adjacency Graph

Since the basis of the proposed text mining approach is the representation of text data as a word adjacency graph, we first introduce the representation of adjacency graph induced from text data in this section.

Word is the basic element in a text data set and can be regarded as being drawn from a vocabulary set. Note that the proposed algorithm would treat a "word" as a basic unit for analysis, thus one can also use whatever units (e.g., phrases, entities) as "words" when constructing the adjacency graph.

Definition 1 *A vocabulary set V is denoted as $V = \{v_1, v_2, ..., v_N\}$, where $v_i (1 \leqslant i \leqslant N)$ is a unique word in the data set.*

Text data can be considered as a special case of the family of sequence data, where a sequence is a series of ordered elements. In text data, we can treat each sentence, paragraph or even document as an individual sequence, and construct the sequence-based adjacency graph from the sequences.

Definition 2 *A sequence s_i is represented as $\langle v_{i1}, v_{i2}, ..., v_{il}\rangle$, where $v_{ij} \in V$ $(1 \leqslant j \leqslant l)$. The length of s_i is defined as l, namely the number of elements in s_i. And a sequence set S consists of n sequences, i.e., $S = \{s_1, s_2, ..., s_n\}$.*

Definition 3 Subsequence *Given two sequences s_p and s_q $(s_p = \langle v_{p,1},\ v_{p,2},\ ...,\ v_{p,l_p}\rangle,\ s_q = \langle v_{q,1},\ v_{q,2},\ ...,\ v_{q,l_q}\rangle)$, then $s_p \subseteq s_q$ holds (i.e., s_p is a subsequence of s_q) if $l_p \leqslant l_q$ and there exists an integer r $(1 \leqslant r \leqslant l_q - l_p + 1)$ such that $v_{p,1} = v_{q,r},\ v_{p,2} = v_{q,r+1},\ ...\ v_{p,l_p} = v_{q,r+l_p-1}$.*

We can derive a series of adjacency graphs from a sequence set by taking variations of adjacency measure. Nodes in the adjacency graph are elements in the sequence set (words in the text data). Both immediate adjacency and non-immediate adjacency co-occurrence can be used to construct the edges. When using immediate adjacency, edges are added to elements which occur next to each other. For non-immediate adjacency, co-occurrence is not restricted to nearest neighbors, but can have a static gap with a fixed distance between elements. The weight on an edge is the total number of co-occurrences of the two nodes connected by the edge.

An adjacency graph G_k derived from sequence set S is constructed in the following way:

Definition 4 *The node set of G_k is denoted as V_k, which is the same as the vocabulary set of S. If there exists a sequence $s' = \langle v_i, v_{r_1}, v_{r_2}, ..., v_{r_{k-1}}, v_j\rangle$ and a sequence $s \in S$ such that $s' \subseteq s$, an edge (v_i, v_j) will be added from v_i to v_j and its weight $w[(v_i, v_j)] = |\{s'|s' = \langle v_i, v_{r_1}, v_{r_2}, ..., v_{r_{k-1}}, v_j\rangle \wedge s \in S \wedge s' \subseteq s\}|$. The edge set of G_k is denoted as E_k.*

When $k = 1$, s' is actually $\langle v_i, v_j\rangle$ and an edge will be added from v_i to v_j if and only if v_i is followed immediately by v_j.

IV. Random Walks on Adjacency Graph

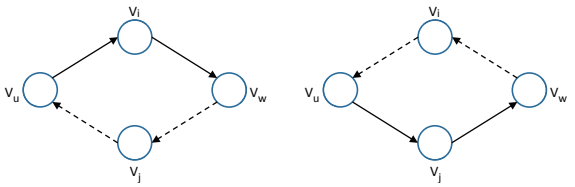
Given an adjacency graph G, two basic types of random walks can be defined, namely forward walking and backward walking. Assume we have a series of edges (v_{r_1}, v_{r_2}), (v_{r_2}, v_{r_3}), ... ,$(v_{r_l}, v_{r_{l+1}})$ in G. A forward walking $v_{r_1} \rightarrow v_{r_2} ... \rightarrow v_{r_{l+1}}$ is to visit v_{r_1}, v_{r_2},..., $v_{r_{l+1}}$ sequentially. And a backward walking $v_{r_{l+1}}$ -↠ $v_{r_l} ...$ -↠ v_{r_1} is to visit $v_{r_{l+1}}, v_{r_l}$,..., v_{r_1} by taking the inverse direction of edges between them. The weights on edges can be normalized in various ways to allow for interpreting the graph as a transition matrix with the nodes of the graph as states. Such a probabilistic interpretation enables us to compute probabilities of different random walks on the graph, which we can then use to mine interesting paths, relations between words (as we explore in this paper), or even interesting subgraphs.

