Information Sciences 289 (2014) 8–24

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Probabilistic correlation-based similarity measure on text records ☆

Shaoxu Song [a,*], Han Zhu [a], Lei Chen [b]

[a] KLiss, MoE; TNList; School of Software, Tsinghua University, China
[b] The Hong Kong University of Science and Technology, Hong Kong

ARTICLE INFO

Article history:
Received 29 July 2013
Received in revised form 12 July 2014
Accepted 3 August 2014
Available online 20 August 2014

Keywords:
Similarity measure
Probabilistic correlation
Text record

ABSTRACT

Large scale unstructured text records are stored in text attributes in databases and information systems, such as scientific citation records or news highlights. Approximate string matching techniques for full text retrieval, e.g., *edit distance* and *cosine similarity*, can be adopted for unstructured text record similarity evaluation. However, these techniques do not show the best performance when applied directly, owing to the difference between unstructured text records and full text. In particular, the information are limited in text records of short length, and various information formats such as abbreviation and data missing greatly affect the record similarity evaluation.

In this paper, we propose a novel probabilistic correlation-based similarity measure. Rather than simply conducting the matching of tokens between two records, our similarity evaluation enriches the information of records by considering correlations of tokens. The probabilistic correlation between tokens is defined as the probability of them appearing together in the same records. Then we compute weights of tokens and discover correlations of records based on the probabilistic correlations of tokens. The extensive experimental results demonstrate the effectiveness of our proposed approach.

1. Introduction

Unstructured text records are prevalent in databases and information systems, such as personal information management systems (PIM) and scientific literature digital library (CiteSeer). Various applications, for example similarity search [12], duplicate record detection [8], information integration [1] and so on, rely on the similarity evaluation among these unstructured records of text values. Table 1 shows an example of unstructured record database which stores several citation records as text attributes. Due to various information formats such as abbreviation and data missing, it is not easy to evaluate the similarity of unstructured records in the real world.

Since unstructured text records are text strings of short length (as shown in Table 1), we can apply approximate string matching techniques such as *edit distance* [21] to measure the similarity. However, these character-based matching approaches can only capture limited similarity and fail in many cases such as various word orders and incomplete information formats. Therefore, other than character-based string matching techniques, we can also treat each unstructured record as a text document and apply full text retrieval techniques to measure the record similarity. Specifically, records are repre-

☆ A preliminary, extended abstract of this paper appears in [26].
* Corresponding author.
E-mail addresses: sxsong@tsinghua.edu.cn (S. Song), leichen@cs.ust.hk (L. Chen).

http://dx.doi.org/10.1016/j.ins.2014.08.007

Table 1
Example of unstructured citation records.

No.	Citation
1	S. Guha, N. Koudas, A. Marathe, D. Srivastava, Merging the results of approximate match operations, in: VLDB'04: Proceedings of the 30th International Conference on Very Large Data Bases, 2004, pp. 636–647
2	Guha, S., Koudas, N., Marathe, A., Srivastava D., Merging the results of approximate match operations, in: the 30th International Conference on Very Large Data Bases (VLDB 2004), 2004, pp. 636–647
3	Sudipto Guha, Nick Koudas, Amit Marathe, Divesh Srivastava: Merging the Results of Approximate Match Operations, in: VLDB, 2004, pp. 636–647
4	S. Guha, et al. Merging the results of approximate match operations, in: Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 – September 3, 2004

sented by a set of weighted token features and similarity is computed based on these features. Cohen [4] proposes a word token based *cosine similarity* with *tf*idf* which can detect the similarity of records with various word orders and data missing. Gravano et al. [9] propose a more effective approach by using *q*-grams as tokens of records, which can handle spelling errors in records.

Unfortunately, the characteristics of unstructured text records are different from those of strings in full texts. First, due to the short length of text records, most words appear only once in a record, that is, the *term frequency (tf)* is 1 in most cases of such short text records in databases. We show the statistics of term frequency in Table 2. More than 90% tokens, even the tokens of *q*-grams, appear only once in a record. Therefore, only the *inverse document frequency (idf)* [27] takes effect in the *tf*idf* [24] weighting scheme and no local features of each record are considered. Moreover, the popular matching similarity measure used for full text, *cosine similarity*, is based on the assumption that tokens are independent of each other, and the correlations between tokens are ignored. Due to various information representation formats of unstructured text records such as abbreviation and data missing, latent correlations of records can hardly be detected by only considering the matching of tokens.

Example 1. Consider records No. 3 and 4 in Table 1 with different author representations of "Sudipto Guha, Nick Koudas, Amit Marathe, Divesh Srivastava" and "S. Guha, et al." respectively. By using the *cosine similarity* which is based on the dot product of two record vectors, we have only one matching token "Guha" and the similarity value is low. Even worse, there is no matching token at all between the different representations of the same conference, "Very Large Data Bases" and "VLDB", and the *cosine similarity* value is 0 between these two representations. As a consequence, the *cosine similarity* of records No. 3 and 4 is low, which actually describe the same citation entity. Cohen et al. [5] conclude that full text retrieval techniques, *tf*idf* and *cosine similarity*, do not show the best performance when they are applied directly to text records in databases.

Motivated by the unsuitability of string matching and full text retrieval techniques in measuring similarity between text attribute records, in this paper, we mainly focus on developing the similarity metrics based on the correlation of tokens, and perform the similarity evaluation over records directly without data cleaning. In our similarity approach, rather than matching tokens of records, the correlations between tokens are considered, which help to discover more correlations of short text records with limited information. The correlations between tokens are investigated based on the probability that tokens appear in the same records. Then, these token correlations are utilized in two aspects, i.e. intra-correlation and inter-correlation. The intra-correlations consider the correlations of tokens in a record, and are utilized in the weighting of tokens. Rather than simply assigning equal term frequencies to tokens, we develop the discriminative importance of each token based on the degree of correlations with other tokens in a record. The inter-correlations represent the correlations of tokens between two records, which can further discover the correlations of records in addition to matched tokens. Based on the correlations of tokens, we can perform the similarity evaluation on text records with more diverse formats, for example with abbreviation and data missing.

Our contributions in this paper are summarized as follow:

- We develop a dictionary to capture the probabilistic correlations of tokens, and represent text records with the consideration of both token frequencies and correlations. Highly correlated tokens are merged as phrase tokens to reduce the size.

Table 2
Statistics of term frequency.

Dataset[a]	Term frequency		
	=1	=2	$\geqslant 3$
Cora (word)	96.8%	3.0%	0.2%
Cora (*q*-grams)	93.6%	5.9%	0.5%
Restaurant (word)	98.4%	1.5%	0.1%
Restaurant (*q*-grams)	96.9%	2.9%	0.2%

[a] *Cora* and *Restaurant* are two datasets used in this paper, please refer to Section 6 for details.

- We propose a probabilistic correlation-based feature weighting scheme, namely *correlation weight*, by considering the intra-correlation of tokens in a record. Instead of term frequency, which is equal to 1 in most records without any discriminative ability, the intra-correlation is employed to serve as local features of tokens in a record.
- We design a probabilistic correlation-based similarity function, called *correlation similarity*, by utilizing the inter-correlation of tokens in two records. In particular, we prove that the existing cosine similarity can be interpreted as a special case of the proposed correlation similarity.
- We extend the existing semantic-based word similarity in *WordNet* to our semantic-based record similarity, named *semantic-based similarity* (sbs). In particular we combine the *sbs* method with our *correlation* similarity and propose the *semantic-based correlation similarity* (scor).
- We report an extensive experimental evaluation, to demonstrate superiority of the proposed approach compared with existing measures and our semantic-based measure.

The rest of this paper is organized as follows. We illustrate the probabilistic correlation of tokens in Section 2. Section 3 presents our probabilistic correlation-based weighting scheme and also the probabilistic correlation-based similarity function. In Section 4, we discuss the effectiveness of our approach from a methodological perspective. The extension on semantic-based similarity is introduced in Section 5. Section 6 demonstrates the performance of our approach through an experimental evaluation. In Section 7, we discuss some related work. Finally, we conclude this paper in Section 8. An early extended abstract of this paper is reported in [26].

2. Probabilistic correlation

The cosine similarity measure [4] makes an assumption that tokens in records are independent of each other, and the correlations between tokens are ignored. In practice, however, token correlations do exist, for example, the token "International" has a high probability of appearing together with the token "Conference" in citation records. In this section, we develop a model of correlations between tokens by considering the conditional probability of token co-occurrence.

2.1. Probabilistic correlation definition

The *probabilistic latent semantic analysis* [13,14] considers the joint probability of documents d (i.e., records in our study) and word tokens w, and the *aspect model* is used as a latent class variable model for co-occurrence data. In this paper, we also use the co-occurrence of tokens in the same records to model token correlations. However, we do not consider the conditional probability among tokens, documents and class variables. Instead, we construct the probabilistic correlations between tokens in records directly, and then apply these token correlations to measure the similarity of records.

