Mining Pure High-Order Word Associations via Information Geometry for Information Retrieval

Yuexian Hou, Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin University
Xiaozhao Zhao, Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin University
Dawei Song, Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin University and The Open University
Wenjie Li, The Hong Kong Polytechnic University

The classical bag-of-word models for Information Retrieval (IR) fail to capture contextual associations between words. In this paper, we propose to investigate the “pure high-order dependence” among a number of words forming an un-separable semantic entity, i.e., the high-order dependence that cannot be reduced to the random coincidence of lower-order dependencies. We believe that identifying these pure high-order dependence patterns will lead to a better representation of documents and novel retrieval models. Specifically, two formal definitions of pure dependence: Unconditional Pure Dependence (UPD) and Conditional Pure Dependence (CPD) are presented. The exact decision on UPD and CPD, however, is a NP-hard in general. We hence prove the sufficient criteria that entail UPD and CPD, within the well-principled Information Geometry (IG) framework, leading to a more feasible UPD/CPD identification procedure. We further develop novel methods to extract word patterns with pure high-order dependence. Our methods are applied to and extensively evaluated on three typical IR tasks: text classification, and text retrieval without and with query expansion.

Categories and Subject Descriptors: H.3.3 [INFORMATION STORAGE AND RETRIEVAL]: Information Search and Retrieval—Retrieval Models; H.2.8 [DATABASE MANAGEMENT]: Database Application—Data Mining; I.5.4 [PATTERN RECOGNITION]: Applications—Text Processing; G.3 [PROBABILITY AND STATISTICS]: Multivariate Statistics

General Terms: Theory, Measurement, Algorithms

ACM Reference Format:
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION
The classical bag-of-word models, such as the Vector Space Model (VSM) [Salton et al. 1975] and unigram language model (LM) [Ponte and Croft 1998], represent a docu-
dependence phrases or named entities. More generally they can be high-order association (also many cases, the semantic entities are not necessarily limited to syntactically valid phrases or named entities. More generally they can be high-order association (also referred to as high-order dependence) patterns, which are often beyond pair-wise relations, e.g. (“climate”, “conference”, “Copenhagen”). Such contextual high-order word association indicates that the words in the same association serve as context of each other, and form a high-level semantic meaning.

Recently, there have been attempts to extract term relationships from text to improve information retrieval (IR), e.g., through the Apriori method [Song et al. 2008] [Kirsch et al. 2012], closed frequent itemset [Zhong et al. 2012], co-occurrence analysis [Schütze 1998], syntactical and statistical phrase [Korkontzelos et al. 2008], and Word-net relations [Mihalcea and Corley 2006]. In this paper, we propose to consider high-order pure dependence, i.e., the high-order dependence that cannot be reduced to the random coincidence of lower-order dependencies. Usually these dependence patterns cannot be simply judged by co-occurrence frequencies. For example, the words a, the and of almost co-occur in every English article. However, we cannot say that they form a pattern representing a semantic entity. The high frequency of their co-occurrence can be explained as some kind of “coincidence”, because each of them or their pair-wise combinations has a high frequency, independently. On the other hand, the co-occurrence of the words “climate”, “conference” and “Copenhagen” implies an un-separable high-level semantic entity, which can not be fully explained as the random coincidence of, e.g., the co-occurrence of “Copenhagen” and “conference” (which can be any other conferences in Copenhagen) and the occurrence of “climate”. We consider a high-order dependence among words “pure”, if and only if the joint probability distribution of these words is significantly different from the product of lower-order joint distributions or marginal distributions, w.r.t all possible decompositions. In the language of graphical model, it requires that the joint distribution can not be factorized unconditionally (UPD) or conditionally (CPD).

Formally, given a set of binary random variables $X = \{X_1, \ldots, X_n\}$, where $X_i$ denotes the occurrence ($X_i = 1$) or absence ($X_i = 0$) of the $i$-th word. Let $x_i \in \{0, 1\}$ denote the value of $X_i$. Let $p(x) = \prod_{i=1}^n x_i$, where $x_i \in \{0, 1\}$ be the joint probability distribution over $X$. Then the $n$-order pure dependence over $X$ can be defined as follows.

**Definition 1.** (UPD): $X = \{X_1, \ldots, X_n\}$ is of $n$-order Unconditional Pure Dependence (UPD), iff it can NOT be unconditionally factorized, i.e., there does NOT exist a $k$-partition $\{C_1, C_2, \ldots, C_k\}$ of $X$, $k > 1$, such that $p(x) = p(c_1) \cdot p(c_2) \cdots p(c_k)$, where $p(c_i), i = 1, \ldots, k$, is the joint distribution over $C_i$.

One example of a multi-word combination whose joint distribution does not meet the UPD requirement is (“the”, “a”, “and”), since the occurrence of “the” or “a” in one document does not necessarily imply the occurrence of “and” in the same document and vice versa. Another example that satisfies the UPD constraint is the word combination (“Napoleon”, “Waterloo”, “Wellington”). This is because these words are usually tied together to describe the “Battle of Waterloo” event and the co-occurrence of two words will make the third word more likely to appear in the same document.

The next question is whether UPD is always enough to capture pure high order dependence. Consider the combination (“Einstein”, “theory of relativity”, “quantum physics”). These three phrases should be unconditionally dependent since Einstein was closely related to the development of both theory of relativity and quantum physics. However, given documents containing the word “Einstein”, the dependence between “theory of relativity” and “quantum physics” would seem to vanish, due to the fact
that “theory of relativity” and “quantum physics” are two relatively independent topics and would simultaneously occur in a context involving “Einstein” just by chance. The conditional independence between “theory of relativity” and “quantum physics”, given “Einstein”, implies that the semantic coupling among these three phrases is not as strong as that of (“Napoleon”, “Waterloo”, “Wellington”). In order to formulate the given “Einstein”, implies that the semantic coupling among these three phrases is not as strong as that of (“Napoleon”, “Waterloo”, “Wellington”). In order to formulate the above intuition and further clarify the semantic subtleness of pure dependence, the conditional pure dependence is proposed to eliminate conditional random coincidences.

**Definition 2.** (CPD): \( X = \{X_1, \ldots, X_n\} \) is of \( n \)-order Conditional Pure Dependence (CPD), iff it can NOT be conditionally factorized, i.e., there does NOT exist \( C_0 \subset X \) and a \( k \)-partition \( \{C_1, C_2, \ldots, C_k\} \) of \( V = X - C_0 \), \( k > 1 \), such that \( p(v|c_0) = p(c_1|c_0) \cdot p(c_2|c_0) \cdots p(c_k|c_0) \), where \( p(v|c_0) \) is the conditional joint distribution over \( V \) given \( C_0 \), and \( p(c_i|c_0) \), \( i = 1, 2, \ldots, k \), is the conditional joint distribution over \( C_i \) given \( C_0 \). In case that \( C_0 \) is an empty set, we define \( p(c_0) = 1 \).

**Remark 1.** Definition 2 permits an empty \( C_0 \). Hence CPD entails UPD.

To our best knowledge, there does not exist any efficient exact algorithm to decide on the above high-order pure dependence in both sufficient and necessary senses. In the graphical model configuration, it can be shown that the decision problem of UPD or CPD is NP-hard [Chickering et al. 2004]. Using probabilistic algorithms based on statistical hypothesis testing [Taskinen et al. 2005] [Bakirov et al. 2006], for a given bipartition \( \{C_1, C_2\} \), we can test whether \( p(x) = p(c_1) \cdot p(c_2) \) with time complexity \( O(n \cdot N^4) \), where \( n \) is the number of variables and \( N \) is the number of samples. Thus the decision of UPD is equivalent to testing \( 2^n \) bipartition configurations, resulting in a complexity approximate to \( O(2^n \cdot N^4) \).

Regarding the issue of efficiency, one may develop heuristics based on pair-wise dependence measures, e.g., covariance and correlation coefficient. However, a vanishing correlation (or covariance) does not necessarily imply the statistical independence. Chi-squared statistic can avoid the ad-hoc nature in tuning a threshold to decide on significant pure dependence. Chi-squared statistic does not necessarily imply the statistical independence. However, it does not guarantee the resulting associations are of pure dependence. The complete n-gram method is straightforward, but it often leads to a large amount of redundant and noisy information. In IR and computational linguistics, there exist methods for extracting significant n-grams in both syntactical and statistical manner [Caropreso et al. 2001], but the extraction of pure dependence is not addressed.

In this paper, we propose to use Information Geometry (IG) [Amari and Nagaoka 2000], which provides relevant theoretical insights and useful tools, to tackle these difficulties in a principled framework. IG studies joint distributions by way of differential geometry. A space of probability distributions is considered as a differentiable manifold, each distribution as a point on the manifold with the parameters of the model as coordinates. There are different kinds of coordinate systems to fit the manifold (detailed in Section 3), and it turns out that the so-called mixed coordinate systems with orthogonality are especially useful for our purpose. Based on the coordinate orthogonality, we can derive a set of statistics and methods for analyzing word dependence patterns by decomposing the dependence into various orders. As a result, the 2-order, 3-order and higher-order pure dependence can be singled out and identified by the log likelihood ratio test (LLRT).

In this paper, we propose two theoretically proven sufficient criteria for identifying UPD or CPD respectively, and the corresponding efficient implementations. The proposed IG-based methods can control the confidence level theoretically and offer a
approximate speedup ranging from $O(2^n N^2)$ to $O(2^n N^3)$, compared to the direct use of the method in [Taskinen et al. 2005] and [Bakirov et al. 2006].

The rest of the paper is organized as follows: Section 2 reviews related work on the extraction and utilization of high-order patterns in IR. Section 3 presents the derivation and proofs of the proposed pure high-order dependence extraction method within the IG framework. In Section 4, we clarify the spectrum of all kinds of high-order pure dependence defined in this paper. In Section 5, efficient implementations for mining the two types of pure dependence (UPD and CPD) are proposed. In addition, we also give some case studies illuminating the meaning of our main proposals. In Section 6, we conduct a series of experimental evaluation by applying the extracted high-order pure dependence patterns in the tasks of text classification, ad hoc retrieval and query expansion. Section 7 discusses several issues on the effectiveness and usability of different pure high-order dependence methods. Section 8 concludes the paper and gives the future work.

2. RELATED WORK

This paper focuses on effective and efficient extraction and utilization of high-order pure word dependence patterns in the context of IR.

There have been studies on incorporating dependence in language models. For example, [Niesler and Woodland 1996] presented a variable-length category-based n-gram language model, and [Zhang and Dong 2003] proposed a framework for combining n-grams of different orders. van Rijsbergen proposed to learn the document’s graphical structure by the mutual information between words, and used a dependence tree model to approximate the document representation [van Rijsbergen 1977]. [Gao et al. 2004] presented a dependence language model to incorporate grammatical linkages. [Song et al. 2008] presented methods for generating word associations based on association rule mining for query language modeling. The Markov Random Field (MRF) model captures short and long range term dependencies [Metzler and Croft 2005][Metzler and Croft 2007]. [Bendersky et al. 2010] further proposed a weighting scheme for the latent concepts used in MRF [Bendersky and Croft 2008] considered using noun-phrases to discover key concepts in verbose query, where the noun-phrases are extracted from queries and classified into key and non-key concepts based on pre-labeled training set. [Kumaran and Carvalho 2009] and [Xue et al. 2010] proposed to reduce long queries to more effective shorter ones (subsets of original query) by interactively removing extraneous terms.

Enhancements to the classical bag-of-word representation of documents have been introduced, e.g., via the use of 2-order co-occurrence information to build context vectors for word sense discrimination [Schütze 1998] and the combination with external knowledge (Wordnet) [Mihalcea and Corley 2006]. The use of both syntactical and statistical phrases for text categorization has been widely studied. [Mladenic and Grobelnik 1998] proposed to enrich the bag-of-word document representation by adding word sequences, which are selected based on feature selection methods (e.g. Odds Ratio feature score). [Caropreso et al. 2001] systematically investigated the usefulness of n-grams for document indexing in text categorization via a leaner-independent evaluation method, where the n-grams could be syntactical phrases (e.g. noun phrases), word sequences or unordered word combinations. In [Tzeras and Hartmann 1993], a Bayesian inference network model was proposed for automatic document indexing with terms from a prescribed vocabulary by using both linguistic phrase extraction and statistical analysis. [Zhong et al. 2012] proposed a pattern discovery technique based on the concept of closed frequent itemset, and a term weighting scheme based on the distribution of terms in the discovered patterns.
However, none of the above methods explicitly considered high-order pure dependence. In this paper, we propose to tackle this problem using IG. The IG is systematically introduced by [Amari and Nagaoka 2000] and has been successfully applied in the fields such as the study of neural spikes [Nakahara and Amari 2002]. Based on IG, [Hofmann 1999] defined a Fisher kernel for learning document similarities by Support Vector Machines (SVM). However, the issue of high-order pure dependence was not considered in his work. In general, the application of IG in text processing tasks is not yet widely studied.

3. INFORMATION GEOMETRY OF WORD ASSOCIATIONS

To illustrate our theoretical results and the corresponding algorithmic methods, it is necessary to explain the relevant background of IG [Amari 2002][Amari and Nagaoka 2000][Rao 1945][Jeffreys 1946]. Section 3.1 is devoted to introduce the fundamental concepts and definitions of IG, in particular the coordinate system of probability distributions used in IG. Section 3.2 illuminates the meaning of $\theta$-coordinate with IG orthogonality, which plays a central role in the identification of high-order pure dependence. Section 3.3 develops a log likelihood ratio test for $\theta$-coordinates. All the IG concepts and formalisms are mapped to and explained in the context of word associations.