Definition 5 *An l-step forward walking $v_i \overset{l}{\rightarrow} v_j = \{v_i \rightarrow v_{r_1} \rightarrow v_{r_2} ... \rightarrow v_{r_{l-1}} \rightarrow v_j | (v_i, v_{r_1}), ..., (v_{r_{l-1}}, v_j) \in E\}$ and an l-step backward walking $v_i \overset{l}{\text{-↠}} v_j = \{v_i$ -↠ v_{r_1} -↠ $v_{r_2} ...$ -↠ $v_{r_{l-1}}$ -↠ $v_j | (v_{r_1}, v_i), ..., (v_j, v_{r_{l-1}}) \in E\}$.*

The probability of an l-step forward walking from v_i to v_j in G is denoted as $P_G(v_i \overset{l}{\rightarrow} v_j)$ and the probability of an l-step backward walking is denoted as $P_G(v_i \overset{l}{\text{-↠}} v_j)$.

To compute $P_G(v_i \overset{l}{\rightarrow} v_j)$ and $P_G(v_i \overset{l}{\text{-↠}} v_j)$, we first compute two diagonal matrix D_F and D_B. Given that the adjacent matrix of G is A and $A(i,j) = w[(v_i, v_j)]$, where $w[(v_i, v_j)]$ is the weight of the edge from v_i to v_j in G. Then

$$D_F(i,i) = \frac{1}{\sum_{j=1}^{|V|} A(i,j)}, \qquad D_B(i,i) = \frac{1}{\sum_{j=1}^{|V|} A(j,i)}.$$



(a) (b)

Fig. 1. Illustration of circle trip for paradigmatic relation mining. (a) Clockwise circle trip; (b) Anti-clockwise circle trip.

Both $D_F(i,j)$ and $D_B(i,j)$ will be 0 when $i \neq j$. Then forward and backward walking transition matrix T_F and T_B can be defined as:

$$T_F = D_F A, \qquad T_B = D_B A^T.$$

It is obvious that $P_g(v_i \overset{1}{\rightarrow} v_j) = T_F(i,j)$ and $P_g(v_i \overset{1}{\dashrightarrow} v_j) = T_B(i,j)$. Based on 1-step walking, $P_g(v_i \overset{l}{\rightarrow} v_j)$ and $P_g(v_i \overset{l}{\dashrightarrow} v_j)$ for multi-step walking can be derived iteratively:

$$P_G(v_i \overset{l}{\rightarrow} v_j) = \sum_{v_r \in V} P_G(v_i \overset{l-1}{\rightarrow} v_r) \cdot P_G(v_r \overset{1}{\rightarrow} v_j) \quad = T_F^l(i,j) \quad (1)$$

$$P_G(v_i \overset{l}{\dashrightarrow} v_j) = \sum_{v_r \in V} P_G(v_i \overset{l-1}{\dashrightarrow} v_r) \cdot P_G(v_r \overset{1}{\dashrightarrow} v_j) \quad = T_B^l(i,j) \quad (2)$$

V. Mining Paradigmatic and Syntagmatic Relations

Paradigmatic and syntagmatic are the two most basic and complementary lexical relations in natural language text. Below we discuss how we can discover both kinds of relations elegantly using "round trip" random walks.