At first, we consider a word token based correlation. Records are cut into word tokens, and correlations between the tokens are computed. The conditional probability is used to model the probability that tokens appear together in a record, which is defined as follows

$$Pr(t_i|t_j) = \frac{Pr(t_it_j)}{Pr(t_j)}, \tag{1}$$

where $Pr(t_it_j)$ denotes the probability that tokens t_i and t_j appear in the same record, which can be estimated as $Pr(t_it_j) \approx \frac{df(t_it_j)}{N}$, that is, the number of records where both token t_i and t_j appear $df(t_it_j)$ divided by the total number of records N [23]. Therefore, the conditional probability of tokens t_i and t_j can also be described as,

$$Pr(t_i|t_j) = \frac{df(t_it_j)}{df(t_j)}, \tag{2}$$

where $df(t_it_j)$ denotes the number of records where both tokens t_i and t_j appear, and $df(t_j)$ denotes the number of records which contain token t_j.

The conditional probability between tokens t_i and t_j is asymmetric, i.e., $Pr(t_i \mid t_j) \neq Pr(t_j \mid t_i)$. While the similarity between records is often regarded as a symmetric relationship, we define the probabilistic correlation of tokens in a symmetric way,

$$cor(t_i, t_j) = Pr(t_i|t_j) \cdot Pr(t_j|t_i), \tag{3}$$

which represents the probabilistic degree that tokens t_i and t_j belong to the same records. According to Eq. (1), the correlation (3) can be rewritten as

$$cor(t_i, t_j) = \frac{Pr(t_it_j)^2}{Pr(t_j) \cdot Pr(t_i)}, \tag{4}$$

where $Pr(t_it_j)$ is the probability that tokens t_i and t_j appear in the same record. When tokens t_i and t_j match, i.e., t_i and t_j are the same token, we have $cor(t_i, t_i) = 1$. Indeed, a correlation value $cor(t_i, t_j) = 1$ means that the probability of t_i and t_j belonging to the same record is equal to 1 (see more discussions below).

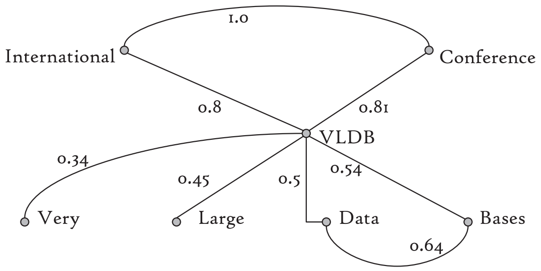

Fig. 1. Dictionary of word token correlation.

Next, we construct a dictionary structure to model the probabilistic correlations of tokens appearing in the dataset. Fig. 1 shows an example of the dictionary. Each node in the dictionary denotes a token, and the undirected edges between any two nodes indicate the correlation of these two tokens. The weight of the edge between nodes t_i and t_j represents the probabilistic correlation weight of token t_i and t_j, i.e. $cor(t_i, t_j)$. We define the dictionary representation formally as follows.

Definition 1 (*Dictionary space*). Given a set of records R, with m tokens appearing in the records, a *dictionary space* is a graph $G_d = \langle V_d, E_d \rangle$, where each vertex $t_i^d \in V_d$ denotes a token t_i, each edge $e_{ij}^d \in E_d$ denotes the probabilistic correlation of t_i and t_j, and the edge weight of e_{ij}^d is equal to $cor(t_i, t_j)$.

The correlation dictionary is basically a statistical thesaurus. Note that some word tokens in the dictionary highly correlate with each other. In fact, some word tokens, for example the page number tokens “636” and “647” of citation records, always appear together in the same records. In other words, we have

$$cor(t_i, t_j) = Pr(t_i|t_j) = Pr(t_j|t_i) = 1 \tag{5}$$

which implies that tokens t_i and t_j always appear in the same records. Therefore, we can merge these kinds of word tokens together into a new token, namely *phrase token*. For instance, “World Wide Web” may appear together in the same records always, therefore, we can merge these three word tokens as a new phrase token.

A *phrase token* t_p is a token comprising several tokens that always appear together in the same records. For any token t_i in t_p, we have $Pr(t_p) = Pr(t_i)$ since all of the tokens in the phrase always appear in the same records and have the same probability $Pr(t_i)$. For any other token t_l in the dictionary, we have $Pr(t_l t_p) = Pr(t_l t_i)$, which implies that the probability of any token t_l appearing together with token t_i of phrase token t_p in a record is equal to the probability that token t_l appears together with phrase token t_p. Thereby, we have the correlation between the new phrase token t_p and any other tokens t_l in the dictionary,

$$cor(t_l, t_p) = \frac{Pr(t_l t_p)^2}{Pr(t_l) \cdot Pr(t_p)} = \frac{Pr(t_l t_i)^2}{Pr(t_l) \cdot Pr(t_i)} = cor(t_l, t_i). \tag{6}$$

By merging word tokens into phrase tokens, we can reduce the size of the dictionary and the records. As shown in Table 3, the total number of tokens in the dictionary is reduced significantly in the phrase token based approach. Furthermore, the average number of tokens in each record is also reduced by using the phrase token representation. After merging tokens into phrases, we have the following property of token correlations.

Proposition 1. *Consider the correlation between tokens t_i and t_j in a dictionary space with phrase token. If t_i and t_j are not matching, i.e., $t_i \neq t_j$, then we have*

$$0 < cor(t_i, t_j) < 1.$$

Proof. First, if a correlation exists between t_i and t_j in the dictionary, these two tokens appear together at least in one record. Therefore, we have $cor(t_i, t_j) > 0$. According to the definition of $cor(t_i, t_j) = Pr(t_i \mid t_j) \cdot Pr(t_j \mid t_i)$, the correlation value satisfies $cor(t_i, t_j) \leqslant 1$. Since all the tokens with $cor(t_i, t_j) = 1, t_i \neq t_j$ are merged as a new phrase token, the condition of $cor(t_i, t_j) < 1$ is satisfied. In summary, we have $0 < cor(t_i, t_j) < 1$ for any $t_i \neq t_j$. □

Table 3
Statistics of tokens.

	Dictionary size	Average record size
Cora (word)	912	23.53
Cora (phrase)	679	21.61
Restaurant (word)	2990	9.34
Restaurant (phrase)	1484	7.45

According to this property, we can generalize the token matching-based similarity function to support the correlation similarity, which is discussed in Section 3.

2.2. Intra-correlation and inter-correlation

The probabilistic correlation between two tokens implies the probability that these two tokens belong to the same record. Once the probabilistic correlations between tokens are investigated, we can utilize the correlations in two perspectives, i.e. *intra-correlation* and *inter-correlation*.

The *intra-correlation* indicates the correlation of tokens in a single record. However, for tokens t_1 and t_2 in a record, the intra-correlation value between t_1 and t_2 is calculated based on the whole corpus. As shown in Fig. 2, the tokens in a record might correlate with each other. Intuitively, a token with more and higher correlations to others implies that this token is more important in the current record where the token is. Therefore, the correlations of tokens in a record can be used in the feature weighting of the record. Section 3.1 discusses the feature weighting scheme by considering the *intra-correlation* of tokens in the record.

The *inter-correlation* indicates the correlation of tokens between two records. For example, consider the token "Very" in a record r_1 and the token "VLDB" in a record r_2 in Fig. 3. As shown in the dictionary in Fig. 1, probabilistic correlation exists between token "Very" and "VLDB", since both tokens may appear in the same records frequently throughout the entire dataset. This large correlation between token "Very" and "VLDB" implies a high probability that these two tokens describe the same record in real world. Considering all token correlations between r_1 and r_2, we can estimate the overall probability that these two records describe the same entity, i.e., the similarity between record r_1 and r_2. In Section 3.2, we present our correlation similarity function based on the *inter-correlation* of tokens between two records.

2.3. Correlation-based representation

Once the dictionary with probabilistic correlation is constructed, we devise the probabilistic correlation-based representation of each record. Two factors of tokens should be represented in the model in our case, i.e., the frequency weights of tokens and the probabilistic correlations between tokens. The classical *vector space model* [24] for full text documents cannot be applied directly, since only the frequency based weights of tokens are represented in the model without the correlations of tokens. Regarding these token correlation considerations, we use a probabilistic correlation space model. For each token, we associate a weight and several correlations with other tokens, instead of a single weight as in the vector space model.

Definition 2 (*Record space*). Given a record r with m tokens, the *record space* of r is a graph $G_r = \langle V_r, E_r \rangle$, which is indeed a sub-graph of the dictionary. Each vertex $t_i^r \in V_r$ is associated with a value w_i^r which denotes the weight of token t_i in the record. Each edge $e_{ij}^r \in E_r$ has the same weight of edge e_{ij}^d in the dictionary, i.e., $cor(t_i, t_j)$.

We show an example of a record with correlations of tokens in Fig. 2. Different from the vector space model with a vector of weighted tokens w_i, we represent both the weights of tokens and the correlations between these tokens. Let M_r be the adjacent matrix of the graph G_r in the record space r, i.e., the record space can also be treated as a sparse matrix. Each element of (row i, col j) in the matrix M_r denotes the correlation weight $cor(t_i, t_j)$.

As discussed, the probabilistic correlation $cor(t_i, t_j)$ can be regarded as the probability that tokens t_i and t_j describe the same record. This correlation can be utilized to determine the weights of tokens and further detect correlations of records, which are discussed in the following section.

3. Record similarity measure

In this section, we illustrate our probabilistic correlation-based record similarity measure. Firstly, we discuss the weighting scheme of tokens in the records. The correlation between tokens is used in the weighting of tokens. Then we introduce our correlation similarity function which is also based on the correlation between tokens.