3.1. Coordinates of Multivariate Joint Distributions

In IG, a family of probability distributions is considered as a differentiable manifold with certain coordinate systems. In the case of binary random variables, we often use three basic coordinate systems, namely $p$-coordinates, $\eta$-coordinates, and $\theta$-coordinates [Nakahara and Amari 2002]. To be specific, if we define an assignment over $X$, denoted by $a_X = <a_1,a_2,\ldots,a_n>$ (or $a_X = a_1a_2\ldots a_n$ in short), which determines a certain value of $x$ by assigning $a_i \in \{0,1\}$ to $X_i$, $1 \leq i \leq n$, then the coordinate systems of IG can be defined as follows:

(1) $p$-coordinates:

$$p_{a_X} = Pr\{X_1 = a_1, \ldots, X_n = a_n\} > 0$$

where $p_{a_X}$ is the joint probability and $a_i \in \{0,1\}, 1 \leq i \leq n$. Note that it is sufficient to determine a $n$-variable joint distribution using $2^n - 1$ probabilities, due to the constraint $\sum_{a_X} p_{a_X} = 1$. Also note that IG requires any probability term be non zero. This requirement can be met by using any common smoothing method.

In the three-word association case, $p(x)$ contains 8 joint probabilities, where, for example, the value $p_{101}$ means the probability of the first and the third words co-occurring, yet without the second word occurring. Any three-word joint distribution can be determined by arbitrary 7 out of 8 probabilities, e.g., \{p_{000}, p_{001}, p_{010}, p_{011}, p_{100}, p_{101}, p_{110}\}

(2) $\eta$-coordinates:

$$\eta_i = E[x_i], \ 1 \leq i \leq n$$

$$\eta_{ij} = E[x_i x_j], \ 1 \leq i < j \leq n$$

$$\vdots$$

$$\eta_{12\ldots n} = E[x_1 x_2 \ldots x_n]$$

Note that we define the order of a $\eta$-coordinate by the number of its subscripts. For example, $\eta_1$ is 1-order, and $\eta_{23}$ is 2-order.
In the IR context, $\eta$-coordinates capture the expectation of words co-occurring in the same documents. A $\eta$-coordinate is effectively equivalent to the document frequency of a single term or a term combination, up to a normalization factor. For example, assume we are considering the word dependence pattern $\{w_1, w_2\}$ in a dataset of 1000 documents. There are 410 documents containing $w_1$, 320 documents containing $w_2$, and 100 documents containing both $w_1$ and $w_2$. So we get $\eta_1 = 0.41$, $\eta_2 = 0.32$, $\eta_{12} = 0.1$. The transformation between $p$-coordinates and $\eta$-coordinates is trivial. As an example, we consider the case of $n = 3$. For the $p$-coordinate system, triple-word joint distribution can be determined by arbitrary 7 out of 8 probabilities, e.g., $\{p_{000}, p_{001}, p_{010}, p_{011}, p_{100}, p_{101}, p_{110}\}$. The transformation is as follows: $p_{111} = \eta_{123}, p_{011} = \eta_{23} - \eta_{123}, p_{100} = \eta_1 - \eta_{12} - \eta_{3} + \eta_{132},$ etc.

(3) $\theta$-coordinates: $\theta$-coordinates can be derived from the log-linear expansion of $p(x)$:

$$\log p(x) = \sum_i \theta_i x_i + \sum_{i < j} \theta_{ij} x_i x_j + \cdots + \theta_{123} x_1 x_2 x_3 - \Psi$$

where $\Psi$ is the normalization term corresponding to $\Psi = -\log p(0)$. It is easy to check that (3) is an exact expansion since all $x_i$’s are binary [Nakahara and Amari 2002]. Note that we can also define the order of a $\theta$-coordinate the same as in the $\eta$-coordinates.

Based on (3), $\theta$-coordinates can be given by the following equation in relation to $p$-coordinates:

$$\theta_I = \sum_{A \subseteq I} (-1)^{|I-A|} \log(p_A)$$

where $I$ is any nonempty subset of $\{1, 2, \ldots, n\}$ and $| \cdot |$ denotes the cardinality operator. And $p_A$ stands for the probability that all variables indicated by $A$ equal to one and the complemented variables are zero. For example, if $A = \{1, 3\}$ and $n = 3$, then $p_A = p_{101} = \text{Prob}(x_1 = 1, x_2 = 0, x_3 = 1)$. Note that when $A$ is null set, $p_{000}$ indicates the probability that all variables are zero. Then we could calculate $\theta$-coordinates of arbitrary order, e.g.,

$$\theta_1 = \log \frac{p_{100}}{p_{000}}, \quad \theta_{13} = \log \frac{p_{101} p_{000}}{p_{100} p_{001}}, \quad \theta_{123} = \log \frac{p_{111} p_{100} p_{101} p_{000}}{p_{110} p_{101} p_{011} p_{000}}.$$

Using the coordinate systems defined above, the set of all $n$-order joint probability distributions forms a $d$-dimensional manifold, where $d = 2^n - 1$.

3.2. Coordinate Orthogonality

A $\eta$-coordinate is indeed the co-occurrence probability of a set of variables, which measures the dependence among variables. Hence, by the definition of high-order pure dependence in Section 1, if a statistic measuring high-order dependence is uncorrelated with all lower-order $\eta$-coordinates, this statistic can be considered a metric of high-order pure dependence in the sense that it rules out the influence of a certain range of lower-order dependence. This intuition can be formulated by the coordinate orthogonality of IG.\footnote{We recommend that the beginners of Information Geometry to skip the Section 3.2 and the proof of Proposition 3.1.}

The coordinate orthogonality is defined by the vanishing Fisher information [Costa and Santos 2005]. Formally, the Fisher information of two coordinate parameters $\xi_i$
and \( \xi_j \) is defined as
\[
\hat{g}_{ij}(\xi) = E \left[ \frac{\partial \log p(x, \xi)}{\partial \xi_i} \frac{\partial \log p(x, \xi)}{\partial \xi_j} \right] 
\] (5)

Here \( E[\cdot] \) refers to the expectation w.r.t. the given \( \xi \) and different \( x \). The coordinate parameters \( \xi_i \) and \( \xi_j \) are called orthogonal if and only if \( g_{ij}(\xi) = 0 \) at any \( \xi \) [Nakahara and Amari 2002]. That is, their influences to the log likelihood function are uncorrelated.

A more technical meaning of orthogonality is that the Maximum Likelihood Estimations (MLE) of orthogonal parameters can be independently performed, and hence we can design a simple log likelihood ratio test, in which the test statistic can be readily solved [Nakahara and Amari 2002]. Next, we explain this important claim in detail.

Let \( \xi = [\xi_1, \ldots, \xi_d]^T \) be the parameter vector of \( p(x; \xi) \). Given an independently and identically distributed sampling set \( \{x_1, \ldots, x_N\} \) of \( p(x; \xi) \), the log likelihood function of parameter \( \xi \) is computed by
\[
\log [L(\xi)] = \sum_{k=1}^{N} \log[p(x_k; \xi)]
\] (6)

The maximum likelihood estimation searches for the maximum of \( \log [L(\xi)] \) and gives the corresponding \( \xi \) as its solution. Let \( d\xi \) be a \( d \)-dimensional column vector whose all elements are zeros except that the \( i \)-th element is a small increment \( d\xi_i \). It is obvious that \( \hat{\xi}_j \) is independent of the value of \( \hat{\xi}_i \), as long as the following equation holds for each \( \xi \):
\[
\frac{\partial \log [L(\xi)]}{\partial \xi_j} = \frac{\partial \log [L(\xi + d\xi_i)]}{\partial \xi_j}
\] (7)

Substituting \( \log L(\xi + d\xi_i) = \log L(\xi) + \frac{\partial \log [L(\xi)]}{\partial \xi_i} d\xi_i \) in (7), we see that (7) is equivalent to
\[
\frac{\partial^2 \log L(\xi)}{\partial \xi_i \partial \xi_j} = 0
\] (8)

Assuming a sufficiently large sample and the commutativity between \( \partial \) and \( \sum \), (8) is equivalent to \(-E[\partial^2 \log p(x; \xi)] = 0\). The left hand side of the latter equation is an alternatively equivalent definition of \( g_{ij} \) in Formula (5).

Under the orthogonality guarantee between \( \xi_i \) and \( \xi_j \), the estimation procedure can find the MLE solutions of \( \xi_i \) and \( \xi_j \) separately. Hence, if we can construct a mixed coordinate system, in which the highest-order coordinate parameter is a proper metric of high-order pure dependence and orthogonal to all lower-order \( \eta \)-coordinates, the log likelihood ratio test (See Section 3.3 for details) for the highest-order parameter will be greatly simplified.

The mixed coordinate system satisfying the above requirements generally exists, due to the following two observations. First, it can be verified that \( \theta_{12\ldots n} \) is orthogonal to any \( \eta \)-coordinate less than \( n \)-order [Nakahara and Amari 2002]. Second, in Section 4, we prove that a significant nonzero \( n \)-order \( \theta \)-parameter entails the \( n \)-order CPD/UPD. Hence the expected \( (2^n - 1) \)-dimensional mixed coordinate system, denoted as \( \zeta \)-coordinates, can be given by \( [\eta_1, \ldots, \eta_{n-1}, \theta_{12\ldots n}]^T \), where \( \eta_1 = [\eta_1, \ldots, \eta_n]^T \), \( \eta_2 = [\eta_1, \eta_{13}, \ldots, \eta_{(n-1)n}]^T \) and so on.

### 3.3. Coordinate Parameter Estimation

The \( \theta \)-coordinates plays a central role in the identification of high-order pure dependence. However, a direct computation for high-order \( \theta \)-coordinates can be numerical-
ly unstable. In addition, we desire a quantitative statistical significance level of the investigated \( \theta \)-coordinate. Owing to the orthogonality between \( \eta \)-coordinates and \( \theta \)-coordinates, [Nakahara and Amari 2002] developed an efficient framework of Log Likelihood Ratio Test (LLRT) for \( \theta \)-coordinates. However, the computation of high-order \( g_{dd} \) (the bottom-right element of the Fisher information matrix of \( \zeta \)-coordinates) was left as an open problem, which is indeed a necessary step for implementing the LLRT framework. To facilitate the LLRT framework, in the following Proposition 3.1, we develop a closed-form formula for computing \( g_{dd} \) in general. 

**PROPOSITION 3.1.**

\[
g_{dd} = \frac{1}{\sum_x 1/p(x)} \tag{9}
\]

**PROOF.** The detailed proof of this proposition is given in Appendix A.1. \( \square \)

As an example of how to calculate \( g_{dd} \) in practises, consider the case of \( n = 2 \), we have

\[
g_{33} = \frac{1}{1/p_{00} + 1/p_{01} + 1/p_{10} + 1/p_{11}} = \frac{\eta_{12}(\eta_1 - \eta_{12})(\eta_2 - \eta_{12})(\eta_1 + \eta_2 - \eta_{12} - 1)}{\eta_1\eta_2(\eta_1 + \eta_2 - 1 - 2\eta_{12} + \eta_{12}^2)}
\]

where \( \eta_1, \eta_2, \eta_{12} \) can be easily estimated by the document frequency of a single term or a term combination.

Based on Proposition 3.1, the Log Likelihood Ratio Test (LLRT) can be generally implemented in the mixed coordinates \( \{\eta, \theta_{12...n}\}^T \), where \( \eta \) is the abbreviation of \( \{\eta_1, \ldots, \eta_{n-1}\} \). The LLRT tests whether \( \theta_{12...n} \) is significantly different from zero. Hence, we have the null hypothesis \( H_0 : \theta_{12...n} = 0 \), against \( H_1 : \theta_{12...n} \neq 0 \). Then two MLE statistics are calculated:

\[
l_0 = \log p(x; \hat{\eta}_0, 0), \quad l_1 = \log p(x; \hat{\eta}_1, \hat{\theta}_{12...n})
\]

The \( \hat{\eta}_1 \) and \( \hat{\theta}_{12...n} \) can be directly obtained from the maximum likelihood estimation of \( p \)-coordinates, which is indeed the sampling distribution. For estimation of \( \hat{\eta}_0 \) w.r.t \( \theta_{12...n} = 0 \), \( \hat{\eta}_0 \) turns out to be identical with \( \hat{\eta}_1 \), since the orthogonality between \( \eta \) and \( \theta_{12...n} \) guarantees that the maximum likelihood estimate of \( \eta \) is invariant when \( \theta_{12...n} \) changes. Hence, we denote \( \hat{\eta}_0 \) and \( \hat{\eta}_1 \) by \( \hat{\eta} \). The \( \rho \) statistic [Nakahara and Amari 2002] of LLRT is derived as follows:

\[
\rho = 2 \log \frac{l_1}{l_0} = 2 \sum_{i=1}^N \log \frac{p(x_i; \hat{\eta}, \hat{\theta}_{12...n})}{p(x_i; \hat{\eta}_0, 0)} \approx 2N \cdot E \left[ \log \frac{p(x; \hat{\eta}, \hat{\theta}_{12...n})}{p(x; \hat{\eta}_0, 0)} \right] \\
= 2N \cdot D \left[ p(x; \hat{\eta}, \hat{\theta}_{12...n}) : p(x; \hat{\eta}_0, 0) \right] \approx N g_{dd} \hat{\theta}_{12...n}^2 \tag{10}
\]

Here \( N \) is the number of samples, \( D[\cdot : \cdot] \) denotes the Kullback-Leibler divergence, \( g_{dd} \) is the right-bottom element of the Fisher information matrix \( G_\zeta \) of the mixed coordinates \( \zeta \) at point \( [\hat{\eta}; \hat{\theta}_{12...n}]^T \), and its value can be estimated by Formula (9). Also note that the first approximation \( (=) \) in Formula (10) holds in an asymptotical sense, and the second approximation is entailed by the approximate relation between Kullback-Leibler divergence and Riemannian distance induced by the metric tensor \( G_\zeta \) [Nakahara and Amari 2002].