A. Mining Paradigmatic Relation

Paragmatic relations can be discovered by taking l-step circle trips, in both clockwise and anti-clockwise directions. In the first step, we need to find all common neighbors that are reachable from v_i and v_j by an l-step forward or backward walking in the adjacency graph. As the example shown in Figure 1, v_u and v_w are common neighbors of v_i and v_j from left side and right side respectively, which satisfy the following conditions: $P_G(v_u \overset{l}{\rightarrow} v_i) > 0$, $P_G(v_u \overset{l}{\rightarrow} v_j) > 0$, $P_G(v_i \overset{l}{\rightarrow} v_w) > 0$ and $P_G(v_j \overset{l}{\rightarrow} v_w) > 0$. Then the clockwise circle trip between them is to walk in the order of $v_i \overset{l}{\rightarrow} v_w \overset{l}{\dashrightarrow} v_j \overset{l}{\dashrightarrow} v_u \overset{l}{\rightarrow} v_i$ (see Figure 1(a)) and the anti-clockwise trip is $v_i \overset{l}{\dashrightarrow} v_u \overset{l}{\rightarrow} v_j \overset{l}{\rightarrow} v_w \overset{l}{\dashrightarrow} v_i$ (see Figure 1(b)). To distinguish v_i and v_j from v_u and v_w, we say that v_i and v_j are vertical ends and v_u and v_w are horizontal ends.

We denote the probability of taking l-step circle trip with vertical ends of v_i and v_j in both clockwise and anti-clockwise directions in adjacency graph G as $P_G(v_i \overset{l}{\circlearrowleft} v_j)$. If we denote $\{v_u | P_G(v_u \overset{l}{\rightarrow} v_i) > 0 \wedge P_G(v_u \overset{l}{\rightarrow} v_j) > 0\}$ as $\mathbb{U}$ and $\{v_w | P_G(v_i \overset{l}{\rightarrow} v_w) > 0 \wedge P_G(v_j \overset{l}{\rightarrow} v_w) > 0\}$ as $\mathbb{W}$, then

$$\begin{aligned} P_G(v_i \overset{l}{\circlearrowleft} v_j) = & \sum_{v_u \in \mathbb{U}} \sum_{v_w \in \mathbb{W}} P_G(v_i \overset{l}{\rightarrow} v_w) \cdot P_G(v_w \overset{l}{\dashrightarrow} v_j) \\ & \cdot P_G(v_j \overset{l}{\dashrightarrow} v_u) \cdot P_G(v_u \overset{l}{\rightarrow} v_i) \cdot \sum_{v_u \in \mathbb{U}} \sum_{v_w \in \mathbb{W}} P_G(v_i \overset{l}{\dashrightarrow} v_u) \\ & \cdot P_G(v_u \overset{l}{\rightarrow} v_j) \cdot P_G(v_j \overset{l}{\rightarrow} v_w) \cdot P_G(v_w \overset{l}{\dashrightarrow} v_i) \end{aligned} \quad (3)$$

Strong paradigmatic relation usually makes it easy to accomplish circle trip by random walk. However, it may cause bias towards frequent words if we only use $P_G(v_i \overset{l}{\circlearrowleft} v_j)$ to extract paradigmatic relation. When $|\mathbb{U}|$ and $|\mathbb{W}|$ are large enough, no matter whether v_i and v_j are substitutable in the contexts or not, the value of $P_G(v_i \overset{l}{\circlearrowleft} v_j)$ will be high. To tackle this problem, we normalize $P_G(v_i \overset{l}{\circlearrowleft} v_j)$ by the number of all possible circle trips in a unique direction, which is $|\mathbb{U}| \cdot |\mathbb{W}|$. The normalized $P_G(v_i \overset{l}{\circlearrowleft} v_j)$ reflects how v_i and v_j are correlated with their common context on average and a high value implies that they are likely to be substitutable.

Given that the random walker can choose different l to complete the circle trip, we will have:

$$\sum_{l=0}^{s} P_G(v_i \overset{l}{\circlearrowleft} v_j) \cdot \frac{1}{|\mathbb{U}| \cdot |\mathbb{W}|} \cdot \alpha_l, \quad (4)$$

where s is the max step allowed in the circle trip. α_l is the prior probability for the random walker to chose l. If short-distance circle trips are more reliable, small l will get higher α_l. If different multi-step circle trips are equally favored regardless of the path length, α_l is set equivalently for different l. Due to the probabilistic interpretation of α_l, it should be non-negative and $\sum_{l=0}^{s} \alpha_l = 1$.