3.1. Feature weighting

We first study the discriminative features of tokens from a global view of all records in the entire dataset. The *inverse document frequency (idf)* [27], first proposed by Karen Sparck Jones, is based on the essential intuition that a token appearing

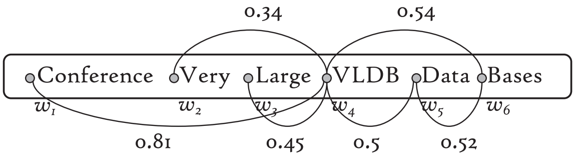

Fig. 2. Record with probabilistic correlation.

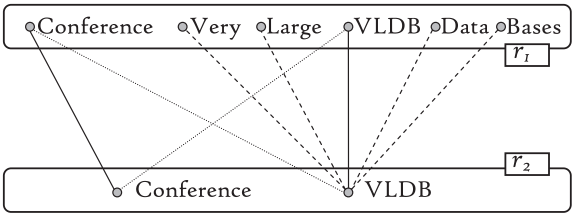

Fig. 3. Inter-correlation between two records.

frequently in different documents (records) is not a good discriminator and should be associated with low feature weights; while a token with low document frequency means it is more relevant to those documents in which it appears. The basic formula of *idf* is

$$idf(t_i) = \log\left(\frac{N}{df(t_i)}\right) \tag{7}$$

where N denotes the total number of documents (records), and $df(t_i)$ is the number of documents (records) that contain token t_i.

In order to discriminate from each other, a local feature weighting of each record is performed. The *term frequency* is adopted in full text retrieval as local features of each document. However, as the statistics in Table 2 show, *term frequency* is probably equal to 1 in most cases of short unstructured text records, which indicates that only *idf* takes effect in *tf*idf* and no local features of records are considered.

In this paper, we merge word tokens into phrase tokens according to their high correlation with each other. Therefore, we can use the size of phrase tokens, i.e., the number of word tokens in phrase tokens, as *phrase weight* $w_p = |t_p|$. Moreover, we utilize the probabilistic correlations between tokens, and develop a correlation-based weighting scheme of records.

Instead of *term frequency (tf)* with $w_i = 1$ in most cases, we propose a new local weighting scheme of tokens in a record, namely *correlation weight*. Since we use the conditional probability as correlations between tokens, tokens with more and higher correlations to the others in the record are more likely to represent the record and can be treated as an important local feature. Therefore, we introduce the new token weighting scheme, which is based on the degree of the token correlation with other tokens in the same record.

Definition 3 (*Correlation weight*). Given a record space r with an initial weight w_i of each token t_i, the *correlation weight* of token t_i in the record r is defined as:

$$cow(t_i) = w_i + \frac{\sum_{t_j \in r} w_j \cdot cor(t_j, t_i)}{|r|} \tag{8}$$

where $cor(t_j, t_i)$ denotes the probabilistic correlation between tokens t_i and t_j in the record, and $|r|$ means the total number of tokens m in the record.

The initial weight w_i of each token t_i can be the phrase weight or the term frequency. The correlation weight denotes the reliability and importance of the token t_i in the record. A higher correlation weight implies a higher probability that if token t_i appears in the record, other tokens t_j will also appear in the records. In other words, the more tokens t_j that show high correlation with token t_i, the higher the probability token t_i is relevant to the record.

Moreover, in correlation weight, only the probabilistic correlations of tokens in the same record are considered. In order to make the record features as discriminative as possible, we can further combine the correlation weight with global statistic weights in the weighting scheme, for example, inverse document frequency (idf). Following the convention of the *tf*idf* approach, we define the *cow*idf* weight as,

$$cow * idf(t_j) = cow(t_j) \cdot idf(t_j). \tag{9}$$

In the experimental section, we will show that our *cow*idf* word weight function performs better than the traditional *tf*idf* word weight function. Actually, w_i and w_j in Formula (12) is *cow*idf* weight.

3.2. Similarity function

The similarity between records can be quantified by the overlaps of common tokens in two records. The classical *cosine similarity* function [29] is widely used in the similarity measure of text strings [4,9], which is described as,

$$cos(r_1, r_2) = \frac{r_1 \cdot r_2}{\|r_1\| \cdot \|r_2\|} = \frac{\sum_{t_i = t_j} w_i w_j}{\sqrt{\sum w_i^2 \sum w_j^2}}, \tag{10}$$

where $cos(r_1, r_2)$ is the cosine similarity value of records r_1 and r_2, w_i denotes the weight of token t_i in record r_1, and w_j denotes the weight of token t_j in record r_2.

The cosine similarity function is based on the matching of tokens. Therefore, records with various representations, for example "Bases" and "VLDB", are treated as two different tokens without any correlation at all. In our study, since text records are always short in length with limited information, we investigate the latent similarity based on token correlations of two records. Our correlation-based similarity function generalizes the cosine similarity between two records by considering not only the matching tokens but also the inter-correlations of tokens.

Note that the relationship between tokens of two records is single-to-single matching in cosine similarity, in other words, one token in record r_1 is related/matched with no more than one token in the other record r_2. However, in our probabilistic correlation, one token may be correlated with multiple tokens in the other record. In order to capture these multiple-to-multiple tokens correlations in the similarity evaluation, we consider three kinds of inter-correlations of tokens between two records.

(1) The first correlation is between matching tokens, for example, the correlation between the "Conference" of two records in Fig. 3. These correlations of matching tokens have already been considered in the cosine similarity.
(2) The second correlation is between two tokens which appear in both records, for example, the correlation between "Conference" and "VLDB" with dotted lines in Fig. 3. Since these two tokens appear in both records and the correlations have been detected respectively by the first type of correlations, we do not have to account for them again. For example, if record r_1 and record r_2 both contain words "very" and "large", we will calculate $cor(very, very)$ and $cor(large, large)$ in the first correlation type. Then, in the second situation, it will be redundant to take $cor(very, large)$ into account since both "very" and "large" have been considered.
(3) The third correlation is between two tokens at least one of which does not appear in both records. For example, the token "Very" does not appear in the second record r_2 in Fig. 3, which means that no relationship exists between "Very" in r_1 and the tokens in r_2 referring to the traditional matching approach. However, the probabilistic correlation does exist between "Very" and "VLDB" and can contribute in finding the correlation of two records. The token correlation should be taken into consideration in this case.

According to the different scenarios of inter-correlations, we can further refine the probabilistic inter-correlation between tokens t_i, t_j in two records r_1, r_2, respectively, as follows.

$$cor(t_i, t_j) = \begin{cases} 1 & t_i = t_j \\ 0 & t_i \neq t_j \text{ and } t_i \in r_2 \text{ and } t_j \in r_1 \\ Pr(t_i|t_j) \cdot Pr(t_j|t_i) & t_i \notin r_2 \text{ or } t_j \notin r_1 \end{cases} \tag{11}$$

Consequently, we can calculate the records similarity not only based on the matching pairs of tokens in cosine similarity, but also based on the probabilistic correlated pairs of tokens in two records. Let M be all the pairs of tokens with inter-correlations of two records described in (11), which satisfy the user specified minimum correlation threshold $cor(t_i, t_j) \geqslant \eta$. The correlation similarity function can be defined as

Definition 4 (*Correlation similarity*). Given two records r_1 and r_2, the *correlation similarity* of r_1 and r_2 is defined as

$$sim(r_1, r_2) = \frac{\sum_{(t_i,t_j)\in M} w_i w_j cor(t_i, t_j)}{\|r_1 \oplus M\| \cdot \|r_2 \oplus M\|} \tag{12}$$

where w_i, w_j denote the weight of token t_i, t_j respectively, $cor(t_i, t_j)$ denotes the probabilistic correlation between t_i and t_j in the token correlation set M of r_1 and r_2, and $\|r_1 \oplus M\|, \|r_2 \oplus M\|$ denote the sizes w.r.t. correlations M of r_1, r_2, respectively.

Unlike the single-to-single relationships of matching pairs in cosine similarity, our correlation set M defines multiple-to-multiple correlations of tokens as shown in Fig. 3. In order to normalize the similarity value, we use $\|r_1 \oplus M\| \cdot \|r_2 \oplus M\|$ rather than $\|r_1\| \cdot \|r_2\|$, where

$$\|r_1 \oplus M\| = \sqrt{\sum_{(t_i,t_j)\in M} (w_i^2 cor(t_i, t_j)) + \sum_{t_i \in r_1 \setminus r_2} w_i^2}$$

and $\|r_2 \oplus M\|$ can be computed in a similar way.

Theorem 1. *The value of* $sim(r_1, r_2)$ *ranges from 0 to 1.*

Proof. Assume that records r_1 and r_2 have m and n tokens respectively. Construct a $m * n$-dimensional vector. We use t_i–t_j to present a dimension in the new vector space, in which $t_i \in r_1$ and $t_j \in r_2$. On dimension t_i–t_j, $\vec{r_1}$ and $\vec{r_2}$ weight $w_i * \sqrt{cor(t_i, t_j)}, w_j * \sqrt{cor(t_i, t_j)}$ respectively.