Asymptotically, we have \( \pm \sqrt{\rho} \sim N(0, 1) \) and \( \rho \sim \chi^2(1) \), i.e., the \( \chi^2 \) distribution with degree of freedom 1. Hence, we can set up the confidence intervals corresponding to the expected confidence level.

\footnote{Recently, Nakahara independently derived a theoretical result (not published yet) similar to Proposition 3.1 (according to our personal communication with Amari and Nakahara).}
We can then get the estimated confidence level $\pi$ to indicate the degree that we believe a given pattern is of pure dependence in accordance with the statistic $\rho$. Formally, given the $\rho$ statistic from Formula (10), the confidence level $\pi$ can be determined by the cumulative distribution function of $\chi^2(1)$:

$$\pi = cdf\chi^2(1)(\rho)$$

(11)

For example, for a pattern $\{w_1, w_2\}$ with $\rho = 5.024$, we have a $\pi(w_1, w_2) = 95\%$ confidence to reject the null hypothesis. That is, we can ensure that $w_1$ and $w_2$ are of pure 2-order dependence with probability 95%.

4. THE SPECTRUM OF HIGH-ORDER PURE DEPENDENCE

In this section, we first introduce two operational definitions on high-order pure dependence, namely Pair-wise Pure Dependence (PPD) and Theta Pure Dependence (TPD), which are the sufficient criteria of UPD and CPD, respectively. Note that, from an algorithmic perspective, PPD or TPD are far more feasible than directly deciding UPD or CPD. Finally, we clarify the spectrum of all kinds of high-order pure dependence defined in this paper.

**Definition 3.** (PPD): $X = \{X_1, \ldots, X_n\}$ is of $n$-order Pair-wise Pure Dependence (PPD), iff every 2-order $\theta$-coordinate $\theta_{ij}$, $1 \leq i < j \leq n$, is significantly different from zero.

**Definition 4.** (TPD): $X = \{X_1, \ldots, X_n\}$ is of $n$-order Theta Pure Dependence (TPD), iff the $n$-order $\theta$ coordinate $\theta_{12\ldots n}$ is significantly different from zero.

In Definitions 3 and 4, the significance level can be decided w.r.t the LLRT described in Section 3.3.

The following proposition shows that the PPD property is indeed equivalent to the requirement that every pair of variables are significantly dependent.

**Proposition 4.1.** Two random variables $X_1$ and $X_2$ are independent iff the 2-order $\theta$-coordinate $\theta_{12} = 0$.

**Proof.** 1) Assume $X_1$ and $X_2$ are independent. We have $p_{00} = p_0 \cdot p_0, p_{01} = p_0 \cdot p_1, p_{10} = p_1 \cdot p_0$ and $p_{11} = p_1 \cdot p_1$, where $p_0 = p_{00} + p_{01}, p_0 = p_{00} + p_{10}$ and so on. It turns out that $\theta_{12} = \log \frac{p_{11}p_{00}}{p_{10}p_{01}} = 0$.

2) Assume $\theta_{12} = 0$, i.e., $p_{11}p_{00} = p_{10}p_{01}$. The independence of $X_1$ and $X_2$ can be shown by the following direct computation:

$$p_0 \cdot p_0 = (p_{00} + p_{01})(p_{00} + p_{10}) = p_{00}p_{00} + p_{00}p_{10} + p_{01}p_{00} + p_{01}p_{10}$$

$$= p_{00}p_{00} + p_{00}p_{10} + p_{00}p_{01} + p_{00}p_{11} = p_{00}$$

We can also verify $p_0 p_1 = p_{01}, p_1 p_0 = p_{10}$ and $p_1 p_1 = p_{11}$ in the same manner. Then the proposition follows. □

Proposition 4.1 implies that the LLRT on 2-order $\theta$-coordinate (2-order PPD) is indeed a test of independence and hence can be replaced, in principle, by any common independence test, e.g., Chi-squared test.

The following two propositions show the spectrum relation between PPD, TPD, UPD, and CPD.

**Proposition 4.2.** PPD $\Rightarrow$ UPD.

**Proof.** We will prove $\neg$UPD $\Rightarrow$ $\neg$PPD. Assume $X = \{X_1, \ldots, X_n\}$ does NOT have the $n$-order UPD, i.e. there exists a nontrivial partition $\{C_1, C_2, \ldots, C_k\}$ of $X$, such that $p(x) = p(c_1) \cdot p(c_2) \cdots p(c_k)$. Without loss of generality, we assume that $X_1$ and $X_2$
belong to \( C_1 \) and \( C_2 \), respectively. Summarizing all variables of \( p(x) \), except for \( X_1 \) and \( X_2 \), we have \( \sum_{x_3 \ldots x_n} p(x) = p(x_1)p(x_2) \). Hence, \( X_1 \) is independent of \( X_2 \), and \( \theta_{12} \) vanishes by the definition of \( \theta \)-coordinates (4). The proposition follows. \( \square \)

**Remark 2.** Proposition 4.2 only offers a sufficient condition for deciding UPD, since there exists UPD probability distribution without the property of PPD.

The following propositions show that TPD is the most strict form of pure dependence in the sense that it implies both UPD and CPD.

**Proposition 4.3.** \( TPD \Rightarrow CPD \)**

**Proof.** The detailed proof of this proposition is given in Appendix A.2. \( \square \)

**Corollary 4.4.** \( TPD \Rightarrow UPD \)**

**Proof.** It directly follows from \( TPD \Rightarrow CPD \) and \( CPD \Rightarrow UPD \). \( \square \)

The spectrum of pure high-order dependencies is summarized by the Venn diagram in Fig. 1.

**5. Implementation and Illustration**

**5.1. Association Mining for PPD**

PPD requires that every pair of variables are significantly dependent of each other. In order to determine whether \( n \) variables form a PPD pattern, we need to perform \( C_n^2 \) times of LLRT tests on the involved 2-order \( \theta \) parameters. In each 2-order LLRT procedure, we need to traverse all samples (documents) to obtain the corresponding four \( p \)-coordinates and compute the corresponding \( g_{33} \). These steps take \( O(N) \) time, where \( N \) is the number of samples. Hence the procedure of identifying \( n \)-order PPD takes \( O(n^2N) \) time in total.

As described in Section 4, the 2-order LLRT test could be replaced by a common independence test, such as Chi-squared test. In this case, similar to LLRT, we also need to obtain the four \( p \)-coordinates. Hence the total time required to decide PPD based on Chi-squared test remains to be \( O(n^2N) \).

Note that although the Chi-squared test is, in principle, equivalent to 2-order LLRT test (Proposition 4.1), the former tends to be more sensitive to data sparseness since it normally requires no more than 20% of the events have expected frequencies below
5 (even 10). Otherwise, Yates’ correction may be applied. Unfortunately, Yates’ correction tends to overcorrect and result in a conservative result that fails to correctly reject the null hypothesis (type II error). So it is suggested that Yates’ correction is unnecessary even with quite low total sample sizes, e.g., 20 [Sokal and Rohlf 2012]. On the other hand, 2-order LLRT can be considered more robust against sparseness since its asymptotical performance is not severely limited by low-frequency events as the former is. We conduct a simple case study in Section 5.5, which gives us an illustration on the different behaviors of Chi-squared test and LLRT with respect to data sparseness.

In practice, we are often interested in finding all maximal PPD patterns up to a given order $n_0 \leq n$. Here the maximal PPD pattern refers to the PPD pattern that cannot be enlarged. This is equivalent to the maximal clique problem over the graph generated by the following rules: (1) A variable is denoted by a vertex; (2) An edge connects two vertices iff the corresponding two variables form a 2-order PPD pattern. As Tsukiyama et al. showed [Tsukiyama et al. 1975], it is possible to list all maximal cliques in a graph in polynomial time per generated clique. Hence our problem can be efficiently solved if the number of all maximal PPD patterns, up to $n_0$-order, is a polynomial function of $n_0$. The number of PPD patterns can be controlled by an appropriate significance level of LLRT (or Chi-squared test).

Based on the above analysis, we propose Algorithm 1 for extracting the PPD patterns from a textual collection. We first pick all the words (after stemming and stop words removal), whose document-frequency is no less than a constant threshold $\text{min}_f$. We then generate all 2-word co-occurrence patterns as candidates, and check all the candidates to get the pure 2-order dependence pairs. In general, if there exist two different pure $k$-order dependence patterns with only one different element, we only need to check the PPD between such two elements to decide if these two $k$-order pure dependence form a pure $(k + 1)$-order dependence. For example, if we have extracted two 3-order dependence \{\(w_1, w_2, w_3\)\} and \{\(w_1, w_2, w_4\)\}, we only need to check whether there is a pairwise pure dependence between $w_3$ and $w_4$. If yes, we get a 4-order dependence \{\(w_1, w_2, w_3, w_4\)\}. The pseudo-code of the algorithm is shown in Algorithm 1.

By induction, the above algorithm is an exact algorithm in the sense that it can find all PPD patterns. In practice, we can use an input parameter to control the max order of search. In the detailed implementation, we need to check whether a word pair has the pure 2-order dependence. This can be done by a 2-order LLRT or Chi-squared test. Hence we implement two versions of PPD mining algorithms, namely PPD-CS (with Chi-squared test) and PPD-IG (with LLRT test).

5.2 Association Mining for TPD

The LLRT is more significant in the TPD framework since the identification of $n$-order ($n > 2$) TPD can NOT be implemented by combining and reusing some existing sub-procedures, e.g., Chi-squared test. Indeed, in order to decide whether $n$ variables form a TPD pattern, we need only to perform a single LLRT on the involved $n$-order $\theta$ parameter. The estimation of a $n$-order $\theta$ takes $O(N)$ time. Hence, the identification of a $n$-order TPD only takes $O(N)$ time in total.

Mining all TPD patterns, up to $n_0$-order, is much more time-consuming since high-order TPD patterns can not be derived from the lower-order TPD patterns. Hence we adopt a filter-and-refine strategy by first pre-selecting a set of candidates patterns. Two choices can be considered: (1) all PPD patterns up to $n_0$-order; (2) All frequent co-occurrence patterns up to $n_0$-order. To finally identify the TPD patterns, we then test whether the corresponding $\theta$-coordinates of the candidate patterns are significantly different from zero.
ALGORITHM 1: Association Mining for PPD

\[ L_1 \leftarrow \emptyset; \]
for every word \( w \) where \( df_w \geq \min_f \) do
\[ L_1 = L_1 \cup \{\{w\}\}; \]
end
\[ k \leftarrow 2; \]
while \( L_{k-1} \) is not empty do
\[ L_k \leftarrow \emptyset; \]
for \( a, b \in L_{k-1} \) do
\[ \text{if } |a \setminus b| = 1 \text{ AND } |b \setminus a| = 1 \text{ then} \]
\[ \text{if } a \cup b - a \cap b \text{ is 2-order pure dependence then} \]
\[ L_k \leftarrow L_k \cup \{a \cup b\} \]
end
end
\[ k \leftarrow k + 1; \]
end
\[ L \leftarrow L_2 \cup L_3 \cup \ldots \cup L_{k-1}; \]
for \( a, b \in L \) do
\[ \text{if } a \subset b \text{ then} \]
\[ L \leftarrow L \setminus \{a\}; \]
end
end
return \( L \);

Note that \( L_k (k \in \mathbb{N}) \) is the set of all word patterns having \( k \)-order pure dependence. Also note that the test of 2-order pure dependence could be LLRT or Chi-squared test.

5.3. Comparative Methods for High-order Term Extraction

In the fields of IR and computational linguistics, there exist various high-order term extraction methods in both syntactical and statistical manners. In this paper, we compare our method with the following two typical ones.

5.3.1. Syntactical Phrases. In linguistic sense, a syntactical phrase follows a grammar of language (English in this paper), and is a textual unit usually larger than a word but smaller than a sentence: examples of noun phrases are information retrieval, President Obama; examples of verb phrases are play basketball and reading books. Different pattern-based models (linguistic filters) have been proposed to identify high-order syntactical phrases [Korkontzelos et al. 2008], and particularly noun phrases are considered as important to identify key concepts [Bendersky and Croft 2008]. In this paper, we adopt a linguistic filter that accepts noun phrases, denoted as NP. More specifically, the noun phrases we adopt here are sequences of words with POS-tags, matching the following pattern:

\[ ((\text{Adj}|\text{Noun}) + |((\text{Adj}|\text{Noun}) * (\text{NounPrep}?))(\text{Adj}|\text{Noun})*)\text{Noun}. \]

For example, for the sentence with POS-tags below:

\"The_DT problem>NN of_IN searching_VBG for_IN patterns>NNS in_IN data>NNS is_VBZ a_DT fundamental JJ one_CD and_CC has_VBZ a_DT long JJ and_CC successful JJ, history>NN.\"

The noun phrases are "problem", "patterns", "data", "fundamental one" and "long and successful history".