Combining different adjacency graphs induced from the same data together, we finally use $Pr(v_i, v_j)$ to measure the paradigmatic relation, which is defined as:

$$Pr(v_i, v_j) = \sum_{k=1}^{K} \beta_k \sum_{l=0}^{s} P_{G_k}(v_i \overset{l}{\circlearrowleft} v_j) \cdot \frac{1}{|\mathbb{U}| \cdot |\mathbb{W}|} \cdot \alpha_l, \quad (5)$$

where $\beta_k \geqslant 0$ can be interpreted as the probability of choosing a particular graph to take the random walk, and $\sum_{k=1}^{K} \beta_k = 1$.

B. Mining Syntagmatic Relation

Syntagmatic relation concerns adjacency and co-occurrence between words, and is usually sensitive to the order of words. This kind of correlation can be captured by round trips on the adjacency graph. If v_i and v_j co-occur a lot and v_i always occurs before v_j, the probability of taking a forward trip from v_i to v_j and then walking back from v_j to v_i is likely to be high. The advantage of round trip over one-way trip is that round trip will be less likely to be dominated by "popular" nodes which have a large number of outlinks or inlinks in the graph. In a round trip, even if it is quite easy for a random walker to reach a popular node, it is unlikely that the walker will easily return to the starting point from the popular node since there are too many paths to chose to walk back.

If the task for a random walker is to take an l-step forward walking from v_i to v_j and then return to v_i by an l-step backward walking, where l should be less than s and can be chosen in advance with a probability of α_l, then the total probability for the random walker to reach v_j as the destination (considering all the possible paths) is:

$$\begin{aligned} P_G^s(v_i \longrightarrow v_j) &= \frac{\sum_{l=1}^{s} P_G(v_i \overset{l}{\rightarrow} v_j) \cdot P_G(v_j \overset{l}{\dashrightarrow} v_i) \cdot \alpha_l}{\sum_{v_{j'} \in V} \sum_{l=1}^{s} P_G(v_i \overset{l}{\rightarrow} v_{j'}) \cdot P_G(v_{j'} \overset{l}{\dashrightarrow} v_i) \cdot \alpha_l} \\ &\propto \sum_{l=1}^{s} P_G(v_i \overset{l}{\rightarrow} v_j) \cdot P_G(v_j \overset{l}{\dashrightarrow} v_i) \cdot \alpha_l \end{aligned} \quad (6)$$

Here $\alpha \geqslant 0$ and $\sum_{l=1}^{s} \alpha_l = 1$.

We can define a similar task of backward-first round trip in which the first step is to take backward walking, and the probability of the random walker to successfully reach v_j is

$$P_G^s(v_i \longleftarrow v_j) = \frac{\sum_{l=1}^{s} P_G(v_i \overset{l}{\dashrightarrow} v_j) \cdot P_G(v_j \overset{l}{\rightarrow} v_i) \cdot \alpha_l}{\sum_{v_{j'} \in V} \sum_{l=1}^{s} P_G(v_i \overset{l}{\dashrightarrow} v_{j'}) \cdot P_G(v_{j'} \overset{l}{\rightarrow} v_i) \cdot \alpha_l}$$
$$\propto \sum_{l=1}^{s} P_G(v_i \overset{l}{\dashrightarrow} v_j) \cdot P_G(v_j \overset{l}{\rightarrow} v_i) \cdot \alpha_l \quad (7)$$