In the new vector space, $\vec{r_1} \cdot \vec{r_2} = \sum_{(t_i,t_j)\in M} w_i w_j cor(t_i,t_j)$. $\|r_1\| \cdot \|r_2\| = \sqrt{\sum_{(t_i,t_j)\in M} (w_i^2 cor(t_i,t_j))} \cdot \sqrt{\sum_{(t_i,t_j)\in M} (w_j^2 cor(t_i,t_j))}$. Obviously, $\|r_1\| \cdot \|r_2\| \leqslant \|r_1 \oplus M\| \cdot \|r_2 \oplus M\|$. The *cosine* distance in the new vector space is $\frac{\vec{r_1}\cdot\vec{r_2}}{\|r_1\|\cdot\|r_2\|} \in [0,1]$. Consequently, our $sim(r_1,r_2) \in [0,1]$. □

4. Methodology analysis

In this section, we analyze the effectiveness of our approach from a methodological perspective, especially in dealing with various information formats of unstructured text records, such as abbreviation and incomplete information.

Since we study the similarity of unstructured text records with short length and limited information, our correlation similarity function relaxes the constraint of token matching in the cosine similarity function, by considering the further inter-correlations of tokens between two records. Therefore, the correlation-based similarity can be interpreted as a generalization of the cosine similarity.

Theorem 2. *The correlation similarity function is a generalization of the cosine similarity function. If the minimum correlation threshold is set to $\eta = 1$, the correlation similarity is equivalent to the cosine similarity.*

Proof. In cosine similarity, only matching pairs of tokens from two records are considered. Therefore, it is sufficient to prove that the correlation $cor(t_i,t_j) = 1$ in correlation set M if and only if $t_i = t_j$, i.e., t_i and t_j are matching. Assume that there exist two different tokens $t_i \in r_1, t_j \in r_2, t_i \neq t_j$ with correlation $cor(t_i,t_j) = 1$. We have $Pr(t_i \mid t_j) = Pr(t_j \mid t_i) = 1$, which means that tokens t_i and t_j always appear together in the same record. However, according to the definition of *phrase token*, tokens with $Pr(t_i \mid t_j) = Pr(t_j \mid t_i) = 1$ should be merged to a new phrase token. In other words, tokens $t_i \in r_1, t_j \in r_2, t_i \neq t_j$ with correlation $cor(t_i,t_j) = 1$ do not exist.

Consequently, we have

$$\|r_1 \oplus M\| = \sqrt{\sum_{(t_i,t_i)\in M} (w_i^2 cor(t_i,t_i)) + \sum_{t_i\in r_1\setminus r_2} w_i^2} = \sqrt{\sum_{t_i\in r_1\cap r_2} w_i^2 + \sum_{t_i\in r_1\setminus r_2} w_i^2} = \|r_1\|,$$

which is similar for r_2. Moreover, the correlation threshold $\eta = 1$ implies

$$\sum_{(t_i,t_j)\in M} w_i w_j cor(t_i,t_j) = \sum_{t_i=t_j} w_i w_j.$$

Combing the above two derivations, we have $sim(r_1,r_2) = cos(r_1,r_2)$. □

The probabilistic correlation-based similarity is effective, especially in evaluating records with data missing. For instance, in Fig. 4, we use "Guha et al." to represent "Guha, S., Koudas, N., Marathe, A., Srivastava D.", if the author list is too long in citation records. Unfortunately, these kinds of highly correlated relationships with data missing are difficult to address by the traditional token matching approaches such as cosine similarity. In our probabilistic correlation-based similarity, we investigate the correlation between the author "Guha" and other authors, since they may appear together in other records without data missing. Then we utilize these token correlations to discover the relationship of "Guha" and "Guha, S., Koudas, N., Marathe, A., Srivastava D." in records r_1 and r_2 respectively.

Furthermore, since we use the probabilistic correlation between tokens in the similarity function, our approach can address the more complicated problem of the abbreviation similarity. Again, the similarity between "VLDB" and "Very Large Data Bases" is not easy to detect by directly using matching techniques such as cosine similarity. However, the words and their abbreviation may appear in the same records frequently, which means that high probabilistic correlation exists between them. As the example in Fig. 3, we can use the correlations between token "VLDB" and {"Very", "Large", "Data", "Bases"} to find the similarity between "VLDB Conference" and "Very Large Data Bases Conference". Therefore, our similarity function is able to capture the similarity of words and their abbreviation by using the probabilistic correlation, no matter what kind of abbreviation rule it applies.

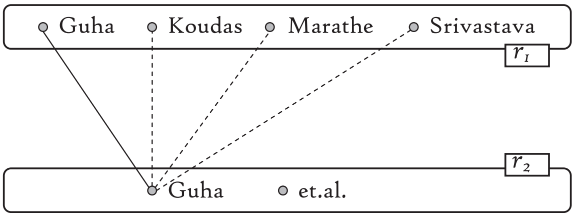

Fig. 4. Correlation of records with data missing.

To calculate the correlation similarity between words and to find phrase tokens efficiently, an inverted file should be maintained. Assume that the corpus has m different words and n records. With the help of inverted file, all the phrase tokens can be find in $O(m * m * n)$ time.

It's very easy to get similarity value between two records with inverted file. Assume that records r_1 and r_2 have m_1 and m_2 tokes respectively, $sim(r_1, r_2)$ can be computed in $O(m_1 * m_2)$ time with $cor(t_i, t_j)$ known.

5. Integrating with semantic-based similarity from external sources

Our proposed correlation technique can successfully obtain a part of the relationships among tokens that appear together, without any external sources. Nevertheless, when external knowledge bases are available, the ontology-based or semantic-based approach can retrieve more token relationships. In this section, we present a novel approach to incorporate our correlation-based similarity with the semantic-based similarity.

5.1. Semantic based similarity

Besides the *tf*idf* approach, there are some ontology-based and semantic-based methods to measure content similarity with corpus information. The outstanding ideas of them, e.g., Lin's information-theoretic definition of similarity [18], exploit *WordNet* taxonomy to define similarity between words. Specifically, the similarity value between two classes, c_1 and c_2, is given as follow:

$$sbs(c_1, c_2) = \frac{2 * \log p(c_0)}{\log p(c_1) + \log p(c_2)} \tag{13}$$

where c_0 is the most specific class that subsumes c_1 and c_2, and $p(c)$ is the probability of c given by *WordNet*.

Based on formula (13), the similarity metric between two words, w_1 and w_2, is defined by:

$$sbs(w_1, w_2) = \max_{c_1, c_2}[sbs(c_1, c_2)] \tag{14}$$

where w_1 belongs to c_1, and w_2 belongs to c_2.

Definition 5 (*Semantic-based similarity*). Given two records r_1 and r_2, the *semantic-based similarity* of r_1 and r_2 is defined as

$$sbs(r_1, r_2) = \frac{\sum_{(t_i, t_j) \in M} w_i w_j sbs(t_i, t_j)}{\|r_1 \oplus M\| \cdot \|r_2 \oplus M\|} \tag{15}$$

where $w_i,\ w_j$ denote the weight of token $t_i,\ t_j$ respectively, and $sbs(t_i, t_j)$ denotes the semantic-based similarity between t_i and t_j in the token correlation set M of r_1 and r_2. M is the set which contains all pairs of (t_i, t_j), where $t_i \in r_1,\ t_j \in r_2$ and $sbs(t_i, t_j) \neq 0$.

5.2. Semantic based correlation similarity

Analogy to our definition of $cor(t_1, t_2)$, we define the semantic-based correlation $scor(t_1, t_2)$ in this section. Let t_i and t_j be tokens in r_1 and r_2, we define $scor(t_i, t_j)$ as follow:

$$scor(t_i, t_j) = \begin{cases} 1 & t_i = t_j \\ 0 & t_i \neq t_j \text{ and } t_i \in r_2 \text{ and } t_j \in r_1 \\ \alpha Pr(t_i|t_j) \cdot Pr(t_j|t_i) + (1-\alpha) sbs(t_i, t_j) & t_i \notin r_2 \text{ or } t_j \notin r_1 \end{cases} \tag{16}$$

In the first and the second cases, *scor* similarity value is the same as *cor*. In the third one, semantic-based similarity and our correlation similarity are combined together, with a factor α, in the range of $[0, 1]$. The proportion of contribution by correlation and semantic similarity could be controlled by α according to specific application scenarios.

Definition 6 (*Semantic-based Correlation Weight*).

$$scow(t_i) = w_i + \frac{\sum_{t_j \in r} w_j \cdot scor(t_j, t_i)}{|r|} \tag{17}$$

Definition 7 (*Semantic-based Correlation similarity*). Given two records r_1 and r_2, the *semantic-based correlation similarity* of r_1 and r_2 is defined as

$$scor(r_1, r_2) = \frac{\sum_{(t_i, t_j) \in M} w_i w_j scor(t_i, t_j)}{\|r_1 \oplus M\| \cdot \|r_2 \oplus M\|} \tag{18}$$

where $w_i,\ w_j$ denote the weight of tokens $t_i,\ t_j$ respectively, $scor(t_i, t_j)$ denotes the probabilistic semantic-based correlation between t_i and t_j in the token correlation set M of r_1 and r_2, and $\|r_1 \oplus M\|,\ \|r_2 \oplus M\|$ denote the sizes w.r.t. semantic-based correlations M of r_1, r_2, respectively.