5.3.2. Statistical Phrases. By statistical phrase, we mean any sequence of words that occur continuously and significantly in text. It can be viewed as a generalization of "n-grams". An n-gram is a contiguous sequence of n items from a given sequence of text or speech. [Caropreso et al. 2001] defines n-gram as "an alphabetically ordered
sequence \(g_n\) of \(n\) unigrams, where the occurrence of an \(n\)-gram \(g_n\) in a document \(d\) indicates that a permutation of \(g_n\) appears in \(d\) sequentially, after stop word removal and stemming. Inspired by this definition of \(n\)-gram, we generalize the problem of extracting frequent \(n\)-grams to the problem of finding frequent itemset within a sliding window and the sequential constraint of the words is dropped. The reason for making such generalization is to reduce the redundant information with \(n\)-grams (i.e., we think, in IR context, that a single combination of \(n\) words carries as sufficient semantic meaning as the corresponding set of all \(n\)-grams, which are not necessarily all meaningful). Note that in Markov Random Field model (MRF) [Metzler and Croft 2005] for IR (Section 6), the sequential word patterns are already incorporated in the model itself and can be switch on/off based on the corresponding threshold. Thus the problem of mining frequent \(n\)-grams is to find the itemsets whose supports are greater than a user predefined threshold, \(min\_sup\), where the support for an itemset is defined as the number of documents where it occurs. The frequent itemsets can be extracted efficiently using the Apriori algorithm [Agrawal et al. 1993], by considering terms as items and segments of text (e.g., through a sliding window) as transactions [Song et al. 2008]. This extraction algorithm is denoted as Apr in the rest of the paper.

5.4. Examples of High-order patterns

In this section, we conduct high-order term extraction on the Robust2004 corpus by utilizing five methods, denoted as Noun Phrase (NP), Frequent Itemsets (Apr), Chi-squared PPD (PPD-CS), IG-based PPD (PPD-IG) and Theta Pure Dependence (TPD).

For comparison, all five methods are used to extract the high-order patterns contained in the \(<desc>\) field of TREC topic 689. We perform a query (\(<desc>\) field of topic 689) by a classic unigram language model [Ponte and Croft 1998] on data collection ROBUST2004, and the top 1000 retrieved documents are used for high-order pattern extraction. For NP, the OpenNLP \(^3\) toolkit is used for POS tagging and chunking. All patterns are separated into two types according to their order, namely 2-order and 3-order. For Apr, top four patterns with highest support (relative frequency) are reported for each order. For PPD-CS, PPD-IG and TPD, we also show four patterns with highest confidence level for each order. Note that the confidence level of TPD patterns and 2-order PPD patterns can be determined directly based on Formula (11). The confidence level of a 3-order PPD pattern is determined by the product of the confidence levels of all pairwise 2-order patterns included in the corresponding 3-order PPD pattern. The significance levels for Chi-squared test and 2-order LLRT are both set to 0.05, a typical setting for statistical significance test. For all patterns, the values of support and confidence level of LLRT test are reported, respectively.

The results are shown in Table I. Note that only two 2-order NPs exist in original topic description. Let us first compare the high-order patterns in the sense of “purity” defined in Section 1. In the 3-order case, patterns extracted by Apr mostly consist of terms with high frequencies, such as “famili”, “support” and “provide”. Moreover, those frequent terms in Apr patterns tend to also occur in 2-order patterns (e.g., “famili provid support” covers two 2-order patterns “famili provid” and “provid support”). This means that some of the high-order frequent itemsets are “unpure” (redundant), since they could be degenerated to the combination of low-order patterns. On the other hand, we can see that PPD and TPD discover some patterns with relatively lower frequencies. For example, “countri famili plan” is extracted by both PPD and TPD, which is of low frequency (0.0096). Another example is that some low frequency words are also included in the top four TPD patterns, such as “refus” with marginal frequency

\(^3\)The Apache OpenNLP library is a natural language processing toolkit for the processing of natural language text, such as tokenization, sentence segmentation, part-of-speech tagging, chunking and so on.
Table I: Samples of high-order dependence
For Apr, PPD-CS, PPD-IG, TPD

<table>
<thead>
<tr>
<th>2-order patterns</th>
<th>Pattern</th>
<th>Frequency</th>
<th>LLRT</th>
<th>3-order patterns</th>
<th>Pattern</th>
<th>Frequency</th>
<th>3-order LLRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
<td>famili provid support</td>
<td>0.0301</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>limit support</td>
<td>0.0249</td>
<td>0.5498</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>famili plan</td>
<td>0.0852</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>plan provid support</td>
<td>0.0843</td>
<td>0.9996</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apr</td>
<td>famili provid support</td>
<td>0.0301</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>famili plan</td>
<td>0.0852</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>plan provid support</td>
<td>0.0843</td>
<td>0.9996</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PPD-CS</td>
<td>aid countri</td>
<td>0.0232</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>countri famili</td>
<td>0.0186</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>aid famili</td>
<td>0.0232</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>famili plan</td>
<td>0.0852</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PPD-IG</td>
<td>aid countri</td>
<td>0.0232</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>countri famili</td>
<td>0.0186</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>aid famili</td>
<td>0.0232</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TPD</td>
<td>countri famili</td>
<td>0.0186</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>aid countri</td>
<td>0.0232</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>famili plan</td>
<td>0.0852</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>famili limit support</td>
<td>0.0249</td>
<td>0.5498</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>countri limit refus</td>
<td>0.0005</td>
<td>0.9609</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note that the 3-order Apr pattern with the highest frequency (0.3031) is rejected by LLRT test with the confidence level 1, while the 3-order Apr pattern with the lowest frequency (0.0005) is accepted by LLRT test with the confidence level 0.9609.

0.0076, which is lower than typical marginal frequency (ranging from 0.05 to 0.1 in our example). Moreover, PPD and TPD could also eliminate the “unpure” patterns to some extend. The top Apr pattern: “famili provid support”, which overlaps with two 2-order Apr patterns, is of zero score by TPD and does not belong to the set of the top four PPD-IG patterns.

We now look at the coverage power of high-order patterns to represent the given topic. The Apr method tends to give a far more incomplete coverage of the whole word set since there are heavy overlaps between the Apr patterns. Especially, the 3-order Apr patterns are often overlapped with each other with more than one word (e.g., “famili provid support” and “famili plan support”). In contrast, PPD and TPD would greatly reduce this kind of redundancy, and are able to extract novel high-order patterns with low marginal frequency. It turns out that PPD and TPD patterns give us a relatively complete coverage than Apr patterns. For example, the union set of top four 3-order Apr patterns contains only four different words, i.e., “[famili “provid “support “plan”], while PPD or TPD patterns include six or eight different words, respectively. We think that the higher coverage power contributes to the better usefulness of IG patterns in IR (to be shown in Section 6).

5.5. Robustness Test: 2-order Case Study

This case study aims to test the stability of different methods for 2-order independence test (covariance statistic, Chi-squared test and IG method) on sparse data. We used the Reuters-21578 dataset, from which 10% documents were randomly selected to form a small subset supposed to be sparse, to some extent. 1000 word pairs were randomly selected from the pure 2-order dependence extracted by our IG method. Then we applied the IG method on the whole dataset which is less sparse. About 79% of the 1000 word pairs were still extracted as 2-order pure dependence. However, when we did the same testing for the covariance statistic and Chi-squared test, the percent-
ages were only about 32% and 75%. This experimentally shows that the IG method is more robust w.r.t data sparsity. As theoretically described before in Section 5.1, one reason for this observation is due to the asymptotic property of the fitness of IG-based method, which is less sensitive to the data sparsity compared to Chi-squared test and covariance statistic.

6. APPLICATION AND EVALUATION

By now, we have presented a unified framework based on IG for extracting pure high-order word associations, which are expected to better capture high level semantic meanings. Intuitively, those semantic entities are more descriptive than single terms, and can be used to enrich document/query representation and improve the effectiveness in IR tasks. In this section, we first propose to extend the VSM using high-order patterns to enhance the document representation, and conduct a systematical study on text categorization task. However, in the ad hoc retrieval and query expansion tasks where the document collection is in much larger scale, the application of high-order patterns is not so straightforward, due to scalability problem if we directly expand the vector representation of documents. Therefore, we adopt the Markov Random Field (M-RF) retrieval model [Metzler and Croft 2005] [Metzler and Croft 2007] which provides a suitable framework and testbed to incorporate high-order patterns.

6.1. Text Classification: Extended Vector Space Model with Pure High-order Dependence

Text classification aims at automatically grouping a set of documents into a predefined set of categories (classes). There are extensive studies on using the syntactical and statistical phrases in text categorization [Mladenic and Grobelnik 1998] [Caropreso et al. 2001]. In this section, we investigate the effectiveness of extending the VSM with high-order dependencies extracted by different methods in the context of text categorization. These methods include Noun Phrase (NP), Frequent Itemset (Apr), Chi-squared based PPD (PPD-CS), IG-based PPD (PPD-IG) and TPD.

Given a document set \( D = \{d_1, d_2, \ldots, d_M\} \) and its vocabulary \( \{w_1, w_2, \ldots, w_n\} \), the classical vector space model (VSM) represents each document as a \( n \)-dimensional vector: \( d_i = [y_1, y_2, \ldots, y_n]^T \), where \( y_j \) is the weight of \( w_j \) to indicate the importance of the word to represent the document. In this experiment, we adopt the commonly used TF-IDF weighting scheme: \( y_j = tf_{w_j} \cdot \log \frac{M}{M_{w_j}} \), where \( tf_{w_j} \) is the term frequency of \( w_j \) in document \( d_i \) and \( M_{w_j} \) is the number of documents in \( D \) containing term \( w_j \). \( M \) is the total number of documents.

The k nearest neighbor (k-NN) algorithm is used as the classifier. Given a test document \( d_i \), the k-NN finds its k nearest neighbors among the training documents (labeled by categories), where the affinity between two documents is measured by the Cosine similarity of corresponding document vectors. Then majority voting among neighborhood documents is used to decide the category for \( d_i \).

Based on the association mining methods in Section 5, we can mine the dependence patterns of words, denoted as \( \{c_1, c_2, \ldots, c_m\} \). We then extend the classical VSM vector \( [y_1, \ldots, y_n]^T \) to \( [y_1, \ldots, y_n, z_1, z_2, \ldots, z_m]^T \), where \( z_i \) indicates the TF-IDF weight of the dependence pattern \( c_i \). Then we apply the extended vector space model in text classification to assess the usefulness of the word dependence patterns and effectiveness of our methods. The following methods are used for comparisons: “NP” simply uses all noun phrases; “Apr” uses the word dependence patterns mined by the Apriori algorithm. “PPD-CS”, “PPD-IG”, “TPD” are implemented according to Section 5. The classical VSM without dependence patterns is the baseline. Note that the TPD utilizes the second choice for pre-selecting candidate patterns in Section 5.2.
**Data Collection:** We use Reuters-21578 and 20 News Groups collections. All documents are pre-processed with Porter Stemming and stop word removed. Then each collection is split into training set and test set. For the Reuters-21578, we adopt the standard “ModApte” split with 9603 documents in training set and 3299 in test set. For the 20 News Groups, we use the “bydate” version with 9840 documents in training set and 6871 in test set. There are 131 classes (classes that contain nonzero positive documents) in Reuters-21578 and 20 classes in 20 News Groups.

**Evaluation Metric:** We use the averaged micro F1 and macro F1 measures as performance metrics. In text classification, assuming the predicted precision for a topic is $p$ and the recall is $r$, the F1 measure (also known as harmonic mean of precision and recall) takes both precision and recall into consideration and evaluates the overall effectiveness of the classifier as follows: $F_1(p, r) = \frac{2pr}{p+r}$.

The micro-averaged F1 is calculated by averaging the F1 scores over all documents regardless of the topics (for each document, either $p = r = 1$ or $p = r = 0$). The macro-averaged F1 is calculated by averaging the F1 scores over all topics.

**Experimental Setup:** The $k$-NN algorithm is used as classifier, where the parameter $k$ is trained to optimize the evaluation metrics. The parameters involved in the comparative methods are selected experimentally. A sliding window with width $W = 15$ is used to count the word co-occurrence frequencies (We have tried $W = \{10, 15, 20, 50\}$, and $W = 15$ generally works best). For Apr, the minimum support value is set to be 0.05 for 20 News Groups and 0.06 for Reuters-21578. For PPD-CS, PPD-IG and TPD, the significance level used in null hypothesis test needs to be set. We systematically tried significant level $= \{0.01, 0.02, \ldots, 0.05\}$, the results show no significant differences. Thus we simply adopt the significant level 0.01 for PPD-CS, and 0.02 for PPD-IG and TPD. For the two IG-based methods (PPD-IG and TPD), in order to clarify where the gains come from, we test different sets of pure patterns with respect to different maximum orders. Then we report the performance changes as we gradually increase the maximum order of patterns used to expand document representation ($MaxOrder$ from 2 up to 4). Note that for NP and Apr, we only report the best performance among different settings of $MaxOrder$ (from 2 up to 4).

**Results and Analysis:** We list the performance measured by micro-averaged and macro-averaged F1 in Table II. It is clear that all the extended VSM methods outperform the baseline. Furthermore, the methods based on random variable dependence test techniques (PPD-CS, PPD-IG, and TPD) outperform the Apr, where the high-order patterns are selected mainly based on the co-occurrence frequency. This observation is not surprising. As described in Section 1, although the co-occurrence of a high-order pattern (two or more words) could be explained by the contextual association among them in some aspects, it may reduce to the random coincidence of lower ordered patterns. However this random coincidence can be greatly reduced by means of statistical multivariate dependence analysis techniques, such as Chi-squared test and IG-based methods. For PPD-CS and PPD-IG, both of them use the same algorithmic framework proposed in Algorithm 1, except the pairwise random variable dependence test methods. From Table II, we can see that PPD-CS and PPD-IG have similar performances. For the two IG-based methods, TPD is generally more stable and works better than PPD-IG.