If $P_G^s(v_i \longrightarrow v_j)$ is high, it indicates that v_j is likely to share strong syntagmatic relationship with v_j in the order of taking v_i as predecessor, whereas $P_G^s(v_i \longleftarrow v_j)$ measures their syntagmatic relatedness by taking v_i as a successor. For example, $P_G^s(v_i \longrightarrow v_j)$ can be high when v_i represents "more" and v_j represents "than", and $P_G^s(v_i \longleftarrow v_j)$ could be high if v_j is "much". If we further make use of multiple adjacency graphs, we can finally measure the syntagmatic relation in the following way:

$$Syn(v_i \rightarrow v_j) = \sum_{k=1}^{K} \beta_k \cdot P_{G_k}^s(v_i \longrightarrow v_j) \quad (8)$$

$$Syn(v_i \leftarrow v_j) = \sum_{k=1}^{K} \beta_k \cdot P_{G_k}^s(v_i \longleftarrow v_j) \quad (9)$$

where $Syn(v_i \rightarrow v_j)$ is the syntagmatic relation between v_i and v_j by taking the order of v_i being followed by v_j, while $Syn(v_i \leftarrow v_j)$ is in the inverse order.

C. Scalability

The random walk algorithms proposed for mining paradigmatic and syntagmatic relations are very efficient and can scale well to potentially very large collections of text data. In many cases, the computation essentially boils down to sparse matrix multiplications and similar operations; such computations can be done very efficiently (see, e.g., [16]) and can also be parallelized (see, e.g., [5]). Furthermore, the complexity of the algorithms can be controlled empirically through imposing a constrain on the lengths of random walks. Such a thresholding strategy makes sense because words with strong associations are generally "close" on the constructed adjacency graph, and thus "local" mining of the graph can be expected to capture most of the useful lexical associations. Finally, both the construction of the adjacency graph initially and updating of the graph can also be done efficiently, allowing the algorithm to be deployed to potentially run continuously on real growing text data sets such as the World Wide Web.

VI. Evaluation

A. Data Sets

We use two different data sets to evaluate the proposed lexical relation mining algorithms. One is a set of news articles from TREC (AP88 and AP89), which contains 164,597 documents with a vocabulary size of 360,788. Another data set is from New York Times corpus, a collection of 20 years' articles (1987-2007). It contains 1,855,658 articles and the vocabulary size is 1,874,360.

B. Paradigmatic relation discovery

We first perform qualitative analysis of the results to examine whether the proposed circle-trip random walk algorithm can indeed discover interesting useful paradigmatic relations, and then perform quantitative evaluation to see whether the method will benefit more from larger corpus. In our experiment, we set s to be 1, as a result α_1 equals 1. K is set to be 2, while $\beta_1 = 0.8$ and $\beta_2 = 0.2$.

1) Qualitative analysis: In Table I, we show the top 10 words of the strongest paradigmatic relations with some selected words discovered from Ap8889. Most of the top associated words are in the same semantic category with the target word. For example, all of the top associated words for "Monday" are weekday names, while auxiliary verbs such as "can't", "will", "could" are retrieved for "can".

These meaningful results clearly demonstrate the effectiveness of our proposed algorithm for discovering paradigmatic relations. Note that while the relations of these words are common sense knowledge to humans, the mining algorithm was *not given any such knowledge*; yet the algorithm could discover such knowledge automatically through purely statistical analysis of semantic relations, suggesting that the proposed method is likely to work well on any text data in any natural language.

TABLE II
Paradigmatic relation retrieval.

Metric	Ap	NYT	NYT 2003	NYT 1999	NYT 1995	NYT 1991	NYT 1987
MAP	0.339	0.434	0.446	0.404	0.373	0.358	0.336
Pre@1	0.900	0.886	0.886	0.914	0.886	0.900	0.943
Pre@2	0.657	0.743	0.750	0.743	0.700	0.707	0.686
Pre@3	0.605	0.662	0.662	0.619	0.652	0.619	0.600
Pre@4	0.546	0.629	0.646	0.611	0.604	0.579	0.550
Pre@5	0.509	0.611	0.623	0.591	0.569	0.546	0.511
Pre@6	0.476	0.591	0.612	0.579	0.543	0.524	0.474
Pre@7	0.469	0.574	0.602	0.565	0.522	0.502	0.455
Pre@8	0.443	0.561	0.591	0.554	0.509	0.488	0.452
Pre@9	0.437	0.546	0.559	0.527	0.483	0.467	0.444
Pre@10	0.433	0.541	0.546	0.499	0.461	0.443	0.423