Corollary 1. *The semantic-based correlation similarity function is a generalization of the cosine similarity function. If the minimum correlation threshold is set to $\eta = 1$, the semantic-based correlation similarity is equivalent to the cosine similarity.*

Proof. Similar to the proof of Theorem 2, we assume that there exist two different tokens $t_i \in r_1,\ t_j \in r_2,\ t_i \neq t_j$ with correlation $scor(t_i, t_j) = 1$. We have $Pr(t_i \mid t_j) = Pr(t_j \mid t_i) = 1$, and $sim(t_i, t_j) = 1$, which means that tokens t_i and t_j always appear together in the same record. But tokens $t_i \in r_1,\ t_j \in r_2,\ t_i \neq t_j$ with correlation $cor(t_i, t_j) = 1$ do not exist.

Consequently, we have

$$\|r_1 \oplus M\| = \sqrt{\sum\nolimits_{(t_i,t_i)\in M}(w_i^2 scor(t_i, t_i)) + \sum\nolimits_{t_i \in r_1 \setminus r_2} w_i^2} = \sqrt{\sum\nolimits_{t_i \in r_1 \cap r_2} w_i^2 + \sum\nolimits_{t_i \in r_1 \setminus r_2} w_i^2} = \|r_1\|,$$

which is similar for r_2. Moreover, the correlation threshold $\eta = 1$ implies

$$\sum_{(t_i,t_j)\in M} w_i w_j scor(t_i, t_j) = \sum_{t_i = t_j} w_i w_j.$$

With the above two derivations, we have $sim(r_1, r_2) = cos(r_1, r_2)$. □

6. Experimental evaluation

In this section, we report experimental results. Section 6.1 evaluates the performance of our probabilistic correlation-based techniques with various settings. And Section 6.2 presents the comparison with existing approaches.

Dataset. We employ two datasets in our experiments, *Cora* and *Restaurant*.[1] The first dataset *Cora*, prepared by McCallum et al. [19], consists of 1295 citation records of 122 research papers. The average length of records in *Cora* is 23.53 (i.e., the average number of word tokens in records separated by blank). The *Restaurant* data set, prepared by Tejada et al. [28], consists of 864 restaurant name and address records with 112 duplicates. The average length of records in *Restaurant* is 9.34.

Data preparation. We merge all the information in each record together in an unstructured text record. In the second experiment, in order to simulate and test a dirty dataset with different data missing rates, we also remove words in the records randomly with certain miss rates. For example, a dataset with a missing rate of 0.2 means that 20% words are missing in the dataset. For each pair of records in the dataset, we compute the similarity to determine whether or not these two records describe the same entity.

Effectiveness criteria. We adopt *f-measure* with *precision* and *recall* [29] as the criteria to evaluate the effectiveness of different similarity measures,

$$precision(S_a, S_f) = \frac{|S_a \cap S_f|}{S_f},$$

$$recall(S_a, S_f) = \frac{|S_a \cap S_f|}{S_a},$$

$$f\text{-}measure(S_a, S_f) = \frac{2 \ * \ recall(S_a, S_f) \ * \ precision(S_a, S_f)}{recall(S_a, S_f) + precision(S_a, S_f)},$$

where S_a means actual pairs of records which represent the same entity and S_f means pairs of records found with high similarity value. All of these three metrics are in the range of $[0, 1]$. Precision denotes the correctness of answers, while Recall denotes the completeness of answers. *f*-Measure is the balance of precision and recall and can be regarded as the overall accuracy performance.

Token generation. In the preprocessing of unstructured records, we first cut the records into tokens. In the word token-based approach, the records are split by the separator ' ', i.e., *blank*. For the *q*-grams tokens, according to the study of Gravano et al. [9], we set $q = 3$. For example, "Very Large" is cut to a set of 3-grams {"Ver","ery","ry ","y L"," La","Lar","arg","rge"}.

6.1. Evaluating probabilistic correlation-based similarity

In the first experiment, we observe the distribution of feature weights by using different weighting schemes in *Cora*. First, we present the token weights of an example record by applying two types of feature weighting, i.e., term frequency (tf) and correlation weight (cow). As the results show in Fig. 5, the term frequency of all tokens is equal to 1, which means all tokens are of equal importance in the record. Then, we apply the correlation weight with phrase token to the same record in Fig. 5

[1] http://www.cs.utexas.edu/users/ml/riddle/data.html.

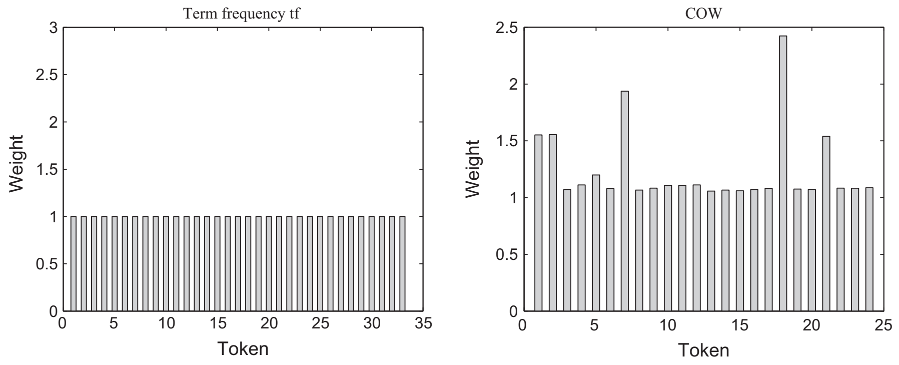

Fig. 5. Comparison of feature weights.

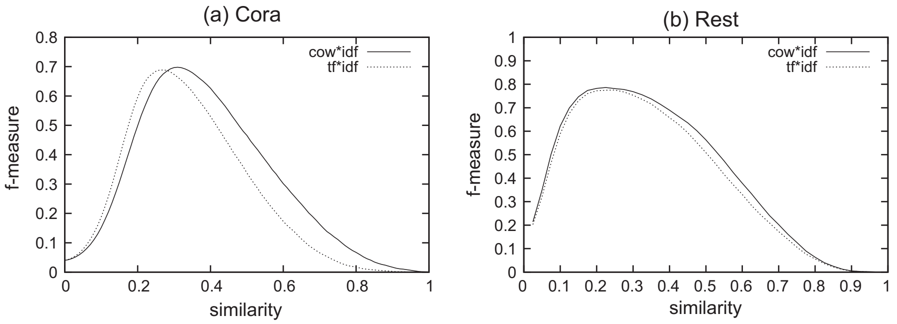

Fig. 6. *f*-Measure of feature weights.

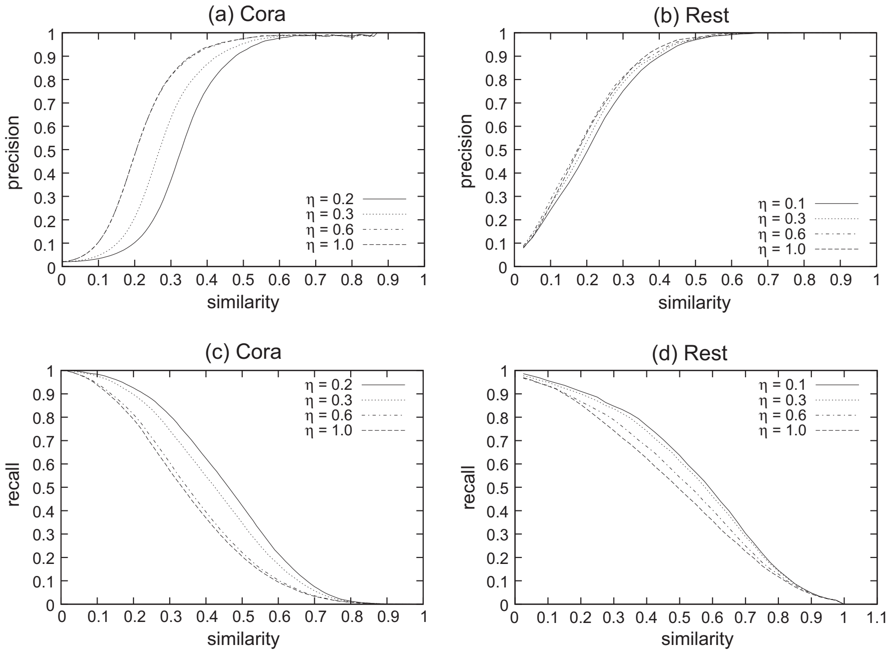

Fig. 7. Precision and recall under different minimum correlation thresholds η.

where tokens show different weights. Rather than the equal term frequency of tokens, the correlation weight (cow) can tell the different importance of tokens in a record. Moreover, the size of the records (i.e., the number of tokens) is reduced from 33 tokens in the term frequency (tf) approach to 24 tokens in our correlation weight (cow) approach.

In order to demonstrate the effectiveness of the correlation weight (cow), we also compare these two weighting schemes under the same similarity function, cosine similarity. The results in Fig. 6 show the improvement of *cow**idf* even by using the cosine similarity. All the experiments under different similarity thresholds produce higher *f*-measure results of *cow**idf* than *tf**idf*. It verifies the ability of our correlation-based approach in distinguishing the importance of different tokens. In fact, the *cow**idf* approach can achieve a better accuracy by combining it with the correlation similarity function which is discussed in the next section.

