1. Introduction

Text clustering is an important processing task in indexing, retrieval, management and mining of abundant text data on the Web or in corporate information systems. Text clustering determines the intrinsic grouping information and partitions content relevant texts together according to similarity measures.

In the view of technology, text clustering is an unsupervised and automatic procedure of grouping text documents into clusters[1]. Clustering algorithms group a collection of texts into natural clusters. Instances in the same cluster are similar to each other and share certain properties. The results of clustering are solely based on text representation, similarity measures and clustering algorithms.

Text clustering tasks[2] with large vocabularies[3] and dispersive datasets[4] will extremely compromise the performance of clustering algorithms. Therefore, the common technique is feature selection from reduced feature dimensions[5,6]. The absence of labels for guiding the clustering process is a prominent difficulty of feature selection in unsupervised text clustering. It is necessary to extract useful and typical features from high dimensional feature space with consideration of the clustering algorithm performance.

The classical clustering algorithms include the agglomerative method[7] and the Ward's method[8,9]. In current stage, the typical clustering algorithms applied (1) statistics and probability, e.g. word term frequency[10], word meaning frequency[11], and itemset frequency[12]; (2) lexical and semantics analysis[13,14]; and (3) even more complicated methods, e.g. ontologies[15], fuzzy clustering[16,17], self-organizing maps[2,18] and knowledge-based vector space[19]. Recent research works[15,16,17,18,19] testify that the performance of a text clustering system can be improved with computation using conceptual, semantic and knowledge-based features.

On the one hand, the purpose of text clustering is to put similar text documents together efficiently to meet human interest in information searching and understanding. Therefore, it is essential to integrate human cognition features into the clustering procedure to present the cognitive process of text understanding or comprehension, which is one of the motivations of this research work.

An important function and property of the human cognitive system is the ability to extract important information out of textually described situations, which plays a vital role in human understanding. When humans read and comprehend a text or document, they try to build up a situation model[20,21] or mental model[22], which describes the state of affairs in human minds. The theory of situation models, connecting many aspects of cognitive philosophy, linguistics and artificial intelligence, are the focal points in this research work.

On the other hand, a well-known challenge in text clustering is handling of text data in large volume, high dimensionality and complex semantics. If we attempt applying situation models to represent texts, the hierarchical clustering task is reformulated to a task of looking for a shortest "path" connecting all the texts in a given set (please refer to Section 3 for details).

As to the choice of computation algorithm, population-based optimization algorithms, such as genetic algorithm (GA)[23] and ant colony optimization (ACO)[24], have attracted a lot of attention among so many methods proposed for combinatorial optimization problems. These methods attempt to achieve better solutions by application of knowledge from previous iterations.

The genetic algorithms (GA) are stochastic global search methods, which provide a means in data mining, especially in searching poorly understood and irregular spaces. The ant colony optimization (ACO) has been proposed as a meta-heuristic approach for solving hard combinatorial optimization problems[25,26]. A typical example that an ant system is apt to is the traveling salesman problem (TSP).

The basis mechanism is that ants exploit on their ground a substance called pheromone, while walking from food sources to the nest and vice versa. Ants can smell pheromone substance and, when choosing their way, they tend to choose, in probability, paths marked by strong pheromone concentrations. An important and interesting behavior of ant colonies appears to be their foraging behavior. In particular, ants are capable of finding the shortest paths between food sources and their nest without using visual cues. It has been proven in experiments that a

colony of ants can find shortest path employing this pheromone trail with following behaviors[24].

The computational results of Aghdam and colleagues[27] indicate that, (a) ACO algorithm achieves good enough solutions in a reasonable amount of computation time, and outperforms GA in most (9 out of 11) testing categories in the task of text feature selection.

This paper introduces a novel and effective system, CogTCA (Cognitive Text Clustering with Ants), a research effort for text clustering. Inspired by cognitive situation models, CogTCA represents texts according to four cognitive situation dimensions in form of cognitive situation matrices and vectors rather than canonical sparse matrices of high dimensions, proposes several new similarity measures among texts, and implements a text clustering task as solving a combinatorial optimization problem (to find a shortest "path" connecting all the texts in a given set) using the encounter ant colony system (E-ACS).

The rest of the paper is organized as follows. Section 2 introduces the cognitive situation models and inherent dimensions selected for text clustering. Section 3 solicits the method of converting text clustering into a traveling salesman problem. Section 4 introduces the structure of CogTCA and relevant processing details. Section 5 presents experimental results and performance evaluation. In final, Section 6 concludes this paper with concise remarks.

2. Cognitive Situation Models and Dimensions

The human cognitive system has the vital ability to extract important information from the textual information. After the information is extracted, how does human represent and use it for understanding? The answer is situation models[21], which represent the mental activities of human understanding and comprehension.

Many tasks based on language processing, including text clustering, are rationally annotated with the situation models[28]. According to most researches in this area, a typical situation model consists of five different dimensions (Temporality, Spatiality, Protagonist, Causality and Intentionality[21]), which refer to different information sources. When new information concerning any of the five dimensions is extracted, the situation model is updated according to the new information. The situation model is very useful for comprehension of texts or single sentences.