2) Quantitative evaluation: For the quantitative analysis of the paradigmatic relation mining, we conduct two evaluation experiments to test our algorithm, namely para-word retrieval experiment and TOEFL task experiment. In the para-word retrieval experiment, the discovery of paradigmatic relation is treated as a retrieval task. We sample 70 words as queries and retrieve the top 10 associated words ranked by paradigmatic relation for each of them. Finally we manually label the associated words to indicate whether they are relevant to their query word from the point of view of paradigmatic relation. Two standard information retrieval evaluation measures, MAP and precision [17], are both used as metrics in our evaluation. MAP captures the overall ranking accuracy and is the main measure, while precision at K items is much easier to interpret and reflects the utility of an algorithm.

We first make a comparison between the two data sets we use, namely, Ap8889 and NYT, to see how our algorithm performs on them respectively. In Table II we show the evaluation results in terms of multiple metrics. In general, our algorithm performs better on NYT corpus, which is in line

TABLE I
TOP 10 ASSOCIATED WORDS RETRIEVED BY PARADIGMATIC RELATION.

Monday	protein	more	year	intimacy	can	slightly	eggplant	Berkeley
Tue	protein	more	week	intimacy	can't	cent	eggplant	Berkeley
Thur	protien	less	year	contact	can	slightly	celery	Livermore
Mon	pellucida	stockier	month	intercourse	will	geffenplantinum	cantaloup	trucke
Thursday	renin	stouter	decade	encount	could't	sharply	kale	Arcata
Wednesday	icam	shrewder	half-century	assault	would	broadly	jalapeno	Irvine
Tuesday	insulin'secret	smoggier	day	relationship	don't	modestly	honeydew	Riverside
Sunday	feedstuff	faser	century	tryst	could	percent	melon	Cupertino
Friday	DNA	worse	year-and-a-half	relate	cannot	issue	whiterib	Greenbrae
Monday	gene-engine	dearer	quarter-century	liaison	must	outnumber	onion	Sacramento
Saturday	lecithin	faster	hour	abuse	should	mostly	watermelon	Lompoc

with our intuition since NYT is much larger. In order to see how the size of the data will impact the performance of our algorithm, we further conduct experiment on different subsets of NYT data. The NYT corpus we use is a collection covers articles from 1987 to 2007. To simulate natural progression of data growth and avoid bias caused by sampling as much as possible, we start from the subset of the articles of 1987 and gradually make the data larger and larger by adding new articles. In all, we sample 5 subsets by adding 4 year's articles at a time. For example, "NYT 2003" in Table II denotes the subset of articles from 1987 to 2003. We can see that the general trend is that the performance improves as the data grows, which indicates that the proposed algorithm can be expected to "respond" well to the growth of the data to enable discovery of more and higher quality of lexical knowledge as the data to be mined grows naturally. In other words, the algorithm can *turn big data into "big" knowledge*.

The second evaluation experiment for paradigmatic relation discovery is to apply our algorithm on a standard TOEFL task. The TOEFL task is first used by Landauer and Dumais [10]. There are 80 questions, each of which is to decide which of the four choice words is most relevant to the target word. For instance, a possible question can be "which of the following words is closest to "urgently" in terms of word meaning: "typically", "conceivably", "tentatively" or "desperately"?"

We test the TOEFL task on the 7 data set (Ap8889, NYT and 5 different subsets of NYT) and show the result in Table III. The accuracy score of Ap8889 data looks much lower than that of NYT. However, as reported in [14], human subjects were only able to solve 51.6 of the test items correctly, so the performance is still acceptable. Generally, better performance can be obtained as the data size grows larger. Different from the state-of-art method proposed in [4], we use our method on the whole data set without any filtering. In [4], stop words are removed. Another difference is that we do not employ any matrix factorization technique on the co-occurrence based transition matrix in our experiment. Matrix factorization is of high probability to help us to further improve the performance, but it is also very computationally expensive when applied to large matrix, thus not suitable for mining big text data.