### 6.2. Ad hoc Retrieval: MRF Retrieval Model with Pure High-order Dependence

Ad hoc retrieval aims at finding relevant documents from a document collection with respect to a given query. The query is usually a textual description (e.g. several keywords) of the user’s information need. Note that the extended VSM model can hardly be used in the ad hoc retrieval directly. This is because expanding the vector repre-
Table II: Micro and Macro F1 Measure for $W = 15$

<table>
<thead>
<tr>
<th>Methods</th>
<th>Micro F1</th>
<th>Macro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max=2</td>
<td>Max=3</td>
</tr>
<tr>
<td>NP</td>
<td>76.22</td>
<td>76.41</td>
</tr>
<tr>
<td>Apr</td>
<td>75.99</td>
<td>75.89</td>
</tr>
<tr>
<td>PPD-CS</td>
<td>79.02</td>
<td>79.21</td>
</tr>
<tr>
<td>PPD-IG</td>
<td>79.02</td>
<td>79.12</td>
</tr>
<tr>
<td>TPD</td>
<td>79.02</td>
<td>79.52</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Micro F1</th>
<th>Macro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max=2</td>
<td>Max=3</td>
</tr>
<tr>
<td>NP</td>
<td>55.39</td>
<td>55.16</td>
</tr>
<tr>
<td>Apr</td>
<td>56.75</td>
<td>57.05</td>
</tr>
<tr>
<td>PPD-CS</td>
<td>58.04</td>
<td>58.10</td>
</tr>
<tr>
<td>PPD-IG</td>
<td>58.03</td>
<td>58.21</td>
</tr>
<tr>
<td>TPD</td>
<td>58.03</td>
<td>58.30</td>
</tr>
</tbody>
</table>

Note the statistical significant improvements (student's t test) over VSM, NP and Apr are marked with $\alpha$, $\beta$ and $\gamma$ respectively. Only the cases (bolded ones) for PPD-CS, PPD-IG and TPD with the best performance are tested.

sentation of documents with high-order dependence patterns mined from a large scale collection is computationally intractable. In this section, we adopt a unified framework based on the Markov Random Field model (MRF) to incorporate high-order dependencies related to the query into the retrieval model, and use it as a testbed to investigate the usefulness of pure high-order dependencies.

In classical bag-of-word based IR models, such as the binary independence model (BIM) [Robertson and Spärck-Jones 1976] and vector space model (VSM) [Salton et al. 1975], both queries and documents are represented as a set of terms that are assumed to be statistically independent. Recently, language modeling approach [Ponte and Croft 1998] to IR has been proposed to estimate a language model for each document and model the relevance of documents according to their capability of generating the given query. The simplest form of language model is unigram language model (UG) [Song and Croft 1999] which does not consider any dependence between words. There have been several attempts to incorporate the short (adjacent) range term dependencies, such as the bigram language model (BG) [Song and Croft 1999] that captures the dependencies between two adjacent words. Later, [Metzler and Croft 2005] proposed a general framework to incorporate different kinds of query term dependencies in documents via the MRF model that is able to utilize arbitrary text features (e.g. term proximity, sequential or statistical cooccurrence) as evidence in the estimation of document query likelihood. [Bendersky et al. 2010] proposed a weighting scheme for the latent concepts used in MRF, where the importance of a concept is empirically determined by the collection and some external data (such as google n-grams, MSN query log, Wiki titles). [Shi and Nie 2010] proposed a framework to incorporate different features to learn the concept importance based on the data collection and external sources such as term pairs occuring in Termium, Wiki titles and so on. In this paper, we are mainly concerned about testing the effect of high-order term dependencies on the MRF-based IR model, and optimal weight learning is not our focus.

[Bendersky and Croft 2008] considered using noun-phrases to discover key concepts in verbose query, where the noun-phrases are extracted from queries and classified into key and non-key concepts based on pre-labeled training set. [Kumaran and Carvalho
2009] and [Xue et al. 2010] proposed to reduce long queries to more effective shorter ones (subsets of original query) by interactively removing extraneous terms. Unlike the aforementioned papers, this section is focused on testing the general, fully automatic and unsupervised statistical methods for extracting high-order word associations, i.e. Apr, PPD-CS, PPD-IG and TPD. We use the MRF framework as testbed to analyze the impact and effectiveness of incorporating different types of pure dependencies among query terms into the retrieval process. We also incorporate the Noun Phrase patterns with syntactical dependencies in the MRF framework for comparison with the above statistical methods.

6.2.1. Introduction to MRF model in IR. Let the original query be \( Q = \{q_1, q_2, \ldots, q_m\} \), where each \( q_i \) represents a query term. The task of IR is to estimate the likelihood that certain document \( D \) is relevant with respect to the given query \( Q \), i.e. \( P(D|Q) \). For ranking purpose, the document query likelihood \( P(D|Q) \) is reformulated as follows:

\[
P(D|Q) = \frac{P(Q,D)}{P(Q)}
\]

where the rank equivalence holds since the \( P(Q) \) is a constant in one query session.

The MRF model [Metzler and Croft 2005] gives us a way to approximate the joint distribution \( P(Q,D) \), detailed as follows.

The Markov Random Field is defined as a set of random variables having a Markov property described by an undirected graph \( G \). One interpretation of the Markov property is that a random variable in the graph is only dependent on its neighbors observed.

In the context of representing \( P(Q,D) \), we denote the undirected graph as \( G = (\nu, \varepsilon) \), where \( \nu = Q \cup D \) is the node set and \( \varepsilon \) is the set of edges. The query nodes \( q_i \) and document node \( D \) are discrete random variables, with their values chosen from the universal vocabulary and the document collection respectively. The edges represent the associations between the nodes. The MRF graph for \( P(Q,D) \) is illustrated in Figure 2a. Given a MRF graph \( G \), the probability \( P(Q,D) \) can be represented by the product of potential functions \( \psi_c \) over all cliques \( c \) in the graph \( G \). In Figure 2a, the node set \( c = \{q_1, q_2, D\} \) forms a clique since all nodes in \( c \) are connected with each other. The value of the potential function \( \psi_c(q_1, q_2, D) \) associated with a particular assignment \( (q_1 = w_1, q_2 = w_2, D = d_1) \) denotes the affinity between these values: the higher the value \( \psi_c(w_1, w_2, d_1) \), the more compatible these values are. Consider the document \( d_1 \) on data mining, we would expect \( \psi_c(data, mining, d_1) > \psi_c(data, structure, d_1) \) as the terms data and mining are much more topically related to \( d_1 \) compared with terms data and structure.

For convenience, the potential functions are expressed as exponentials \( \psi_c(c) = \exp(\lambda_c f_c(c)) \), where \( c \) is a clique and \( f_c(c) \) is called the feature function of \( c \). The weights \( \lambda_c \) are parameters that need to be tuned to optimize the goal (e.g. retrieval precision).
In this section, we propose to apply the pure high-order dependence into MRF retrieval model [Metzler and Croft 2005] based on two observations.

First, the MRF retrieval model incorporates all combinations of query terms in the estimation of \( P(Q,D) \) (see Formula (14)). However, not all the term patterns are meaningful contextual associations, especially for long queries. For example, consider the

\[
P(Q,D) = \frac{1}{Z} \exp \left\{ \sum_c \lambda_c f_c(c) \right\} \tag{13}\]

where \( Z \) is the normalization factor. As we can see in (13), the joint distribution \( P(Q,D) \) is represented as the normalized exponential of the weighted summation of feature functions over all cliques.

In practice, we can approximate the joint \( P(Q,D) \) as follows (referred to as MRF retrieval model):

\[
P_{MRF}(Q,D) \propto \exp[\lambda_{T_D} \sum_{c \in T_D} f_{T_D}(c) + \lambda_{O_D} \sum_{c \in O_D} f_{O_D}(c) + \lambda_{U_D} \sum_{c \in U_D} f_{U_D}(c)] \tag{14}\]

where three kinds of cliques are incorporated and \( \lambda_{T_D}, \lambda_{O_D}, \lambda_{U_D} \) are the weighting parameters. Note that same kind of cliques share the same weighting parameter. The description of three kinds of cliques and their corresponding feature functions are listed in Table III and IV.

### 6.2.2. MRF Model with Pure High-order Dependence

In this section, we propose to apply the pure high-order dependence into MRF retrieval model [Metzler and Croft 2005] based on two observations.

First, the MRF retrieval model incorporates all combinations of query terms in the estimation of \( P(Q,D) \) (see Formula (14)). However, not all the term patterns are meaningful contextual associations, especially for long queries. For example, consider the

\[
P(Q,D) = \frac{1}{Z} \exp \left\{ \sum_c \lambda_c f_c(c) \right\} \tag{13}\]

where \( Z \) is the normalization factor. As we can see in (13), the joint distribution \( P(Q,D) \) is represented as the normalized exponential of the weighted summation of feature functions over all cliques.

In practice, we can approximate the joint \( P(Q,D) \) as follows (referred to as MRF retrieval model):

\[
P_{MRF}(Q,D) \propto \exp[\lambda_{T_D} \sum_{c \in T_D} f_{T_D}(c) + \lambda_{O_D} \sum_{c \in O_D} f_{O_D}(c) + \lambda_{U_D} \sum_{c \in U_D} f_{U_D}(c)] \tag{14}\]

where three kinds of cliques are incorporated and \( \lambda_{T_D}, \lambda_{O_D}, \lambda_{U_D} \) are the weighing parameters. Note that same kind of cliques share the same weighing parameter. The description of three kinds of cliques and their corresponding feature functions are listed in Table III and IV.

### Table III: Description of Clique Sets

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_D )</td>
<td>set of cliques formed by the document node and one query term</td>
</tr>
<tr>
<td>( O_D )</td>
<td>set of cliques formed by the document node and two or more terms that occur in sequential order within the query</td>
</tr>
<tr>
<td>( U_D )</td>
<td>set of cliques formed by the document node and two or more terms that appear in any order within the query</td>
</tr>
<tr>
<td>( E_D )</td>
<td>cliques formed by the document node and expansion concept</td>
</tr>
<tr>
<td>( E_Q )</td>
<td>set of cliques formed by the expansion concept and one or more query terms</td>
</tr>
<tr>
<td>( E )</td>
<td>cliques formed by the single expansion concept</td>
</tr>
</tbody>
</table>

### Table IV: Feature Functions of Cliques

<table>
<thead>
<tr>
<th>Name</th>
<th>Feature Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{T_D}(q_i,D) = \log[p(q_i</td>
<td>D)] ), where ( p(q_i</td>
</tr>
<tr>
<td>( f_{O_D}(q_i,q_{i+1},...,q_{i+k},D) = \log[p(q_i,q_{i+1},...,q_{i+k}</td>
<td>D)] ), where ( q_i,q_{i+1},...,q_{i+k} ) is query term sequence and ( p(q_i,q_{i+1},...,q_{i+k}</td>
</tr>
<tr>
<td>( f_{U_D}(q_i,...,q_{i+j},D) = \log[p(q_i,...,q_{i+j}</td>
<td>D)] ), where ( q_i,...,q_{i+j} ) is term combination and ( p(q_i,...,q_{i+j}</td>
</tr>
<tr>
<td>( f_{E_D}(E,D) = \log[p(E</td>
<td>D)] ), where ( p(E</td>
</tr>
<tr>
<td>( f_{E_Q}(E,q_1,...,q_j) = \log[p_\pi(LLRT(E,q_1,...,q_j))] ); For LCE-PPD-IG ( f_{E_Q}(E,q_1,...,q_j) = \log[p_\pi(LLRT(E,q_1))</td>
<td>\sum_{q_i \in {q_1,...,q_j}} \log[p_\pi(LLRT(E,q_i))] ] for LCE-PPD-CS ( f_{E_Q}(E,q_1,...,q_j) = \log[p_\pi(\chi^2(E,q_1))</td>
</tr>
<tr>
<td>( f_E(E) = idf(E) ), where ( idf(E) ) is ( E )'s inverse document frequency.</td>
<td></td>
</tr>
</tbody>
</table>
Table V: Results for Ad hoc Retrieval (title query)

<table>
<thead>
<tr>
<th>Methods</th>
<th>WSJ87-92 Max=2</th>
<th>Max=3</th>
<th>Max=4</th>
<th>AP88-89 Max=2</th>
<th>Max=3</th>
<th>Max=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline: UG=32.98</td>
<td></td>
<td></td>
<td></td>
<td>Baseline: UG=32.36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MRF-SD</td>
<td>34.82</td>
<td>34.50</td>
<td>34.86</td>
<td>33.04</td>
<td>32.99</td>
<td>32.99</td>
</tr>
<tr>
<td>MRF-FD</td>
<td>35.09</td>
<td>34.72</td>
<td>34.60</td>
<td>34.09</td>
<td>33.93</td>
<td>33.89</td>
</tr>
<tr>
<td>MRF-NP</td>
<td>34.88</td>
<td>34.59</td>
<td>34.54</td>
<td>33.66</td>
<td>32.91</td>
<td>32.82</td>
</tr>
<tr>
<td>MRF-Apr</td>
<td>34.81</td>
<td>34.33</td>
<td>34.20</td>
<td>34.00</td>
<td>33.99</td>
<td>33.76</td>
</tr>
<tr>
<td>MRF-PPD-CS</td>
<td>35.10</td>
<td>34.72</td>
<td>34.66</td>
<td>34.11</td>
<td>34.02</td>
<td>33.89</td>
</tr>
<tr>
<td>MRF-PPD-IG</td>
<td>35.10</td>
<td>34.97</td>
<td>34.72</td>
<td>33.95</td>
<td>34.10</td>
<td>33.89</td>
</tr>
<tr>
<td>MRF-TPD</td>
<td>35.10</td>
<td>35.37</td>
<td>35.36</td>
<td>33.95</td>
<td>34.18</td>
<td>33.89</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>ROBUST Max=2</th>
<th>Max=3</th>
<th>Max=4</th>
<th>WT10G Max=2</th>
<th>Max=3</th>
<th>Max=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline: UG=18.10</td>
<td></td>
<td></td>
<td></td>
<td>Baseline: UG=13.51</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MRF-SD</td>
<td>18.87</td>
<td>19.24</td>
<td>18.50</td>
<td>14.36</td>
<td>14.40</td>
<td>14.49</td>
</tr>
<tr>
<td>MRF-PPD-CS</td>
<td>18.44</td>
<td>20.00</td>
<td>19.76</td>
<td>14.67</td>
<td>14.59</td>
<td>14.73</td>
</tr>
</tbody>
</table>

Note the statistical significant improvements over UG, both MRF-SD and MRF-FD and both MRF-NP and MRF-Apr are marked with $\alpha$, $\beta$ and $\gamma$ respectively.

query “Controlling the Transfer of High Technology” (TREC 4 topic 100), the term patterns, such as (“control”, “high”) and (“transfer”, “high”), do not seem to be meaningful associations. Our IG-based methods (PPD-IG and TPD) could help to single out those term combinations that do not have pure dependencies.