In addition, the combination and comprehension of several texts and sentences can be much better explained by the theory of situation models. For examples, situation models can integrate information across sentences; the explanation of similarities in comprehension performances across modalities can only be implemented with situation models; situation models have strong influence for effects of domain expertise on comprehension[29]; situation models can explain the cognitive procedure[30] of human multiple source learning.

Obviously, an important feature about situation models is the multidimensionality. To be specific, in a sentence, "Temporality" is the temporal information; "Spatiality" is the spatial information; "Protagonist" is the subject or a noun phrase (NP) that plays the role of subject, which might involve anaphoric inference with previous sentences; "Causality" is the causal connection between text elements of sentences when situation changes; "Intentionality" is the intentional connections among the goals of protagonists.

After examine above five dimensions, we conclude that (a) "Temporality", "Spatiality" and "Protagonist" can be extracted at the syntactic level, (b) "Causality" and "Intentionality" can be perceived via analysis of predicates in sentences, as predicates take the main responsibility of information delivery in traditional English. Therefore, "Activity" is recruited as a complementary dimension to present predicates, the necessary "doing" information, in text clustering tasks. In the view of practice, the cognitive situation dimensions include Temporality, Spatiality, Protagonist, and Activity, which are implemented in CogTCA to represent texts in a concise manner.

3. Restate the computation procedure for text clustering task

Fig.1(a) presents two clusters, A1 and B1, in a two dimensional space for demonstration purpose. Fig.1(b) presents that, after A1 exchanges the closest point with B1, A1 becomes A2 and B1 turns into B2. Let's define

$A1 = \{a_1, a_2, \ldots, a_m\}$, $B1 = \{b_1, b_2, \ldots, b_n\}$, $m \in \boldsymbol{N}$, $n \in \boldsymbol{N}$

After exchange a pair of points, we get

$A2 = \{a_1, a_2, \ldots, a_x, \ldots, a_m\}$, $B2 = \{b_1, b_2, \ldots, b_x, \ldots, b_n\}$,

$$m \in \boldsymbol{N}, n \in \boldsymbol{N}$$

a_x and b_x are the exchange points. This exchange operation illustrates that either A2 or B2 includes a heterogeneous point, which reduces the purity of either cluster before exchange. In order to measure the closeness and compactness of a cluster, we propose a parameter as follows to calculate the average distance between any pair of points in the cluster.

Definition 1. Define the average cluster internal distance (ACID) for clusterA1 in Eq. (1).

$$\mathrm{ACID(A1)}=\frac{1}{C_m^2}\sum_{1\le j<k\le m}\left\|a_j-a_k\right\|,$$
$$(C_m^2=\frac{m!}{2!(m-2)!}=\frac{m\cdot(m-1)}{2}) \quad (1)$$

Here, C_m^2 is the number of *2*-combinations in the *m*-element set (A1 = $\{a_1, a_2, ..., a_m\}$). The total number of internal distances for any pair of points equals the combination of *m* points taken *2* at a time without repetitions. $\left\|a_j-a_k\right\|$ is the Euclidean distance between a_j and a_k in a multi-dimension space. If a_j and a_k fall in a *z* dimension space, $\left\|a_j-a_k\right\|$ is presented in Eq. (2).

$$Euclid(a_j, a_k)_z = \left\|a_j-a_k\right\|_z=\sqrt{\sum_{i=1}^{z}\left(a_{ji}-a_{ki}\right)^2} \quad (2)$$

The condition for sum up, $1\le j<k\le m$, ensures that the Euclidean distance for a pair of points is used only once for calculating ACID. For example, as $\left\|a_j-a_k\right\|$ is equivalent to $\left\|a_k-a_j\right\|$, $\left\|a_k-a_j\right\|$ is redundant and excluded from Eq. (1). From definition 1, it is not difficult to prove (see Appendix A.) that, ACID(A1) < ACID(A2) and ACID(B1) < ACID(B2), which means that, once a cluster encloses an impure element (impurity), its average cluster internal distance will increase in a notable manner.

Definition 2. Define the internal open route (IOR) as a sequence containing each point once and only once from a finite cluster A1 = $\{a_1, a_2, ..., a_m\}$. Based on the knowledge of combinatorics, the number of IORs for cluster A1 equals the number of such *m*-permutations of *m* as denoted in Eq. (3a)

$$P_m^m = m! \quad (3a)$$

An example IOR of A1, the sequence of "a_1-a_2- …-a_m", is composed with (*m*-1) edges, "a_1a_2", "a_2a_3", …, "$a_{m-1}a_m$". The length of internal open route (LIOR) for current sequence is calculated with Eq. (3b).

$$\mathrm{LIOR(A1)}_1 = \mathrm{LIOR}(\text{"}a_1\text{-}a_2\text{- …-}a_m\text{"}) = Euclid(a_1, a_2) + Euclid(a_2, a_3)+\ldots+ Euclid(a_{m-1}, a_m) \quad (3b)$$

The shortest internal open route (SIOR) is denoted as the minimal value of these *m*! LIORs.

$$\mathrm{SIOR(A1)} = \min_{1\le k\le m!}\{\mathrm{LIOR(A1)}_k\} \quad (3c)$$

Based on definitions 1 and 2, the text clustering task is restated as (1) dividing all texts (points) of set **S** in Fig.2(a) into several processed clusters (S1, S2, …, Sn) in Fig.2(b) according to designated or intrinsic dimensions ("Temporality", "Spatiality", "Protagonist", "Causality", "Intentionality" and "Activity"), (2) searching the final optimal solution that satisfies following conditions as (3d).