C. Syntagmatic relation discovery

In the syntagmatic relation mining experiment, s is set to be 2. α_1 and α_2 are set to be 0.7 and 0.3 respectively. K is set to be 3. β_1 = 0.7, β_2 = 0.2 and β_3 = 0.1.

1) Qualitative analysis: Table IV shows the top-10 words that have the strongest syntagmatic relation with some selected words extracted from Ap8889 data set. In the 5 columns on the left side, words that tend to follow the target word are retrieved, whereas in the 5 columns on the right side, words always followed by the target word are retrieved. It is clear that the associated words are all semantically or syntactically related to the target word. Moreover, we can also see that these syntagmatic relations reveal interesting topics about a target word in this data set. For example, in the case of "Chinese", we can see the associated words show different aspects of topics, such as education and politics. It is very interesting that our method can discover distant pairs such as "neither ... nor" and "soon ...possible" as well.

2) Quantitative evaluation: We can view the discovery of syntagmatic relation as a retrieval process as well for the purpose of quantitative evaluation. Syntagmatic relation is asymmetric and is thus sensitive to the order of words. Given a word, we can both retrieve the words which are frequently followed by it (denoted as backward retrieval in the rest part of this section) and the words which frequently follow it (denoted as forward retrieval). We sample 20 words for each type of retrieval and make evaluation on both of the data sets. The results are shown in Table V. Generally, as the data size grows, the syntagmatic relation mined from the data set becomes more accurate. We can observe this trend from both forward retrieval and backward retrieval.

VII. CONCLUSIONS

In this paper, we propose a novel probabilistic approach to mine paradigmatic and syntagmatic relations from large text data based on random walks defined on word adjacency graphs. The approach is completely unsupervised and general, and thus can be applied to arbitrary text data in different natural languages without requiring manual effort. It is also efficient and easy to scale up to handle very large data sets. Furthermore, it has a solid statistical foundation based on stochastic processes and random walks, and thus can be adapted to perform different mining tasks in a principled way. Evaluation results show that the proposed algorithms are effective for discovering meaningful paradigmatic and syntagmatic relations of words from text data, and that for both relations, the algorithms respond well to the growth of big data and are able to generate higher quality of knowledge as we obtain more data for analysis, suggesting that they are effective for turning big data into "big knowledge".

TABLE III
TOEFL TASK.

Data Set	Ap	NYT	NYT 2003	NYT 1999	NYT 1995	NYT 1991	NYT 1987
Accuracy(%)	56.25	80	77.50	78.75	76.25	77.50	72.50

TABLE IV
TOP 10 ASSOCIATED WORDS RETRIEVED BY SYNTAGMATIC RELATION.

Chinese →	school→	express→	long→	mount →	→ information	→ market	→ spokesman	→ war	→ own
student	district	concern	island	Rushmore	classify	over-the-count	ministry	world	their
mainland	superintendant	regret	overdue	Everest	non-public	stock	police	civil	his
leader	dropout	satisfaction	time	Hermon	confidential	bond	department	Iran-Iraq	wholly
government	student	dismay	beach	Rainier	inside	exchange	house	Vietnam	its
opera	board	optimism	distance	Pleasant	withhold	the	a	star	our
author	teacher	wieczorny	way	Vernon	mcgrawhill	financial	embassy	postworld	my
dissident	diploma	gratitude	enough	Holyoke	IDD	broader	white	cold	her
cheongsam	principal	sympathy	enough	Clemen	sensitive	credit	army	eight-year	your
riben	bus	confidence	haul	Kisco	provide	secondary	pentagon	Korean	already
embassy	gymnasium	displeasure	ago	Tokachi	gather	BcCredit	FAA	beanfield	jointly

TABLE V
SYNTAGMATIC RELATION RETRIEVAL.