Second, for each query term combination, we need to extract corresponding feature functions (i.e. $f_c(c)$) from the corpus which is usually time consuming especially for large corpus. Our IG-based methods could filter out redundant term patterns and reduce the computational burden.

Based on above observations, we extract the cliques in $O_D$ and $U_D$ only from those associations that pure high-order dependence term patterns. For example, $\{q_1, q_2, D\}$ is selected if $\{q_1, q_2\}$ is a pure dependence association by our IG-based methods (PPD-IG and TPD). The refined $O_D$ and $U_D$ are denoted as $\tilde{O}_D$ and $\tilde{U}_D$ respectively. Then the MRF retrieval model with pure high-order dependence is reformulated as follows:

$$P_{MRF-IG}(Q, D) \propto \exp\{\lambda_T \sum_{c \in T_D} f_T(c) + \lambda_O \sum_{c \in O_D} f_O(c) + \lambda_U \sum_{c \in U_D} f_U(c)\}$$ (15)

6.2.3. Experimental Setup. The aim of this experiment is to investigate the usefulness of pure dependencies in ad hoc retrieval using the MRF model as testbed. We implement our IG-based MRF retrieval model with two kinds of pure dependence mining methods, namely PPD-IG, TPD, as defined in Section 5, and the corresponding retrieval models are denoted as MRF-PPD-IG, MRF-TPD. As a comparison, we implement the MRF retrieval model with the term dependencies mined by Apr, denoted as MRF-Apr, and mined by Chi-squared method, denoted as MRF-PPD-CS. We also compare with other ad hoc retrieval models that incorporate term dependencies: the MRF re-

---

4TREC, short for Text REtrieval Conference, is an annual information retrieval conference and competition focusing on different information retrieval (IR) research tasks. For each task, they provide the test data collections, test topics and user relevance judgements.
Table VI: Results for Ad hoc Retrieval (desc query)

<table>
<thead>
<tr>
<th>Methods</th>
<th>WSJ87-92 Max=2</th>
<th>WSJ87-92 Max=3</th>
<th>WSJ87-92 Max=4</th>
<th>AP88-89 Max=2</th>
<th>AP88-89 Max=3</th>
<th>AP88-89 Max=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline: UG=25.56</td>
<td>25.10</td>
<td>27.33</td>
<td>27.30</td>
<td>27.91</td>
<td>27.76</td>
<td>27.76</td>
</tr>
<tr>
<td>MRF-SD</td>
<td>25.91</td>
<td>25.95</td>
<td>25.45</td>
<td>27.10</td>
<td>27.39</td>
<td>27.27</td>
</tr>
<tr>
<td>MRF-FD</td>
<td>26.49</td>
<td>26.10</td>
<td>26.15</td>
<td>27.91</td>
<td>27.25</td>
<td>27.12</td>
</tr>
<tr>
<td>MRF-NP</td>
<td>26.24</td>
<td>26.67</td>
<td>26.37</td>
<td>27.78</td>
<td>28.21</td>
<td>27.60</td>
</tr>
<tr>
<td>MRF-Apr</td>
<td>26.02</td>
<td>25.99</td>
<td>25.73</td>
<td>27.87</td>
<td>28.25</td>
<td>27.27</td>
</tr>
<tr>
<td>MRF-PPD-CS</td>
<td>26.29</td>
<td>26.43*</td>
<td>26.15</td>
<td>27.91*</td>
<td>27.39</td>
<td>27.27</td>
</tr>
<tr>
<td>MRF-PPD-IG</td>
<td>26.30</td>
<td>26.64*</td>
<td>26.13</td>
<td>27.91*</td>
<td>27.99*</td>
<td>27.89</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>ROBUST Max=2</th>
<th>ROBUST Max=3</th>
<th>ROBUST Max=4</th>
<th>WT10G Max=2</th>
<th>WT10G Max=3</th>
<th>WT10G Max=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline: UG=18.46</td>
<td>15.00</td>
<td>14.89</td>
<td>14.70</td>
<td>14.72</td>
<td>15.12</td>
<td>14.95</td>
</tr>
<tr>
<td>MRF-SD</td>
<td>19.20</td>
<td>19.14</td>
<td>19.14</td>
<td>15.00</td>
<td>14.89</td>
<td>14.70</td>
</tr>
<tr>
<td>MRF-FD</td>
<td>19.47</td>
<td>19.30</td>
<td>19.30</td>
<td>15.11</td>
<td>15.23</td>
<td>15.16</td>
</tr>
<tr>
<td>MRF-NP</td>
<td>19.03</td>
<td>19.15</td>
<td>19.34</td>
<td>14.73</td>
<td>15.01</td>
<td>14.93</td>
</tr>
<tr>
<td>MRF-PPD-CS</td>
<td>19.20</td>
<td>19.61*</td>
<td>19.46</td>
<td>14.88</td>
<td>15.34*</td>
<td>15.10</td>
</tr>
<tr>
<td>MRF-TPD</td>
<td>19.22</td>
<td>19.99*</td>
<td>19.78</td>
<td>14.72</td>
<td>15.69</td>
<td>15.78*</td>
</tr>
</tbody>
</table>

Note the statistical significant improvements over UG, both MRF-SD and MRF-FD and both MRF-NP and MRF-Apr are marked with α, β and γ respectively.

In information retrieval model with sequential dependence (MRF-SD) and MRF with full dependence (MRF-FD). Unlike the MRF-SD and MRF-FD proposed in [Metzler and Croft 2005], the orders of term dependencies utilized by the MRF-SD and MRF-FD are constrained to be no more than certain MaxOrder. Note that the sequential dependence between adjacent terms (i.e. bigrams and n-grams) are already embedded in MRF-SD. We also incorporate the noun phrases into the MRF model, denoted as MRF-NP, where the noun phrases are extracted from the query. The Unigram language model (UG) based on full independence setting is used as the baseline. In order to clarify which of the high-order dependencies (for n-order patterns \( n = 2, 3, 4 \)) give the most contributions to the performance improvement, we report the retrieval performances when different ordered patterns are used in MRF model respectively. More specifically, we report the performances as we gradually increase the max order of patterns used in MRF model (MaxOrder from 2 up to 4).

**Data Collection:** We evaluate all the aforementioned models on TREC collections: AP8889, WSJ8792, ROBUST2004 and WT10G. Lemur 4.12 is used for indexing. The collection is pre-processed by removing stop words and applying the Porter stemmer. For WSJ8792 and AP8889, the title field and desc field of topics 151-200 are used. For ROBUST2004, the title field and desc field of TREC topics 601-700 are used. For WT10G, the title field and desc field of TREC topics 501-550 are used.

**Evaluation Metric:** Mean Average Precision (MAP) is used as evaluation metric, which is the mean of average precision scores over all the queries. For each query, suppose the retrieved document rank list is \( \{d_1, d_2, ..., d_M\} \), the average precision is calculated as follows:

\[
AveragePrecision = \frac{\sum_{j=1}^{M} [P(j) \times \delta_r(j)]}{\sum_{i=1}^{M} [\delta_r(i)]}
\]  

5Lemur is an open source project that develops search engines and text analysis tools for research and development of information retrieval and text mining softwares
where $P(j)$ is the percentage of relevant documents among the top $j$ documents. $\delta_r(j)$ is a function that equals to one if the document $d_j$ is relevant to the given query and zero otherwise.

**Parameter Estimation:** In the estimation of language models, the Dirichlet smoothing parameter $\mu$ [Lv and Zhai 2009] is set to 1000. For MRF-SD and MRF-FD, the weighting parameters $\lambda_{TD}, \lambda_{OD}, \lambda_{UD}$ are set based on the optimal ones reported in [Metzler and Croft 2005]. For MRF-SD, the weighting parameters are \{\lambda_{TD} = 0.85, \lambda_{OD} = 0.1, \lambda_{UD} = 0.05\}. For MRF-FD, \{\lambda_{TD} = 0.8, \lambda_{OD} = 0.1, \lambda_{UD} = 0.1\} are used. For MRF-NP, MRF-Apr, MRF-PPD-CS, MRF-PPD-IG, MRF-TPD, the weighting parameters are set to be the same as MRF-FD. For PPD-CS, PPD-IG and TPD, the significant level is set to be the same as in the text categorization experiment (Section 6.1): 0.01 for PPD-CS, 0.02 for PPD-IG and TPD.

6.2.4. **Results and Analysis.** From Tables V and VI, we can see that all the retrieval models utilizing term dependencies (MRF-SD, MRF-FD, MRF-NP, MRF-Apr, MRF-PPD-CS, MRF-PPD-IG, MRF-TPD) consistently outperform the baseline (UG). All the MRF-FD-based retrieval models outperform MRF-SD. These observations show the usefulness of term dependencies in ad hoc retrieval.

Comparing all MRF-based models, MRF with TPD patterns works best over all topics. For the title queries, although the TPD-based method outperforms other models, the improvement is insignificant. One possible reason is that the title queries are relatively short (about 1-6 words) and there is little space for our IG-based methods to further refine the term dependencies used in MRF model. For the longer desc queries (about 10-20 words), the TPD-based methods work best among all MRF-based models. The significant improvement affirms the intuition that our TPD-based MRF model could favor long queries. The improvements also show the effect of introducing pure dependence into the MRF model.

The TPD-based retrieval model generally achieves better performance than the ones based on PPD, especially in the case of desc queries. There are two main reasons: First, based on Proposition 4.3, TPD is the most strict approach since it entails both UPD and CPD, while the PPD only entails UPD. This means TPD generates more salient and succinct contextual associations than PPD patterns; Second, in practice, we need to perform only one LLRT to decide a TPD dependence, while $C^2_n$ times LLRT is needed for identifying PPD dependencies. Thus, the PPD mining algorithm suffers from more estimation complexities.

Regarding the different orders of dependencies (the order of patterns from 2 up to 4), most methods have best performance at $MaxOrder = 3$. That means patterns with order $n \in \{2, 3\}$ are the main contributors to the improvement over the Unigram model. One reason why higher order patterns (for order $n$ greater than 3) did not bring benefit to the retrieval precision may be due to the fact that we utilize a simple frequency-based feature weighting function in the MRF setting. Consequently, the features selected by our PPD and TPD algorithms tend to be under-weighted and the profits of introducing higher-order patterns are limited under the MRF framework.

Another observation is that there is limited improvement after introducing the pure high-order dependence (e.g., MRF-TPD over MRF-Apr). The reason is that in this experimental setting we extract high-order dependencies from the whole collection, which could cause noisy and redundant information. Our conjecture is that the effect of pure dependence would be more significant in the framework of pseudo relevance feedback, where we can utilize the top retrieved documents as source for pure high-order dependence mining and those term patterns can be applied in the query expansion task. This will be verified in the next section.
6.3. Query Expansion: Latent Concept Expansion with Pure High-order Dependence

In relevance or pseudo-relevance feedback\(^6\), query expansion aims at refining the original query with other topically related concepts (single or multiple words) to produce a better representation of the underlying information need.

Typical query expansion methods are based on the assumption that all terms are independent of each other, for example, the Rocchio algorithm [Rocchio 1971] used in vector space model, the model-based feedback [Zhai and Lafferty 2001] and relevance model (RM) [Lavrenko and Croft 2001] used in language modelling framework.

As described in Section 6.2.1, the MRF retrieval model incorporate the word dependencies among query terms. Based on the MRF framework, Latent Concept Expansion (LCE) [Metzler and Croft 2007] was proposed for modeling term dependencies in query expansion.

The expanded concepts (single or multiple words) in LCE are chosen from the top ranked documents in the initial retrieval results, and weighted independently of the original query terms (Figure 2b). This assumption may lead to the risk of topic drift, especially with long documents. In order to tackle the problem, [Lang et al. 2010] proposed a Hierarchical MRF model based on LCE to incorporate the dependencies between expansion terms and original query indirectly through documents.