When the average cluster internal distance (ACID) for cluster Si (i=1,2, …,n) approaches its minimum, which indicates the most compact state of Si, the length of internal open route (LIOR) for current cluster Si simultaneously turns into the shortest internal open route (SIOR) of Si.

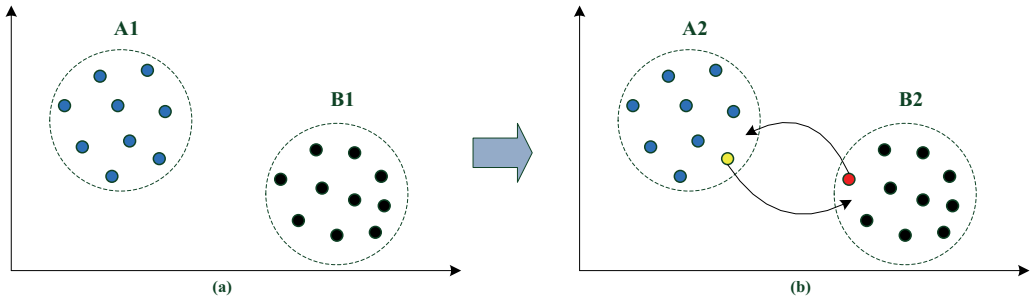


Fig.1. Analysis of cluster internal distances

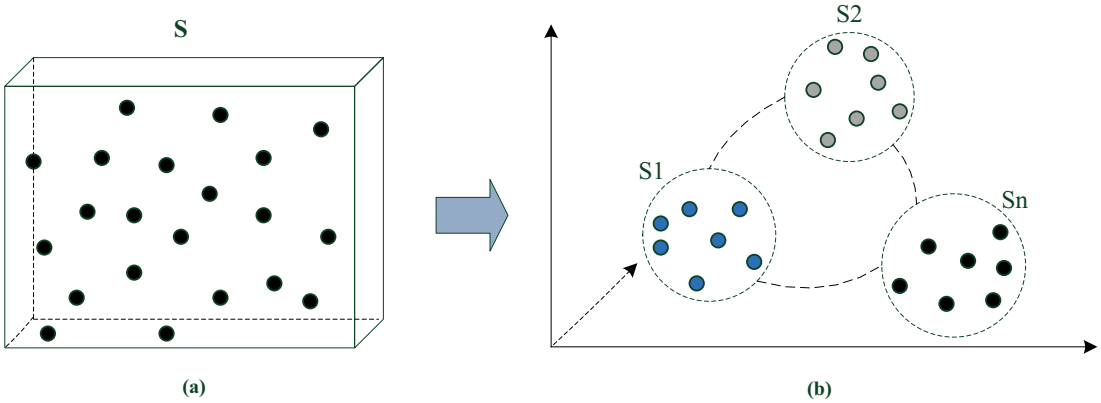


Fig.2. Clustering States. (a) original state; (b) final optimal state.

$$\begin{aligned} &\text{ACID(S1)} \rightarrow \textit{minimum} \\ &\text{ACID(S2)} \rightarrow \textit{minimum} \\ &\cdots\ \cdots \\ &\text{ACID(Sn)} \rightarrow \textit{minimum} \end{aligned} \quad \Rightarrow \quad \begin{aligned} &\text{LIOR(S1)} \rightarrow \text{SIOR(S1)} \\ &\text{LIOR(S2)} \rightarrow \text{SIOR(S2)} \\ &\cdots\ \cdots \\ &\text{LIOR(Sn)} \rightarrow \text{SIOR(Sn)} \\ &\\ &\text{LIOR(}\mathbf{S}\text{)} \rightarrow \text{SIOR(}\mathbf{S}\text{)} \end{aligned} \tag{3d}$$

4. Structure of CogTCA

4.1 Text representation

CogTCA is composed with seven modules (in Fig. 3).The phase of text representation includes modules "Sentence Parsing", "Extraction of Cognitive Situation Dimensions", "Construction of Cognitive Situation Vectors" and "Construction of Cognitive Situation Matrices". In order to extract the four cognitive situation dimensions ("Protagonist", "Temporality", "Spatiality", and "Activity" described in Section 3) in an accurate manner, the cognitive interactionist sentence parser[30] is applied to parse each sentence and obtain a corresponding syntactic structure, which consists of a set of labeled links connecting pairs of words, and a constituent-tree containing conventional constituents (e.g., noun phrases, verb phrases, and prepositional phrases).

In the module of "Extraction of Cognitive Situation Dimensions", WordNet[31] is referred to facilitate semantic analysis. As a useful tool for computational linguistics and natural language processing, WordNet groups nouns, verbs, adjectives and adverbs into distinct sets of cognitive synonyms (synsets), which are interlinked with conceptual-semantic and lexical relations.

Temporality is the temporal information in each sentence. In most conditions, the temporal information contains a time-relevant noun phrase. Spatiality should satisfy two constraints: a noun phrase and location-relevance. Activity, normally identified as a verb phrase or predicate, is the dominant part of a sentence. Identification of activities relies on the extraction of verb phrases from constituent-trees. Protagonists also rely on the noun phrases, appearing mostly as subjects and partially as objects, and anaphoric inference is an unavoidable language phenomenon. The corresponding extraction algorithms are based on our previous work[32].