Metric	Forward Retrieval							Backward Retrieval						
	Ap	NYT	NYT 2003	NYT 1999	NYT 1995	NYT 1991	NYT 1987	Ap	NYT	NYT 2003	NYT 1999	NYT 1995	NYT 1991	NYT 1987
MAP	0.5792	0.6362	0.6164	0.5926	0.5589	0.5167	0.4280	0.5922	0.6682	0.6485	0.6311	0.6124	0.6474	0.5652
Pre@1	0.7500	0.9000	0.9500	0.9500	0.9500	0.9000	0.8000	0.8000	0.8000	0.8000	0.7500	0.6500	0.7500	0.7500
Pre@2	0.8000	0.9000	0.9500	0.9000	0.8500	0.7250	0.6250	0.8000	0.8000	0.7750	0.7500	0.7500	0.7500	0.7500
Pre@3	0.8000	0.8833	0.9000	0.8000	0.8167	0.6833	0.6167	0.7667	0.7667	0.7667	0.7833	0.7833	0.8000	0.7333
Pre@4	0.7875	0.8250	0.8250	0.8000	0.7750	0.6875	0.6000	0.7250	0.7750	0.7750	0.7875	0.7875	0.8000	0.7125
Pre@5	0.7500	0.8100	0.8200	0.7800	0.7400	0.6500	0.5800	0.7200	0.7900	0.7750	0.7700	0.7800	0.7800	0.7100
Pre@6	0.7167	0.7833	0.7750	0.7500	0.7417	0.6583	0.5917	0.7000	0.7917	0.7750	0.7667	0.7500	0.7750	0.7000
Pre@7	0.7071	0.7714	0.7571	0.7429	0.7143	0.6714	0.5786	0.6857	0.7929	0.7857	0.7643	0.7286	0.7643	0.7071
Pre@8	0.6938	0.7438	0.7188	0.7125	0.6750	0.6688	0.5688	0.6813	0.7875	0.7813	0.7500	0.7250	0.7438	0.7000
Pre@9	0.6833	0.7222	0.7000	0.7000	0.6611	0.6611	0.5556	0.6611	0.7778	0.7556	0.7333	0.7056	0.7389	0.6722
Pre@10	0.6700	0.7150	0.6850	0.6800	0.6550	0.6500	0.5500	0.6600	0.7600	0.7400	0.7250	0.7050	0.7300	0.6550

The representation of text data based on word adjacency graphs opens up a whole new promising strategy for mining big text data. The specific algorithms we propose have just scratched the tip of the iceberg. There are many interesting future directions for further exploration. First, the graph can be constructed based on semantic units such as entities and relations. Random walks on such a graph would be able to discover more interesting semantic knowledge of language than what we have explored in this paper. Even a direct application of our algorithms to a graph with entities as units (to replace words) would potentially be able to generate interesting results. Second, in addition to discover interesting associations of words, random walks can also be leveraged to mine topical sentences or phrases, or even subgraphs. Third, while we have motivated our work from text mining perspective, the proposed approach can be generalized to mine arbitrary sequence data for discovering interesting associations of elements.

VIII. ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation under Grant Number CNS-1027965.

REFERENCES

[1] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, Mar. 2003.

[2] M. Brand. A random walks perspective on maximizing satisfaction and profit. SIAM.

[3] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, Dec. 1992.

[4] J. A. Bullinaria and J. P. Levy. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and svd. *Behavior research methods*, 44(3):890–907, 2012.

[5] A. Buluc and J. R. Gilbert. Parallel sparse matrix-matrix multiplication and indexing: Implementation and experiments. *SIAM Journal on Scientific Computing*, 34(4):C170–C191, 2012.

[6] K. Ganesan, C. Zhai, and J. Han. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Comput. Linguist.*, pages 340–348, 2010.

[7] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Comput. Linguist.-Volume 2*, pages 539–545, 1992.

[8] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[9] I. Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. In *Proceedings SIGIR'02*, pages 195–202, 2009.

[10] T. K. Landauer and S. T. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*.

[12] M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. 1999.

[14] R. Rapp. The computation of word associations: comparing syntagmatic and paradigmatic approaches. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7, 2002.

[15] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

[16] R. Yuster and U. Zwick. Fast sparse matrix multiplication. *ACM Trans Algorithms*, 1(1):2–13, 2005.

[17] C. Zhai. *Statistical Language Models for Information Retrieval*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2008.