Next, we study the impact and selection of the minimum correlation threshold η in the correlation similarity function in *Restaurant*. According to Theorem 2, the correlation similarity is equivalent to the cosine similarity when we set the threshold to $\eta = 1$. As shown in Fig. 8, Rest dataset, when the minimum threshold η is shrinking from 1 to 0.1, more correlations will be taken into account and the *f*-measure improves. If the threshold is too large, then only a few correlations will be considered in the similarity and the accuracy might be slightly improved. On the other hand, however, the accuracy is not improved significantly from the low thresholds of $\eta = 0.3$ to $\eta = 0.1$. Since these correlations are too small, many of them with low weights may not influence the results very much. Therefore, as shown in Fig. 8, the highest accuracy is achieved with similar values when the threshold η is around 0.3. Moreover, as shown in Fig. 7, the precision under different thresholds does not change very much, while the recall improves when more correlations are considered. The results indicate that our similarity function could find more latent correlated pair of records by considering the token probabilistic correlations. Meanwhile, $\eta = 0.3$ gives the highest *f*-measure in Cora dataset.

6.2. Comparing different similarity measures

In the second experiment, we compare three similarity measures, including our probabilistic correlation-based similarity with phrase tokens (in short, "Correlation with phrase"), *q*-grams based cosine similarity with *tf**idf* (in short, "Cosine with *q*-grams"), and word token based cosine similarity with *tf**idf* (in short, "Cosine with word").

Cora dataset. First, we present the results of three approaches in the *Cora* dataset. The minimum correlation threshold of correlation similarity is $\eta = 0.2$. The results in Fig. 9 demonstrate the superiority of our correlation-based similarity measure. We consider not only the matching tokens between records but also the probabilistic correlation of the tokens that do not match. Therefore, our approach can detect more similar records. Furthermore, rather than the term frequency weighting with almost the same weight of each token (equal to 1 as shown in Fig. 5), the correlation-based weighting scheme enhances the local feature of each token in a record and improves the accuracy of the similarity measure. As shown in Fig. 9, the correlation-based similarity achieves higher *f*-measure than the cosine similarity. The results of word tokens and *q*-grams in the cosine similarity approaches are quite similar.

In order to prove that our correlation approach is significantly better than the other two approaches, we run the *t*-test for *f*-measure in Fig. 9 In *Cora* dataset, $t = 6.5491$ between correlation approach and cosine similarity, and $t = 9.1835$ between correlation approach and *q*-grams cosine similarity. For the *t*-test where sample number $n = 40$, if the value of t is no less than 2.0211, then we have a possibility of 95% to say that these two samples are significantly different. That's to say, the results of the *t*-test show that the improvement of our correlation approach is pretty obvious.

Restaurant dataset. Then, we run the experiment on the second dataset Restaurant. The minimum correlation threshold of correlation similarity is $\eta = 0.2$. As the best performance of these three approaches in Fig. 10, the correlation-based approach outperforms the word token based cosine similarity by about 10% in terms of *f*-measure over *Restaurant* dataset. Moreover, the cosine similarity with word tokens achieves even significantly better results than the *q*-grams.

All the experiments on these two datasets show that our probabilistic correlation-based similarity measure achieves higher accuracy than the cosine similarity approaches. And the recall can be significantly improved by using the proposed

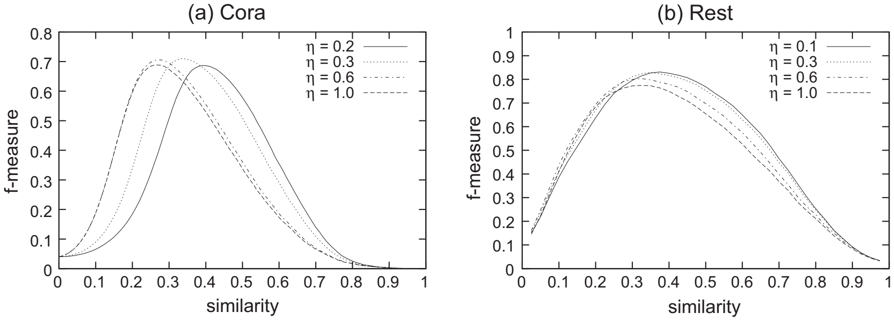

Fig. 8. *f*-Measure under different minimum correlation thresholds η.

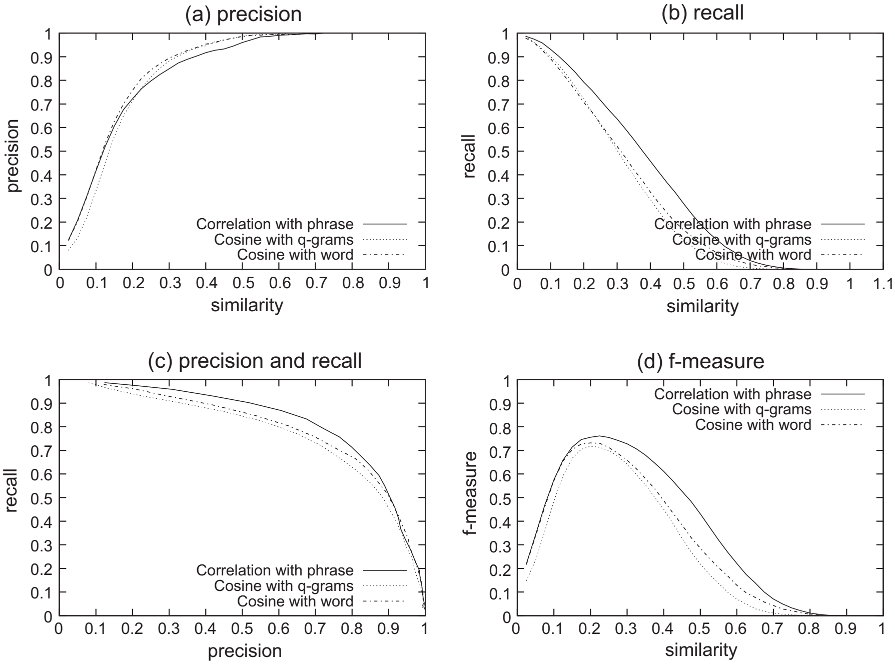

Fig. 9. Accuracy comparison of similarity measure approaches in *Cora*.

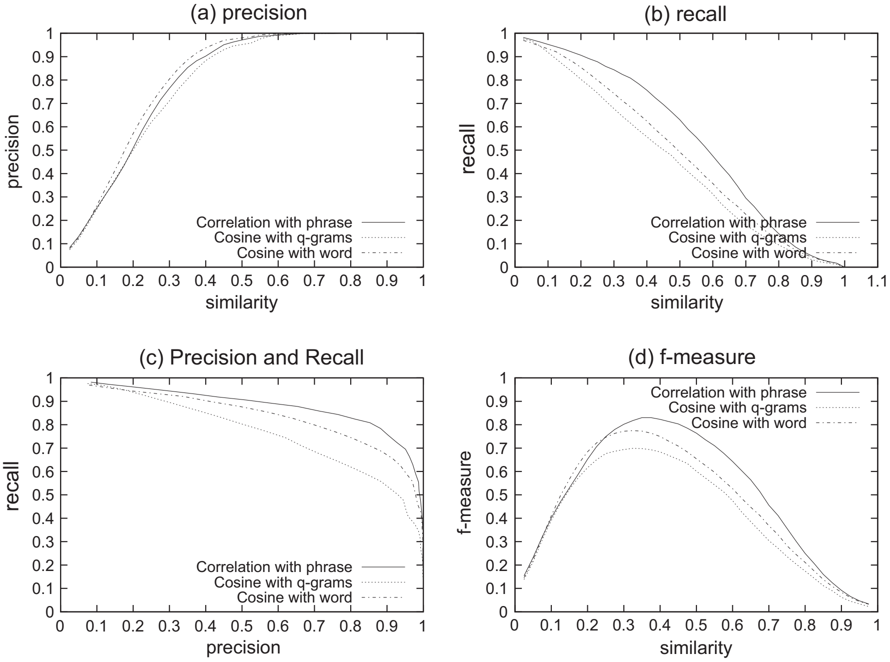

Fig. 10. Accuracy comparison of similarity measure approaches in *Restaurant*.

correlation approach, compared with existing methods, while the corresponding precision keeps high. By using the probabilistic correlations of tokens, we can further find the latent correlated records and consequently improve the accuracy of similarity measure. The *q*-grams does not improve the performance of cosine similarity compared with the word tokens, since the *q*-grams cannot contribute more than the word tokens in dealing with various data formats such as abbreviations.

Like the *t*-test in *Cora*, we also run *t*-test for *Restaurant* dataset. In the results, $t = 6.5541$ between correlation approach and cosine similarity, and $t = 9.0249$ between correlation approach and *q*-grams cosine similarity. From the *t* values, we can make the same conclusion with *Cora* dataset.