A notable capability of human cognitive systems is to extract the most dominant information from textual contents. Therefore, human beings can represent the context in multiple dimensions (including central concepts in theories of situated cognition) and monitor situational

information for understanding and comprehension. To simulate above cognitive features, each text is presented with a set of situation vectors, which are composed of four dimensions extracted from each sentence. A cognitive situation vector is defined in Eq. (4).

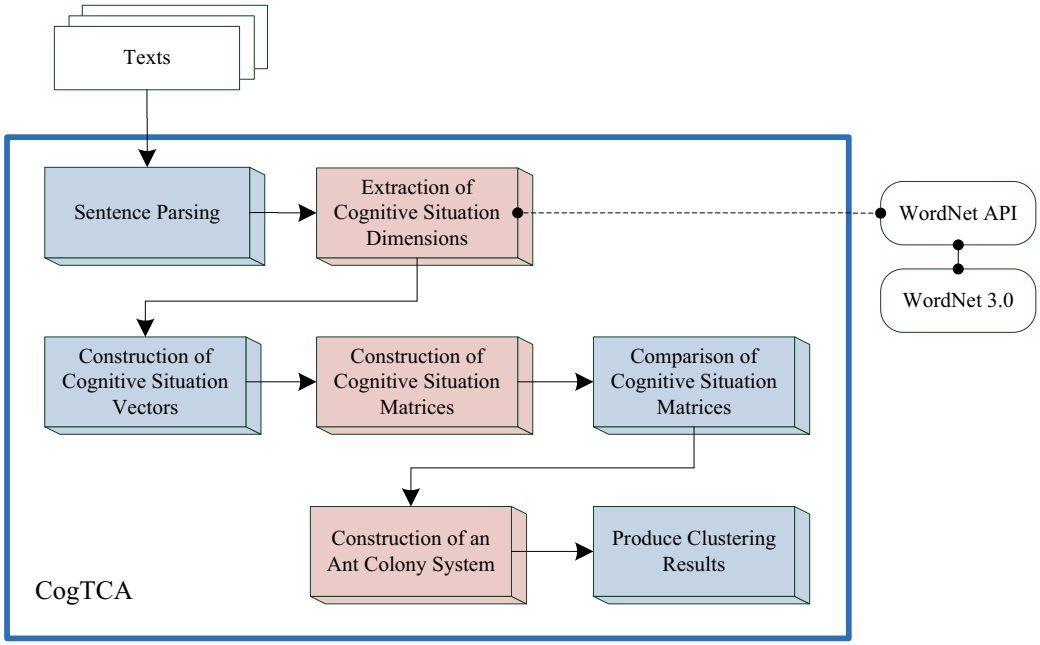


Fig.3. Structure of CogTCA

Definition 3. Define a cognitive situation vector (CSV) as

$$\boldsymbol{SV}_i = (P_i,\ A_i,\ T_i,\ S_i), \quad i \in (1,2,...m) \qquad (4)$$

m is the total number of sentences in a text, i is the sequence of a sentence in a text.
For the i^{th} sentence,
$\boldsymbol{SV}_i$ presents corresponding situation vector,
P_i is the subject/object information, Protagonist,
A_i is the Activity information,
T_i is the Temporality information,
S_i is the Spatiality information.

Based on definition 3, a text is represented with a cognitive situation matrix (CSM) in Eq. (5). A CSM (M_p) is filled with a set of situation vectors ($SV_{p,i}$), each of which is composed with cognitive situation elements, e.g. words, phrases, and clauses.

Definition 4. Define a cognitive situation matrix (CSM) as

$$M_p = (SV_{p,1}, SV_{p,2}, \ldots, SV_{p,i}, \ldots, SV_{p,m})^T,\ 1 \le i \le m,\ 1 \le p \le k \qquad (5)$$

m is the total number of sentences in a text,
k is total number of texts in collection $\boldsymbol{D}$,
p is a subscript to identify different texts in a collection,
T presents the transpose operation.

4.2 Similarity of cognitive situation matrices

In order to compare any pair of texts, we need to define and calculate the correlation between their CSMs.

Definition 5. If the CSMs of any two texts are represented with M_p and M_q as Eq.s (6) and (7), the correlation between M_p and M_q is defined as $Cor(M_p, M_q^T)$ and calculated with Eq. (8).