In this section, we propose to incorporate the pure dependence patterns as features into the estimation of expansion term likelihoods based on LCE. More specifically, we choose those expanded concepts that have a pure high-order dependence with a sufficient number of query terms (above a threshold) for query expansion. Instead of assuming that the expanded terms are independent of the original query in as in LCE (Figure 2b), we consider the direct high-order dependencies among the expansion terms and original query (Figure 2c).

6.3.1. Model Formulation. Let the original query be \( Q = \{q_1, q_2, ..., q_m\} \), the concept to be expanded be \( E \). The set of feedback documents are used as evidence for user’s information needs, denoted as \( \mathcal{R} \). Now our task is to estimate \( P(E|Q, \mathcal{R}) \), a probability distribution over expansion concepts, which is usually approximated by summing over all documents \( D \in \mathcal{R} \) as follows:

\[
P(E|Q, \mathcal{R}) \approx \frac{1}{Z} \sum_{D \in \mathcal{R}} P(E, Q, D)
\]

where \( Z \) is the normalization factor. For a query, \( Z \) is a constant with respect to all expansion concepts. To build an expanded query, the \( k \) terms with highest estimated expansion term likelihoods are selected.

Then we need to estimate the expansion likelihood for each document \( D \), w.r.t \( P(E, Q, D) \) in the MRF framework. As illustrated in Figure 2c, we formulate \( P(E,Q,D) \)

---

\(^6\)Relevance feedback in IR is a technique to use the relevance information of a set of feedback documents to improve the performance of the second round retrieval (usually through query expansion). Pseudo-relevance feedback assumes top \( k \) ranked feedback documents are relevant.
as follows:

\[
P(E, Q, D) \propto \exp\{F_D(Q) + F_D(E) + F_Q(E) + F_C(E)\}
\]

\[
\approx \exp\{\lambda_D \sum_{c \in T_D} f_{T_D}(c) + \lambda_D \sum_{c \in O_D} f_{O_D}(c) + \lambda_D \sum_{c \in U_D} f_{U_D}(c)\}
\]

\[
+ \lambda_D f_{E_D}(E, D) + \lambda_D f_{E}(E) + \lambda_D \sum_{c \in E_Q} f_{E_Q}(c)\}
\]

(18)

where \(F_D(Q)\) is the sum of all feature functions for cliques formed by the original query \(Q\) and document node \(D\) (e.g. \(\{q_1, q_2, D\}\) in Figure 2c), which is exactly the document relevance score for the original query used in Formula (14); \(F_D(E)\) is feature function for the clique formed by the expansion concept \(E\) and document node \(D\) (i.e. \(\{E, D\}\)), which represents the document relevance score with respect to expansion concept \(E\); \(F_C(E)\) is the feature function for single expansion concept node \(E\) (i.e. \(\{E\}\)), for which we take into account the \(idf\) to differentiate rare and common expansion concepts. Note that the feature functions used in \(F_D(Q)\), \(F_D(E)\) and \(F_C(E)\) have been well studied previously in [Metzler and Croft 2005][Metzler and Croft 2007]. We simply use similar feature functions, and the detailed experimental settings for those feature functions are listed in Table III and IV. \(F_Q(E)\) is the sum of all feature functions for cliques formed by the expansion concept \(E\) together with one or more query terms (e.g. \(\{E, q_1, q_2\}\) in Figure 2c).

Now we focus on how to define the proper feature functions in \(F_Q(E)\). For cliques containing the expansion concept and one or more query terms, the feature function should measure the compatibility among them, i.e., the degree of contextual association between the expansion concept and original query. One simple choice for the feature function is the logarithmic of co-occurrence frequency. However, as described in Section 1, the higher frequency of co-occurrence does not necessarily indicate stronger affinity, nor does lower frequency indicate weaker compatibility. In this paper, we naturally set the concept importance as the estimated probability of confidence for that the concept is a pure dependence pattern.

Therefore, the confidence level of the pure dependence, which eliminates the influences of random coincidence in the co-occurrence frequency, seems a suitable statistic for measuring the compatibility. Recall that the value of \(\theta - coordinate\) carries sufficient information needed for identifying a TPD. In the LLRT test, the confidence level \(\pi\) (Formula (11)) indicates to what degree we belief given a pattern is of pure dependence. We thus define the feature functions as follows:

\[
f_{E_Q}(E, q_i, \ldots, q_j) = \log[\pi_{LLRT}(E, q_i, \ldots, q_j)]
\]

(19)

where \(E_Q\) denotes the set of cliques containing the expansion concept \(E\) and one or more query terms \(q_i, \ldots, q_j\), and \(\pi_{LLRT}(E, q_i, \ldots, q_j)\) is the confidence level of the LLRT test for pattern \(\{E, q_i, \ldots, q_j\}\), which indicates to what degree that we belief this pattern is of pure dependence (Section 3.3).

Instead of using the TPD, the compatibility can also be estimated by the product of pairwise confidence levels between the expansion term and original query terms (e.g., \(\pi_{LLRT}(E, q_k)\)). Thus we have alternative feature functions based on PPD as follows:

\[
f_{E_Q}(E, q_i, \ldots, q_j) = \frac{1}{|\{q_i, \ldots, q_j\}|} \sum_{q_k \in \{q_i, \ldots, q_j\}} \log[\pi_{LLRT}(E, q_k)]
\]

(20)
Similarly, we also introduce the Chi-squared based feature function for $f_{E, Q}(E, q_i, ..., q_j)$. The feature function is as follows:

$$f_{E, Q}(E, q_i, ..., q_j) = \frac{1}{|\{q_i, ..., q_j\}|} \sum_{q_k \in \{q_i, ..., q_j\}} \log \left[ \pi_{\chi^2}(E, q_k) \right]$$

where $\pi_{\chi^2}(E, q_k)$ is the confidence level for Chi-squared test.

As we can see from Formula (18), we neglect the cliques formed by the original query $Q$, expansion concept $E$ and document $D$. We make the simplification because the high-order dependencies among the expansion term and original query are extracted from the whole set of feedback documents instead of any particular individual document.

6.3.2. Experimental Setup. In this experiment, we aim to investigate how the pure high-order dependence enhanced MRF models for query expansion performs in pseudo-relevance feedback. Note that for simplicity, in this experiment we focus on expanding with the single term concepts rather than the multi-term ones, although there is no such limitation in the model described above.

For the first round retrieval, we adopt the MRF retrieval model [Metzler and Croft 2005] described in Section 6.2.1. After the first round retrieval, the topmost $K$ documents are selected as pseudo-relevance feedback documents ($\mathcal{R}$). Then based on Formula 17 and 18, we can calculate the expansion term likelihoods. Then $k$ expansion concepts with the highest likelihood are selected, denoted as $E$. For the second round retrieval, the ranking function in LCE model [Metzler and Croft 2007] is used, shown as follows:

$$P_{LCE}(Q, E, D) \propto \exp \left\{ \lambda_{TD} \sum_{c \in TD} f_{TD}(c) + \lambda_{ED} \sum_{c \in ED} f_{ED}(c) + \lambda_{OD} \sum_{c \in OD} f_{OD}(c) + \lambda_{UD} \sum_{c \in UD} f_{UD}(c) \right\}$$

where $E_D$ is the set of cliques formed by the document node and one expansion concept $E \in \mathcal{E}$. The corresponding feature function is listed in Table IV.

We evaluate and compare the performances of the following five models: Unigram language model (UG), MRF retrieval model [Metzler and Croft 2005], Relevance Model (RM)- a widely recognized query language modeling approach, LCE [Metzler and Croft 2007], LCE with NP (LCE-NP), LCE with PPD-CS (LCE-PPD-CS), LCE with PPD-IG (LCE-PPD-IG) and LCE with TPD (LCE-TPD). In LCE-NP, the noun phrases are extracted from the feedback documents and the corresponding compatibility among words within noun phrase is calculated by the term frequency in the feedback documents. Note that Apr method is not included in this experiment, as it is already embedded in LCE. This is because in the implementation of LCE, the candidate expansion terms are pre-selected according to the co-occurrence of the expansion term and original query terms, which is equivalent to utilize Apr method to extract patterns with high frequency.

Data Collection: We evaluate our model using four TREC collections: AP8889, WSJ8792, ROBUST2004 and WT10G with selected TREC topics (Table VII). Lemur 4.12 is used for indexing. All collections are stopwords removed and Porter stemmed. The title fields of the TREC topics are used as queries.

Parameter Estimation: In the estimation of document language models, the Dirichlet smoothing parameter $\mu$ [Lv and Zhai 2009] is set to 1000. For MRF and LCE, the weighting parameters for feature functions are selected to optimize Mean Average Precision (MAP). For LCE-NP, LCE-PPD-CS, LCE-PPD-IG and LCE-TPD, the weighting parameters are set to the same as LCE. The number of feedback documents $K$ is set to 50, and the number of expanded terms $k$ is set to 50. MAP is used as evalu-
Table VII: Summary on TREC collections

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Documents</th>
<th>topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP8889</td>
<td>164,597</td>
<td>151-200</td>
</tr>
<tr>
<td>WSJ8792</td>
<td>173,252</td>
<td>151-200</td>
</tr>
<tr>
<td>ROBUST2004</td>
<td>528,155</td>
<td>601-700</td>
</tr>
<tr>
<td>WT10G</td>
<td>1,692,096</td>
<td>501-550</td>
</tr>
</tbody>
</table>

Table VIII: Summary of Query Expansion Results (Maxorder=2,3,4)

**Summary of Query Expansion Results (Maxorder = 2)**

<table>
<thead>
<tr>
<th>Methods</th>
<th>AP88-89</th>
<th>WSJ87-92</th>
<th>ROBUST2004</th>
<th>WT10G</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAP chg%</td>
<td>chg%</td>
<td>MAP chg%</td>
<td>chg%</td>
</tr>
<tr>
<td>UG</td>
<td>32.36</td>
<td>base –</td>
<td>32.98</td>
<td>base –</td>
</tr>
<tr>
<td>MRF</td>
<td>34.09</td>
<td>5.3 –</td>
<td>35.09</td>
<td>6.4 –</td>
</tr>
<tr>
<td>RM3</td>
<td>38.28</td>
<td>18.3 base</td>
<td>36.66</td>
<td>11.2 base</td>
</tr>
<tr>
<td>LCE</td>
<td>41.96</td>
<td>29.7 9.6</td>
<td>39.31</td>
<td>19.2 7.2</td>
</tr>
<tr>
<td>LCE-NP</td>
<td>41.64&lt;α&gt;β</td>
<td>28.7 8.8</td>
<td>38.48&lt;α&gt;β</td>
<td>16.7 5.0</td>
</tr>
<tr>
<td>LCE-PD-CS</td>
<td>41.57&lt;α&gt;β</td>
<td>28.5 8.5</td>
<td>39.00&lt;α&gt;β</td>
<td>18.3 6.4</td>
</tr>
<tr>
<td>LCE-PD-IG</td>
<td>42.18&lt;α&gt;β</td>
<td>30.3 11.2</td>
<td>39.29&lt;α&gt;β</td>
<td>19.1 7.1</td>
</tr>
<tr>
<td>LCE-TDP</td>
<td>42.18&lt;α&gt;β</td>
<td>30.3 11.2</td>
<td>39.29&lt;α&gt;β</td>
<td>19.1 7.1</td>
</tr>
</tbody>
</table>

**Summary of Query Expansion Results (Maxorder = 3)**

<table>
<thead>
<tr>
<th>Methods</th>
<th>AP88-89</th>
<th>WSJ87-92</th>
<th>ROBUST2004</th>
<th>WT10G</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAP chg%</td>
<td>chg%</td>
<td>MAP chg%</td>
<td>chg%</td>
</tr>
<tr>
<td>UG</td>
<td>32.36</td>
<td>base –</td>
<td>32.98</td>
<td>base –</td>
</tr>
<tr>
<td>MRF</td>
<td>34.24</td>
<td>5.8 –</td>
<td>34.97</td>
<td>6.0 –</td>
</tr>
<tr>
<td>RM3</td>
<td>38.28</td>
<td>18.3 base</td>
<td>36.66</td>
<td>11.2 base</td>
</tr>
<tr>
<td>LCE</td>
<td>41.81</td>
<td>29.2 9.2</td>
<td>38.46</td>
<td>16.6 4.9</td>
</tr>
<tr>
<td>LCE-NP</td>
<td>40.91&lt;α&gt;β</td>
<td>26.4 6.9</td>
<td>38.31&lt;α&gt;β</td>
<td>16.2 4.5</td>
</tr>
<tr>
<td>LCE-PD-CS</td>
<td>40.93&lt;α&gt;β</td>
<td>26.5 6.9</td>
<td>37.12&lt;α&gt;</td>
<td>12.5 1.3</td>
</tr>
<tr>
<td>LCE-PD-IG</td>
<td>42.57&lt;α&gt;βγ</td>
<td>31.5 11.2</td>
<td>39.86&lt;α&gt;βγ</td>
<td>20.8 8.7</td>
</tr>
<tr>
<td>LCE-TDP</td>
<td>42.41&lt;α&gt;βγ</td>
<td>31.0 10.8</td>
<td>40.13&lt;α&gt;βγ</td>
<td>21.7 9.5</td>
</tr>
</tbody>
</table>

**Summary of Query Expansion Results (Maxorder = 4)**

<table>
<thead>
<tr>
<th>Methods</th>
<th>AP88-89</th>
<th>WSJ87-92</th>
<th>ROBUST2004</th>
<th>WT10G</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAP chg%</td>
<td>chg%</td>
<td>MAP chg%</td>
<td>chg%</td>
</tr>
<tr>
<td>UG</td>
<td>32.36</td>
<td>base –</td>
<td>32.98</td>
<td>base –</td>
</tr>
<tr>
<td>MRF</td>
<td>33.89</td>
<td>4.7 –</td>
<td>34.59</td>
<td>4.9 –</td>
</tr>
<tr>
<td>RM3</td>
<td>38.28</td>
<td>18.3 base</td>
<td>36.66</td>
<td>11.2 base</td>
</tr>
<tr>
<td>LCE</td>
<td>41.30</td>
<td>27.4 7.7</td>
<td>38.32</td>
<td>16.2 4.5</td>
</tr>
<tr>
<td>LCE-NP</td>
<td>40.87&lt;α&gt;β</td>
<td>26.3 6.7</td>
<td>38.13&lt;α&gt;β</td>
<td>15.6 4.0</td>
</tr>
<tr>
<td>LCE-PD-CS</td>
<td>40.80&lt;α&gt;β</td>
<td>26.1 6.6</td>
<td>37.52&lt;α&gt;β</td>
<td>13.8 2.3</td>
</tr>
<tr>
<td>LCE-PD-IG</td>
<td>42.42&lt;α&gt;βγ</td>
<td>31.1 10.8</td>
<td>39.90&lt;α&gt;βγ</td>
<td>21.0 8.8</td>
</tr>
<tr>
<td>LCE-TDP</td>
<td>42.81&lt;α&gt;βγ</td>
<td>32.3 11.8</td>
<td>40.38&lt;α&gt;βγ</td>
<td>22.4 10.1</td>
</tr>
</tbody>
</table>

Note the statistical significant improvements over MRF, RM3 and LCE are marked with α, β and γ respectively.