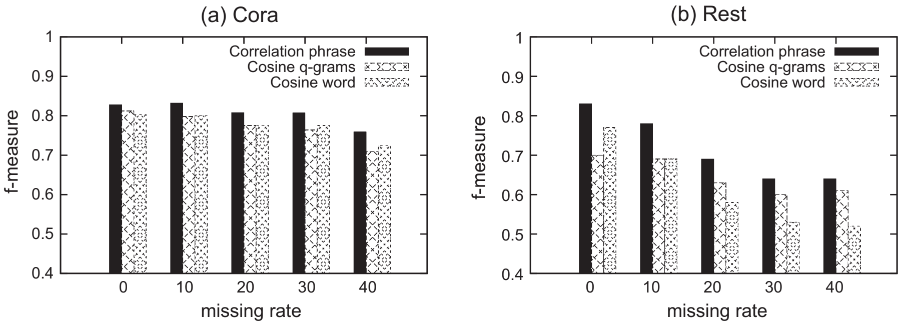

Fig. 11. Accuracy of similarity measure approaches with different data missing rates.

Data missing rate. Finally, we evaluate the performance of three similarity measures under different data missing rates of *Cora*. Fig. 11 reports the best results of each measure under several missing rates. Our correlation-based similarity achieves a higher *f*-measure under all studied missing rates. When the data missing rate is high, e.g. 0.4, too many tokens are absent from the records and the token correlations cannot be constructed accurately. Thus, the accuracy of correlation similarity drops largely as well as the cosine similarity. Furthermore, the *q*-grams approach achieves a lower accuracy than the word token approach in *Cora*, especially when the missing rate is high. However, in *Restaurant*, *q*-grams method drops much slower than cosine similarity.

The experimental results demonstrate the superiority of our correlation-based approach, especially under the dataset with data missing. Owing to the missing data, e.g. “Very Data” with a word “Large” missing, the matching pairs of tokens reduce, which makes it difficult for cosine similarity to find the correlations between records. Even worse, the *q*-grams approach reserves the connections between words, for example, the token “y D” of “Very Data”. However, “y D” does not appear in the original data “Very Large Data” without data missing. Since the record sizes are probably small, such an error token affects the similarity value largely. Therefore, the *q*-grams approach conducts a worse performance than the word token approaches in *Cora*, due to the error tokens caused by data missing.

In *Restaurant* we exploit another method to generate error in data file. When data missing rate is 20 percent, the letter “a” has a probability of 0.2 to be attached to a single word. For example, we may change the word “very” to “avery”. With this method, it’s reasonable that the *q*-grams method performs better than cosine similarity, cause that the 3-grams keep unchanged.

In summary, our probabilistic correlation-based similarity is effective in dealing with various information formats such as abbreviation and data missing. The experimental evaluation shows the effectiveness of our approach, including the correlation-based feature weighting scheme and the correlation similarity function. The probabilistic correlation-based similarity measure achieves higher accuracy than the cosine similarity measures with either word tokens or *q*-grams.

6.3. Comparison with semantic-based similarity

In this section, we compare our semantic-based correlation similarity in two datasets by changing the parameters of α and η in Figs. 12 and 13. Both *Restaurant* and *Cora* show that a moderately large value of α and η give the highest *f*-measure. Nevertheless, the semantic-based correlation method performs better than semantic-based similarity, correlation similarity and cosine similarity.

Next, we compare our semantic-based correlation similarity (in short, “sbc*idf–sbc”) and the semantic-based similarity (in short, “tf–idf–sbs”). Again, we use three figures to show the superiority of our *scor* method in both datasets in Fig. 14.

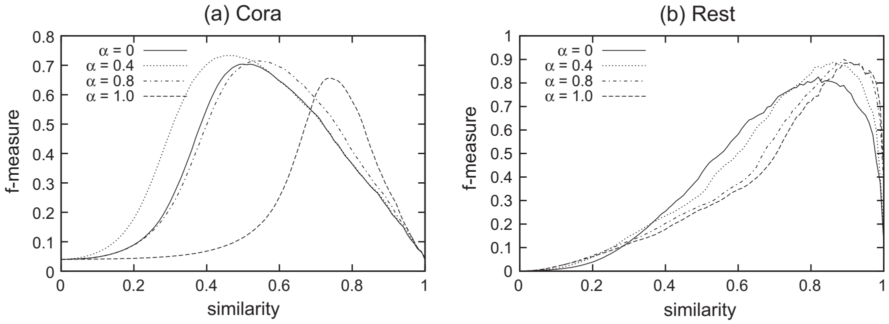

Fig. 12. α Change.

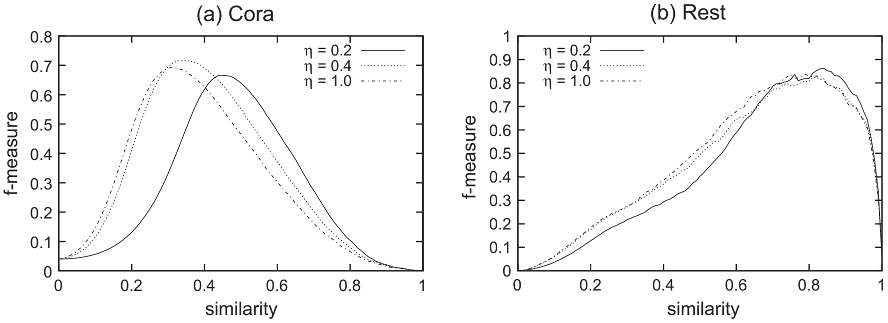

Fig. 13. η change.

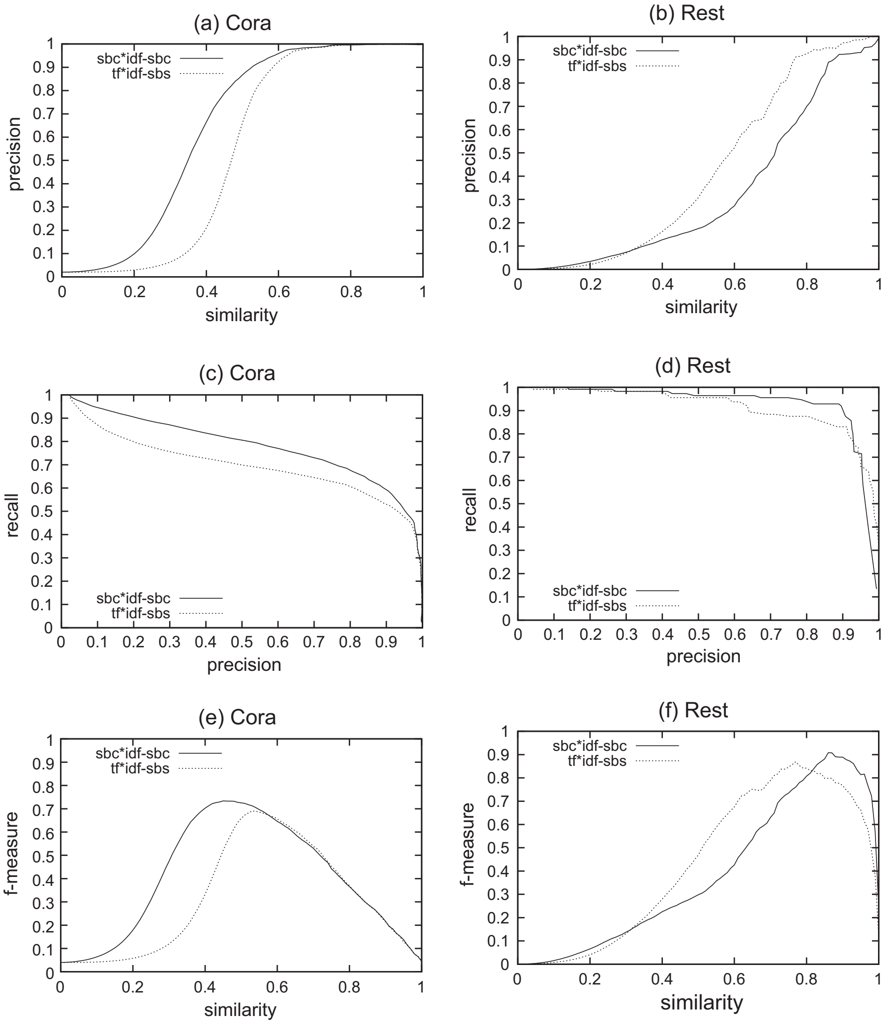

Fig. 14. Scor and sbs compare.

In both dataset, *scor* method gives a higher largest *f*-measure than *sbs* method.

Data missing rate. Finally, we evaluate the performance of these two similarity measures under different data missing rates. Fig. 15 reports the highest *f*-measure score of each method under several missing rates. We introduce errors by randomly attaching letters to words, similar to the previous experiment settings.

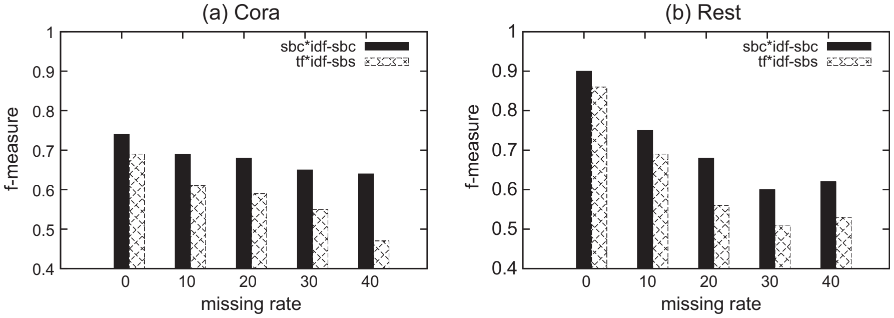

Fig. 15. Missing rate.