$$M_p = (SV_{p,1},\ SV_{p,2},\ \ldots,\ SV_{p,i},\ \ldots,\ SV_{p,m})^T \qquad (6)$$

$$M_q = (SV_{q,1},\ SV_{q,2},\ \ldots,\ SV_{q,j},\ \ldots,\ SV_{q,n})^T \qquad (7)$$

$$1 \le i \le m,\ 1 \le j \le n,\ 1 \le p, q \le k$$

$$Cor(M_p, M_q^T) = Cor\left(\begin{pmatrix} SV_{p,1} \\ SV_{p,2} \\ \vdots \\ SV_{p,m} \end{pmatrix}, \begin{pmatrix} SV_{q,1} & SV_{q,2} & \cdots & SV_{q,n} \end{pmatrix}\right) =$$

$$\begin{pmatrix} Cor(SV_{p,1},SV_{q,1}) & Cor(SV_{p,1},SV_{q,2}) & \cdots & Cor(SV_{p,1},SV_{q,n}) \\ Cor(SV_{p,2},SV_{q,1}) & Cor(SV_{p,2},SV_{q,2}) & \cdots & Cor(SV_{p,2},SV_{q,n}) \\ \vdots & \vdots & \ddots & \vdots \\ Cor(SV_{p,m},SV_{q,1}) & Cor(SV_{p,m},SV_{q,2}) & \cdots & Cor(SV_{p,m},SV_{q,n}) \end{pmatrix} \quad (8)$$

In Eq. (8), the correlation between any pair of CSVs is expressed and calculated in Eq. (9), which heavily relies on the semantic relations upon these cognitive dimensions.

$$Cor(SV_{p,i}, SV_{q,j}) = w_P \cdot Sem\,(P_{p,i}, P_{q,j}) + w_A \cdot Sem\,(A_{p,i}, A_{q,j}) + w_T \cdot Sem\,(T_{p,i}, T_{q,j}) + w_S \cdot Sem\,(S_{p,i}, S_{q,j}) \quad (9)$$

w_P, w_A, w_T and w_S present the weights for dimension Protagonist, Activity, Temporality and Spatiality, and satisfy following conditions: $0< w_P, w_A, w_T, w_S<1$, and $w_P+w_A+w_T+w_S=1$. The four weights are equally initialized with empiristic value 0.25, and also dynamically adjusted at the end of each main loop during the computing procedure (please refer to Algorithm-1 (3h)).

In Eq. (9), $Sem(P_{p,i},P_{q,j})$ present the semantic product between $P_{p,i}$ and $P_{q,j}$, which indicates whether $P_{p,i}$ and $P_{q,j}$ are semantically equal or not. The semantic product of $P_{p,i}$ and $P_{q,j}$ is described in Eq. (10). If and only if $P_{p,i}$ and $P_{q,j}$ are synonyms or phrases of same meanings, they are treated lexically or semantically equal.

$$Sem(P_{p,i},P_{q,j}) = 1, \text{ when } P_{p,i} \text{ and } P_{q,j} \text{ are lexically or semantically equal}$$
$$Sem(P_{p,i},P_{q,j}) = 0, \text{ otherwise} \quad (10)$$

Eq. (10) also applies to the dimensions of Activity ($A_{p,i}$, $A_{q,j}$), Temporality ($T_{p,i}$, $T_{q,j}$), and Spatiality ($S_{p,i}$, $S_{q,j}$).

In Eq. (8), if we use C_{ij} ($1\le i \le m$, $1\le j \le n$) to replace $Cor(SV_{p,i}, SV_{q,j})$, Eq. (8) can be denoted as Eq. (8^A) in a simple form.

$$Cor(M_p, M_q^T) = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1n} \\ C_{21} & C_{22} & \cdots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{m1} & C_{m2} & \cdots & C_{mn} \end{pmatrix} \quad (8^A)$$

Definition 6. To measure the closeness between any pair of texts using their CSMs (M_p and M_q), a new parameter *Similarity* is computed with Eq. (11) based on above correlation matrix in Eq. (8^A).

$$Similarity(M_p, M_q) = \frac{1}{m} \cdot \left(\sum_{i=1}^{m} (\max_{1\le j\le n} \{C_{ij}\}) \right) \quad (11)$$

where $\max_{1\le j\le n} \{C_{ij}\}$ is the maximum value in the i^{th} row in $Cor(M_p, M_q^T)$. In another word, when comparing the i^{th} sentence of M_p with each sentence of M_q, only the i^{th} sentence of M_p and its most similar sentence of M_q are decisive elements for the similarity. The similarity is the average of maximum values for each row in matrix $Cor(M_p, M_q^T)$. As $C_{ij} \in [0,1]$, $\max_{1\le j\le n} \{C_{ij}\} \in [0,1]$. Therefore, the value of *Similarity* falls in the range of [0, 1] as well.

Definition 7. Define another parameter *CogDist* to describe the cognitive semantic distance between M_p and M_q as follows.

$$CogDist(M_p, M_q) = [Similarity(M_p, M_q)]^{-1} = Similarity^{-1}(M_p, M_q) \quad (12)$$

The value of a *CogDist* is inversely proportional to the *Similarity*, which means that the more similar two texts are, the closer they are in the cognitive semantic distance. As $Similarity(M_p, M_q) \in [0,1]$, $CogDist(M_p, M_q) \in [1, \infty)$.

4.3 Construction of an ant colony system for clustering

Based on definitions 1, 2 and 7, the best solution for the clustering a text collection ***D*** into appropriate clusters (D_1, D_2, …, D_n) is interpreted as searching for the shortest internal open route (SIOR) of ***D***, which connects all points (texts). Meanwhile, each cluster D_i ($1\le i\le n$) reaches its minimum value $ACID_{min}(D_i)$.

$$ACID(D_i) = \frac{2}{m \cdot (m-1)} \sum_{0<j<k<m} CogDist(M_j, M_k) \rightarrow minimum \quad (13)$$

m is the number of texts in D_i, , M_j and M_k are the CSMs for any pair of text in D_i.