6.3.3. Results and Analysis.

The results are given in Table VIII. We can see that MRF model, RM, LCE, LCE-NP, LCE-PD-CS, LCE-PD-IG and LCE-TPD all significantly outperform UG. The LCE, LCE-NP, LCE-PD-IG, LCE-TPD show significant improvements over RM on all data sets. LCE-PD-CS only shows significant improvement over RM on data collection AP8889, and slightly better or even worse for the other three collections.

Comparing the LCE-based approaches, LCE-TPD and LCE-PD-IG outperform the original LCE in all cases. The improvement indicates the benefit of adding the pure
dependence features into the estimation of expansion term likelihoods. For example, when querying for “Gene Therapy and Its Benefits to Humankind” (TREC topic 198), LCE-TPD can expand terms like names of geneticist (e.g. “Steven”, “Blease”) and some disease that can be treated by gene therapy (e.g. “melanoma”), while the co-occurrence statistics for them with respect to query terms is insignificant. Both LCE and RM fail to capture these concepts by using term occurrence (or co-occurrence for multi terms) features only.

Next we will compare LCE-based methods with different orders of dependencies. LCE generally gives best performance at MaxOrder = 3 on all collections except WT10G. The performances of both LCE-PPD-IG and LCE-TPD tend to increase when the MaxOrder increases though the differences of LCE-PPD-IG between MaxOrder = 3 and MaxOrder = 4 is not as significant as in LCE-TPD. These results confirm the discussion we had in Section 6.2.4: with more relevant resources with respect to given query, e.g. pseudo relevance feedback documents rather than the whole data collection, our PPD and TPD methods could offer more benefits to the practical tasks.

The LCE-PPD-CS generally has worse performance than LCE. The main difference between LCE-PPD-CS and LCE-PPD-IG is the statistical methods for the pairwise variable independence test. To understand the main reason why PPD-CS and PPD-IG perform differently in query expansion task, recall that, in previous studies in Section 5.5, 2-order LLRT is more accurate and robust than Chi-squared test with respect to data sparsity. In the previous experimental study on tasks of text categorization and ad hoc retrieval, the whole data collection is used to mine the high-order word associations, and hence PPD-CS and PPD-IG have similar performances. However, in the pseudo relevant feedback framework, only top K ranked documents in the first round retrieval are used to extract high-order patterns. In this case the term co-occurrences are much more sparse. Therefore, our IG-based methods are more robust in the query expansion task than other statistical independence test techniques, such as Chi-square test for independence.

Comparing two IG-based models (LCE-PPD-IG and LCE-TPD), they have the same performance when MaxOrder = 2 since 2-order PPD and 2-order TPD is indeed equivalent to each other. However, as Maxorder increases up to 4, LCE-TPD gradually outperforms LCE-PPD-IG for collection AP8889, WSJ8792, and WT10G. For ROBUST2004, the performance gaps between LCE-TPD and LCE-PPD-IG is relatively small. Hence the TPD method can be considered generally works the best among all comparative methods in query expansion task.

7. DISCUSSION

Several issues on the effectiveness and usability of the IG-based methods need to be further clarified. First, although TPD does not directly entail PPD, it can still be considered more strict than PPD in the sense it entails both UPD and CPD, as illustrated in Figure 1. Our experiments in Section 6 show that the TPD-based models can generally outperform PPD-based models in practical tasks. This observation seems to indicate the semantic importance of CPD. We think that CPD is more complete for extracting the high-order semantic entity, since compared with UPD it can eliminate the conditional random coincidences.

Second, the PPD mining algorithm suffers from more estimation complexities than TPD, because we only need to perform one LLRT test to decide a TPD dependence, while $C_n^4$ LLRT tests are needed for identifying a PPD dependence. This issue is also reflected by our experiments in Section 6 that the TPD-based models generally achieve better results than PPD.

Third, regarding time complexity, although the extraction of TPD patterns has a higher complexity than that of PPD patterns (based on the complexity analysis in Sec-
tion 5), the TPD patterns could be effectively extracted by pre-selecting some candidate patterns via a proper scheme, e.g., the mining algorithm of PPD.

Finally, as stated in the experimental studies, as a test for pairwise independence, 2-order LLRT seems to be more distinctive and robust than common statistical independence test methods such as Chi-squared test, especially in dealing with sparse data.

8. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed two types of pure high-order dependence: Unconditional Pure Dependence (UPD) and Conditional Pure Dependence (CPD). We analytically clarify a spectrum of high-order pure dependence, and propose a general framework based on Information Geometry to extract high-order pure dependence patterns of words from documents (PPD and TPD). In this IG-based framework, we have developed a set of rigorously-established justifications and feasible algorithms to single out high-order pure dependence by a well-founded statistical procedure (i.e. the log likelihood ratio test). We also integrate high-order pure dependence patterns into the VSM model for text classification, the MRF and LCE models for text retrieval (with and without query expansion). Evaluation results demonstrated the usefulness of the high-order pure dependence, and the effectiveness and robustness of our IG-based approach.

Our future work will be focused on addressing the following issues. First, we will further clarify semantic distinctions between PPD and TPD and expand the spectrum of pure high-order dependence. Second, it is possible to investigate the application of pure high-order dependence into more extensive information retrieval tasks and other research areas, such as image retrieval, object recognition, bioinformatics and so on. Finally, we will systematically develop novel high-order document representation model involving the pure high-order dependencies among terms.

APPENDIX

A.1. Proof of Proposition 3.1

PROPOSITION A.1.

\[ g_{dd} = \frac{1}{\sum_x 1/p(x)} \]  

(23)

PROOF. The Fisher information matrix \( G_\zeta \) is too complicated to be calculated directly. Instead, we obtain \( G_\zeta \) by the Jacobian matrix of coordinate transformation. First, the Fisher information of the \( \eta \)-coordinates \( G_\eta \) can be calculated. The Jacobian matrix of the coordinate transformation from \( \zeta \)-coordinates to \( \eta \)-coordinates is given by

\[ J = \begin{pmatrix}
\frac{\partial \eta_1 \zeta_1}{\partial \zeta_1} & \cdots & \frac{\partial \eta_d \zeta_1}{\partial \zeta_1} \\
\vdots & \ddots & \vdots \\
\frac{\partial \eta_1 \zeta_d}{\partial \zeta_d} & \cdots & \frac{\partial \eta_d \zeta_d}{\partial \zeta_d}
\end{pmatrix} \]  

(24)

i.e., only the elements on the main diagonal and the last row are non-zero since the only difference between \( \eta \)-coordinates and the mixed \( \zeta \)-coordinates is the last element. We can also get the inverse of \( J \), i.e., \( J^{-1} \). Note that there are \( n \) variables, and hence \( d = 2^n - 1 \).

Then the Fisher information matrix w.r.t. the mixed coordinates can be derived by

\[ G_\zeta = (J^{-1})^T G_\eta (J^{-1}). \]  

(25)

We can verify that the last row and the last column of the \( G_\zeta \) are all zeros (except for the bottom-right element \( g_{dd} \)). This observation indicates the orthogonality. In
addition, we have

\[ g_{dd} = \frac{h_{dd}}{j_{dd}^2}, \quad (26) \]

where \( h_{dd} \) denotes the bottom-right element of \( G_\eta \) and \( j_{dd} \) denotes the bottom-right element of \( J \).

By the definition of \( G_\eta \), we have

\[ h_{dd} = E \left[ \left( \frac{\partial \log p(x; \eta)}{\partial \eta_{12...n}} \right)^2 \right] = \sum_x \frac{\left( \partial_{\eta_{12...n}} p(x; \eta) \right)^2}{p(x; \eta)} = \sum_x \frac{1}{p(x)}. \quad (27) \]

This last identity in Formula (27) holds because we have Formula (28) when expanding \( p(x) \) by the inclusive-exclusive principle,

\[ \partial_{\eta_{12...n}} p(x; \eta) = (-1) \cdot C(x), \quad (28) \]

where \( C(x) \) is the number of 1’s in vector \( x \). For example, if \( n = 3 \), we have \( \partial_{\eta_{12}} p([1, 1, 0]^T) = \partial_{\eta_{12}} (\eta_1 - \eta_2 + \eta_3) = -1 \), and \( \partial_{\eta_{12}} p([1, 0, 0]^T) = \partial_{\eta_{12}} (\eta_1 - \eta_2 - \eta_3 + \eta_{12}) = 1 \). Considering the bottom-right element of matrix \( J \), we have

\[ j_{dd} = \partial_{\eta_{12...n}} \theta_{12...n} = \partial_{\eta_{12...n}} \sum_x (-1)^{n-C(x)} \log p(x) \]

\[ = \sum_x \frac{(-1)^{n-C(x)}}{p(x)} \partial_{\eta_{12...n}} p(x) = \sum_x \frac{1}{p(x)}. \quad (29) \]

This last identity in Formula (29) comes into existence because of Formula (28). At last, by Formula (27), (29) and (26), we get Formula (23). □

A.2. Proof of Proposition 4.3

PROPOSITION A.2. TPD ⇒ CPD

PROOF. We will prove \( \neg CPD \Rightarrow \neg TPD \). Let \( C \subseteq X \), \( a_C \) is a sub-assignment of \( a_X \) iff \( a_C \) assigns the same value to \( C \) as \( a_X \). We call an assignment (or sub-assignment) odd if it assigns odd number of 1’s to variables. Otherwise, it is an even assignment.

Let us consider the term inside the logarithmic function of \( \theta_{12...n} \), i.e., \( \prod_{k=0}^{n-1} \prod_{a_x \in A^{(k)}_X} p_{ax}^{(-1)^{n-k}} \). It is clear that, if \( n \) is odd, the numerator and denominator of this term can be rewritten as \( \prod_{a_x \text{ is odd}} p_{ax} \) and \( \prod_{a_x \text{ is even}} p_{ax} \), respectively. On the other hand, if \( n \) is even, the numerator and denominator will be interchanged.

If the joint distribution \( p(x) \) can be conditionally factorized, without loss of generality, assume that there exists a partition \( \{ C_0, C_1, C_2 \} \) of \( X \), such that \( p(c_1, c_2 | c_0) = p(c_1 | c_0) \cdot p(c_2 | c_0) \). Let \( V = C_1 \cup C_2 \). Then, given an arbitrary assignment \( a_X \), we have

\[ p_{av|ac} = p_{ac_1|ac} \cdot p_{ac_2|ac} \]

i.e., \( p_{ax} = p_{ac_0|c_1} \cdot p_{ac_0|c_2} / p_{ac_0} \). Substituting this relation into Formula (4), it is easy to check that the occurrence number of \( p_{ac_0} \) in the numerator is the same as that of \( p_{ac_0} \) in the denominator, since the number of odd (sub-) assignments is exactly the same as the number of even (sub-) assignments, and hence they can be eliminated with each other. In a similar manner, we can see that all occurrences of \( p_{ac_0|c_1} \) and \( p_{ac_0|c_2} \) can also be eliminated. It turns out that we have \( \prod_{k=0}^{n} \prod_{a_x \in A^{(k)}_X} p_{ax}^{(-1)^{n-k}} = 1 \), which entails a vanishing \( \theta_{12...n} \). □
ACKNOWLEDGMENTS

We would like to thank anonymous reviewers for their valuable comments. We also thank Liang He, Qian Yu and Dazhao Pan for their helpful discussions and support on experiments and coding. This work is partially supported by the Natural Science Foundation of China (grants no. 61070044, 611130190 and 61272265), National Program on Key Basic Research Project (973 Program, grant no. 2013CB329304, and the European Union Framework 7 Marie-Curie International Research Staff Exchange Programme (grant no. 247590).

REFERENCES


KIRSCH, A. ET AL. 2012. An efficient rigorous approach for identifying statistically significant frequent itemsets. JACM 59, 12.