In *Cora*, the accuracy of *scor* method drops much slower than the *sbs* method. Meanwhile, in *Restaurant*, although the accuracies of both methods decrease significantly with the increase of missing rate, *scor* always performs better than *sbs* in all data missing rates.

All the experiments about semantic-based methods show that our semantic-based correlation performs better than the semantic-based similarity, especially when data contains many errors.

7. Related work

In this paper, we concentrate on the similarity measure on unstructured text records in databases and concern several issues in this application, i.e., the length of the record is always short and the information in such a short record are limited.

One solution for addressing the problem is to conduct data cleaning and formatting first, by segmenting unstructured text records into structured entities with certain attributes [3] or recovering missing values [30]. Then, the similarity evaluation can be performed on records with certain attributes rather than unstructured data, which has been studied for decades [8,10]. Correlations between multi-attributes of records are considered. For the entity extraction approach, text segmentation [3] is adopted as a word token based entity extraction from unstructured data. Probabilistic representing models [11], including per row model, one row model, and multiple row model, are also developed. Moreover, external dictionary [6] is exploited to improve entity extraction. Xi et al. [31] use a unified relationship matrix to represent and detect latent relationships among heterogeneous data objects of multi-domains.

Another alternative approach is to perform similarity measures directly on text records, by utilizing full text retrieval techniques. Cohen [4] proposes a word token based cosine similarity with *tf**idf* which can detect the similarity of records with various word order and data missing. Gravano et al. [9] propose a more effective approach by using *q*-grams, which can handle spelling errors. The correlations of tokens are ignored in these kinds of vector space model [24] based approaches. Furthermore, learnable similarity metrics have been investigated in recent studies. Jin et al. [15] propose a supervised term weighting scheme by considering the correlation between the word frequency and category information of documents. Bilenko et al. [2] compute the comparison similarity vector of two records and classify the vector as similar or not with a similarity value output. Sarawagi et.al. [25] propose an active learning approach by picking up the most uncertain data which will be labeled manually.

Compared with the above related work, our probabilistic correlation-based approach performs the similarity evaluation on the records without segmenting them into certain attributes and considers the probabilistic correlation of tokens rather than among attributes.

Moreover, Hofmann [13,14] proposes the probabilistic latent semantic analysis (PLSA), which also addresses the problem of different words with a similar meaning or the same concept. The aspect model is used as a latent class of posterior probabilities to compute the joint probability over documents and word tokens. However, the probabilistic latent semantic analysis as an extension of latent semantic analysis LSA [7] is a type of dimension reduction technique. Rather than removing the tokens, our correlation-based similarity enriches the correlation between tokens and finds the correlations between tokens without any class knowledge. In fact, we can still apply our similarity measure after dimension reduction operations, such as PLSA.

The retrieval of small text snippets as the sentence level is also studied by learning from certain training data. Li and Croft [17,16] learn sentence level information patterns from the training data to identify potential answers. Murdock and Croft [20] conduct the sentence retrieval as translations from the query to the results. A parallel corpus have to be exploited for training the translation model. Without requiring training data, usually domain specific, our proposed method seeks the correlations inside databases for general purposes.

The semantic-based similarity, such as [18,22], employs *WordNet* to measure the similarity between two words. The main idea is that the shorter the path of two words in taxonomy is, the more similar they are. Indeed, these semantic-based methods are similar to our correlation when exploring relationships between two words. The major difference is that we explore the correlations inside the dataset, without any external sources such as *WordNet*.

8. Conclusions

In this paper, we propose a novel similarity measure for text records based on the probabilistic correlation of tokens. We define the probabilistic correlation between two word tokens as the probability that these tokens appear in the same records. Then we merge words with high correlations into phrase and extend the correlation between phrase tokens. A feature weighting scheme is performed based on the intra-correlation of tokens in a record. Furthermore, we develop a correlation-based similarity function, which is also based on the token probabilistic correlation. Rather than the dot product of two records in the cosine similarity function, we consider the inter-correlation of tokens in two records in our correlation similarity function. Indeed, we show that the proposed correlation measure is a generalization of the existing cosine similarity.

In the analysis, we show that our probabilistic correlation-based similarity measure is effective in dealing with various information formats such as abbreviation and data missing. Furthermore, the extensive experimental results also verify that our approach achieves higher accuracy than that of the cosine similarity on measuring the similarity of text records.

Acknowledgment

This work is supported in part by China NSFC Grant No. 61202008, 61232018, 61370055, Hong Kong RGC/NSFC Project No. N HKUST637/13, National Grand Fundamental Research 973 Program of China Grant No. 2012-CB316200, Microsoft Research Asia Gift Grant, Google Faculty Award 2013.

References

[1] E. Agichtein, S. Sarawagi, Scalable information extraction and integration, in: Tutorial of KDD'06, 2006.
[2] M. Bilenko, R.J. Mooney, Adaptive duplicate detection using learnable string similarity measures, in: KDD'03, 2003, pp. 39–48.
[3] V. Borkar, K. Deshmukh, S. Sarawagi, Automatic segmentation of text into structured records, in: SIGMOD'01, 2001, pp. 175–186.
[4] W.W. Cohen, Integration of heterogeneous databases without common domains using queries based on textual similarity, in: SIGMOD'98, 1998, pp. 201–212.
[5] W.W. Cohen, P. Ravikumar, S.E. Fienberg, A comparison of string distance metrics for name-matching tasks, in: IJCAI'03 Workshop IIWeb'03, 2003, pp. 73–78.
[6] W.W. Cohen and S. Sarawagi, Exploiting dictionaries in named entity extraction: combining semi-markov extraction processes and data integration methods, in: KDD'04, 2004, pp. 89–98.
[7] S.C. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, R.A. Harshman, Indexing by latent semantic analysis, J. Am. Soc. Inf. Sci. 41 (6) (1990) 391–407.
[8] A.K. Elmagarmid, P.G. Ipeirotis, V.S. Verykios, Duplicate record detection: a survey, IEEE Trans. Knowl. Data Eng. 19 (1) (2007) 1–16.
[9] L. Gravano, P.G. Ipeirotis, N. Koudas, D. Srivastava, Text joins in an rdbms for web data integration, in: WWW'03, 2003, pp. 90–101.
[10] S. Guha, N. Koudas, A. Marathe, D. Srivastava, Merging the results of approximate match operations, in: VLDB'04, 2004, pp. 636–647.
[11] R. Gupta, S. Sarawagi, Creating probabilistic databases from information extraction models. in: VLDB'06, 2006, pp. 965–976.
[12] G.R. Hjaltason, H. Samet, Index-driven similarity search in metric spaces (survey article), ACM Trans. Database Syst. 28 (4) (2003) 517–580.
[13] T. Hofmann, Probabilistic latent semantic analysis, in: Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99), 1999, pp. 289–29.
[14] T. Hofmann, Probabilistic latent semantic indexing, in: SIGIR'99, 1999, pp. 50–57.
[15] R. Jin, J.Y. Chai, L. Si, Learn to weight terms in information retrieval using category information, in: ICML'05, 2005, pp. 353–360.
[16] X. Li, W.B. Croft, Novelty detection based on sentence level patterns, in: CIKM, 2005, pp. 744–751.
[17] X. Li, W.B. Croft, Improving novelty detection for general topics using sentence level information patterns, in: CIKM, 2006, pp. 238–247.
[18] D. Lin, An information-theoretic definition of similarity, in: ICML, vol. 98, 1998, pp. 296–304.
[19] A. McCallum, K. Nigam, L.H. Ungar, Efficient clustering of high-dimensional data sets with application to reference matching, in: KDD'00, 2000, pp. 169–178.
[20] V. Murdock, W.B. Croft, A translation model for sentence retrieval, in: HLT/EMNLP, 2005.
[21] G. Navarro, A guided tour to approximate string matching, ACM Comput. Surv. 33 (1) (2001) 31–88.
[22] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy, Proceedings of the 14th International Joint Conference on Artificial Intelligence, vol. 1, Morgan Kaufmann Publishers Inc., 1995, pp. 448–453.
[23] S. Robertson, Understanding inverse document frequency: on theoretical argument for idf, J. Doc. 60 (5) (2004) 503–520.
[24] G. Salton, Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer, Addison-Wesley, 1989.
[25] S. Sarawagi, A. Bhamidipaty, Interactive deduplication using active learning, in: KDD'02, 2002, pp. 269–278.
[26] S. Song, L. Chen, Probabilistic correlation-based similarity measure of unstructured records, in: CIKM, ACM, 2007, pp. 967–970.
[27] K. Sparck Jones, Index term weighting, Inf. Storage Ret. 9 (11) (1973) 619–633.
[28] S. Tejada, C. Knoblock, S. Minton, Learning domain independent string transformation weights for highaccuracy object identification, in: KDD'02, 2002, pp. 350–359.
[29] C.J. van Rijsbergen, Information Retrieval, Butterworth-Heinemann, Newton, MA, USA, 1979.
[30] J. Wang, S. Song, X. Zhu, X. Lin, Efficient recovery of missing events, PVLDB (2013).
[31] W. Xi, E.A. Fox, W. Fan, B. Zhang, Z. Chen, J. Yan, D. Zhuang, Simfusion: measuring similarity using unified relationship matrix, in: SIGIR'05, 2005, pp. 130–137.