In this condition, SIOR(D_i) and SIOR($\boldsymbol{D}$) reach the state to produce the shortest internal open routes

$$\text{SIOR}(\boldsymbol{D}) = (\sum_{0<i<n} \text{SIOR}(D_i)) + \textit{BridgeSum} \quad (14)$$

BridgeSum is the sum-up of cognitive distances in the shortest internal open route (SIOR) connecting clusters (D_1, D_2, …, D_n) together.

So far, clustering texts is transformed into a combinatorial optimization problem, suitable for approaching with an ant colony optimization system[24]. Traditional ant colony algorithm has advantages in computation convergence and limitations in computing speed and achieving diverse solutions[25].

To avoid these limitations, based on the encounter phenomena of ants, this paper implements an improved ant colony system, the encounter ant colony system (E-ACS) constructed as follows. Its core optimization algorithm is presented in Algorithm-1 (pseudo code). In E-ACS, when an ant has visited more than half texts, it will try to exchange "information" with another ant. If the sum-up of visited texts exceeds the total text number, replica texts will be eliminated to update a new list of visited texts and the pheromone trail of the edges visited by the two ants.

Suppose that m artificial ants are assigned within n texts. An artificial ant k (at text i) chooses the text j to move to among those which do not belong to its working memory N_k by applying the following probabilistic formula:

$$p_{ij}^k(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}}{\sum_{u \in N_k} \tau_{iu}^{\alpha}(t)\eta_{iu}^{\beta}} & j \notin N_k \\ 0 & otherwise \end{cases} \quad (15)$$

$p_{ij}^k(t)$ is the probability with which ant k chooses to move from text i to text j. $\tau_{ij}(t)$ is the amount of pheromone trail on edge (i, j) at time t, η_{ij} is a heuristic function, which was chosen to be the inverse of the cognitive semantic distance between texts i and j, α is a parameter which weighs the relative importance of pheromone trail and of closeness. β is a parameter which weighs the relative importance of the heuristic function. This formula favors those edges which are shorter and have a higher level of pheromone trail.

The pheromone trail is changed both locally and globally. Global updating is intended to reward edges belonging to shorter tours. As soon as all artificial ants have completed their tours in n texts, the best ant deposits pheromone on its visited edges. The other edges remain unchanged.

In the global trail updating formula as follows,

$$\tau_{ij}(t+n) \leftarrow (1-\rho)\tau_{ij}(t) + \rho\sum_{k=1}^{m}\Delta\tau_{ij}^k \quad (16)$$

the amount of pheromone $\sum_{k=1}^{m}\Delta\tau_{ij}^k$ deposited on each visited edge (i, j) by the best ant is inversely proportional to the length of the tour: the shorter the tour, the greater the amount of pheromone deposited on edges, which can be presented in Eq. (17), if T_L is defined as the length of the tour. ρ is the parameter for pheromone reinforcement, and $(1-\rho)$ is the parameter for pheromone evaporation. This manner of depositing pheromone is intended to emulate the property of differential pheromone trail accumulation, which in the case of real ants was due to the interplay between the length of the path and continuity of time.

$$T_L \propto (\sum_{k=1}^{m}\Delta\tau_{ij}^k)^{-1},$$

$$\text{when} \sum_{k=1}^{m}\Delta\tau_{ij}^k \to maximum,\ T_L \to minimum \quad (17)$$

$$\Delta\tau_{ij}^k = \begin{cases} \dfrac{Q}{L_k} & \text{If ant k visits the edge } (i, j) \text{ in this loop cycle} \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

where Q is a computation constant to mark the initial strength of pheromone when an ant starts a new loop, and L_k is the total distance that ant k visits in a loop. In this paper, the empiristic value of Q is 36.

Algorithm-1: The encounter ant colony system (E-ACS)

Begin

(1) Initialize and prepare computing parameters (texts, ant number, *CogDist*, N_k, α , β , $\tau_{ij}(0)$, M, ρ , w_P, w_A, w_T, w_S, …);

(2) FOR i = 1, 2, …, M /* main loop for iteration */

(3) FOR each ant /* Eq. (15)-(18) are used */
(3a) Building up its own route
(3b) IF (half texts have been visited)
(3c) THEN Loop Start:
(3d) IF ((the total number of texts visited by any two ants) > (the total text number))
THEN Label the two ants as "encounter ants";
Eliminate replica texts;
Build a new list of visited texts for the two ants;
Update the pheromone trail of the edges visited by the two ants;
ELSE continue loop
END IF
(3e) IF ((the number of encounter ants) > (the threshold $\mathcal{V}$))
THEN Build an encounter route;
Update the pheromone trail of the edges in the encounter route;
End and Exit
ELSE continue loop
END IF
(3f) Visit a new text
(3g) Loop End:
END IF
END FOR
(3h) Adjust w_P, w_A, w_T, w_S
END FOR /* end of main loop for iteration */
End

4.4 Produce clustering results

The previous computing procedure builds up a shortest internal open route connecting all texts together. The length of any edge presents the cognitive semantic distance between texts on both ends. A top-down splitting operation is required to produce the final clustering result, which depends on the required number of clusters or the threshold of diameter of each cluster. Two different splitting algorithms, number-based (NBSA) and diameter-based (DBSA), are presented as follows. In NBSA, saying $\boldsymbol{N} = 20$, which states that the target texts are required to be divided into 20 clusters. The more clusters required, the result is more fine-grained. In DBSA, the diameter threshold D_{thres} has direct impact on the granularity of clustering results. The shorter the diameter threshold D_{thres} is, the more fine-grained clusters we have. NBSA is more efficient and decisive (or arbitrary sometimes), while DBSA is more practical in control of clustering granularity as implemented in this paper.

Algorithm-2: The number-based splitting algorithm (NBSA)

Begin
(1) Set up the cluster number $\boldsymbol{N}$;
/* $\boldsymbol{N}$ is a pre-assigned number of required clusters for the splitting result */
(2) Search for the longest ($\boldsymbol{N}$-1) edges;
(3) Cut off these ($\boldsymbol{N}$-1) edges;
(4) Produce each segment as a cluster.
End

Algorithm-3: The diameter-based splitting algorithm (DBSA)

Begin
(1) Define the diameter as the longest cognitive semantic distance between any pair of texts in a cluster;
(2) Set up the diameter threshold D_{thres};
/* D_{thres} is a pre-assigned value to determine the granularity of the splitting result */
(3) Present the shortest internal open route as $\{d_1, d_2, \ldots, d_n\}$, n is the total number of texts, d_1 and d_n are the start and end points;
(4) Initialize the cluster counter $k = 1$;
(5) FOR (i=1; i<n+1; i++)
Put d_i in queue $Q_k \rightarrow \{d_1, d_2, \ldots, d_i\}$
IF ($\max_{d_p, d_q \in Q_k} CogDist(d_p, d_q) > D_{thres}$)
THEN $Q_k = \{ d_1, d_2, \ldots, d_{i-1} \}$;
$k = k + 1$;
$Q_k = \{d_i\}$
END IF
END FOR
(6) The final clusters $\rightarrow \{Q_1, Q_2, \ldots, Q_k\}$
End

5. Experiments and evaluation

5.1 Implementation and text corpus

CogTCA is developed with Perl/Java in Windows® 7 Ultimate / Fedora 14 using a personal computer (Dell Precision T5500, Intel® Xeon® Quard Core (E5620) 2.4GHz, 12GB DDR3 RDIMM, 1333MHz). The following experiments are also implemented in the same hardware and software environment.

We use Reuters-21578[33] and RCV1-v2[34] in our experiments. Reuters-21578, currently the most widely used test collection for text processing research, includes 21578 tagged documents and 669 indexed categories spanning across 9 various domains (shown in table 2). RCV1-v2 contains 35 times as many newswire stories (804,414 for RCV1-v2) as the popular Reuters-21578 collection. These texts are organized in four hierarchical groups (top level topic categories): CCAT (Corporate/Industrial), ECAT (Economics), GCAT (Government/Social), and MCAT (Markets). RCV1-v2 provides benchmark data on all categories, which include: (1) 103 Topic categories—101 with one or more positive training examples on the training set and all 103 (including all the 101) have one or more positive test examples on the test set, (2) 354 Industry categories—313 with positive training examples, and 350 (including all of the 313) with positive test examples, and (3) 366 Region categories—228 with positive training examples and 296 (including all of the 228) with positive test examples. The standard clusters are constructed on the basis of the category label for each text.

5.2 Evaluation methods

Three metrics, *Purity*, *F-Score* and normalized mutual information (*NMI*)[35], shown in Eq. (19)-(21), are used to measure the clustering performance. *Purity* is a simple measure, in the range of [0, 1], to compute the proportion of the documents which are correctly clustered. *F-Score* is a multiple evaluation method that combines recall and precision measures. A perfect clustering solution will be the one in which every cluster has a corresponding cluster containing the exactly same documents in the resulting hierarchical tree, in which case the *F-Score* will be one. In general, the higher the *F-Score* values, the better the clustering solution is. The *NMI* value is 1 when clustering results perfectly match the external category labels and close to 0 for a random partitioning. *NMI* is a better measure than entropy which is biased towards high *K* solutions[36,37].

$$Purity = \sum_{k=1}^{K}\left(\frac{1}{n}\cdot\max_{1\le j\le C}\{n_{jk}\}\right) \tag{19}$$

$$F\text{-}Score = \sum_{j=1}^{C}\left(\frac{n_j}{n}\max_{1\le k\le K}\left\{\frac{2\cdot\frac{n_{jk}}{n_j}\cdot\frac{n_{jk}}{n_k}}{\frac{n_{jk}}{n_j}+\frac{n_{jk}}{n_k}}\right\}\right) \tag{20}$$

$$NMI = \frac{\sum_{j,k}\left[n_{jk}\log\left(\frac{n\cdot n_{jk}}{n_j\cdot n_k}\right)\right]}{\sqrt{\left[\sum_j\left(n_j\log\frac{n_j}{n}\right)\right]\left[\sum_k\left(n_k\log\frac{n_k}{n}\right)\right]}} \tag{21}$$

In these equations, relevant parameters are defined as follows. For an experimental set $\boldsymbol{D}$, texts are labeled in C classes, each being noted as L_j ($1\le j\le C$). $\boldsymbol{D}$ is clustered into K clusters, each being noted as U_k ($1\le k\le K$). n is the number of texts in $\boldsymbol{D}$; n_j is the number of texts in Lj; n_k is the number of texts in U_k; n_{jk} is the number of mutual texts for L_j and U_k.

5.3 Experimental tracks and result analysis

Three reference clustering models, representing different technical strategies, are selected to evaluate CogTCA. These models are frequent term-based text clustering model (FT)[10], a conceptional self- organizing map model (ConSOM, impact factor equals 0.6)[18] and knowledge-based vector space model (KBVSM, based on Hirst and St-Onge's ontology [38]) [19].

To examine the performance of CogTCA with above three references our experiments are designed to implement in four tracks (shown in table 1):

(1) Clustering with single domains (CSD) uses seven domains (shown in table 2) of Reuters-21578. In each domain, 120-150 texts are selected randomly to conduct the test for each domain 12 times. The average results is shown in Fig.4(*a*)-(*c*).

(2) Clustering across domains (CAD) includes 10 scheduled experiments (identified with roman numbers: I,

II, III, …, X) in each of which 300 texts are selected in random from the categories across 9 domains of Reuters-21578. The experimental results are displayed in Fig.4(*d*)-(*f*). The "Ave" columns indicate averages for each algorithm.

(3) Clustering with single groups (CSG) uses four hierarchical groups (CCAT, ECAT, GCAT and MCAT) of RCV1-v2. In each group, 20 thousand texts are selected randomly to conduct the test for each group 10 times. The average results is shown in Fig.5(*a*)-(*c*).

(4) Clustering across groups (CAG) includes 10 scheduled experiments (identified with roman numbers: I, II, III, …, X) in each of which 25 thousand texts are selected in random from RCV1-v2. The experimental results are displayed in Fig.5(*d*)-(*f*). The "Ave" columns indicate averages for each algorithm.

The experimental result of track (1), Fig.4(*a*)-(*c*), presents that, for each algorithm in test, its performance varies on different domains, but is still not strongly domain-dependent or domain-sensitive. FT achieves lower scores than the other three (KBVSM, ConSOM and CogTCA), which could be due to FT heavily relies on term frequency, without involving other language factors, lexical or semantic. KBVSM and ConSOM are peer to peer on the measure of *Purity* in Fig.4(*a*), so are ConSOM and CogTCA on the measure of *NMI* in Fig.4(*c*). Overall, CogTCA scores higher than the other three reference systems on three standard measure parameters, *Purity*, *F-Score* and *NMI*.

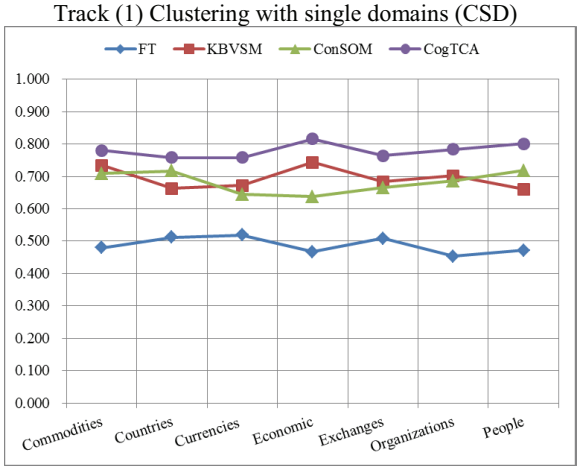
Table 1. List of experimental tracks

Track ID and name	Target corpus	Number of texts for each test	Notes
(1) CSD	Reuters-21578	120-150	All the test texts are randomly selected within a single domain, and very likely come from various categories
(2) CAD	Reuters-21578	300	All the test texts are randomly selected without constraints of domains, and definitely come from various categories.
(3) CSG	RCV1-v2	20,000	All the test texts are randomly selected within a single group (CCAT, ECAT, GCAT or MCAT) of RCV1-v2.
(4) CAG	RCV1-v2	25,000	All the test texts are randomly selected across the boundaries of groups (CCAT, ECAT, GCAT and MCAT) of RCV1-v2.

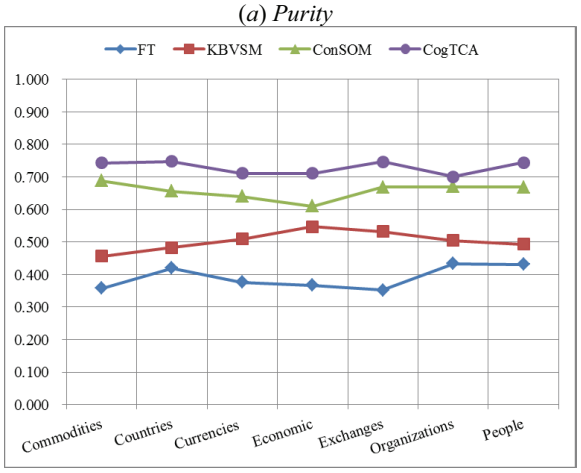
Table 2. Statistic of Categories of Reuters-21578 used in experiments

Codes for Categories (*Domains*)	*Numbers of Categories*
Commodities	78
Countries	175
Currencies	27
Economic Indicators	16
Exchanges	39
Organizations	56
People	267

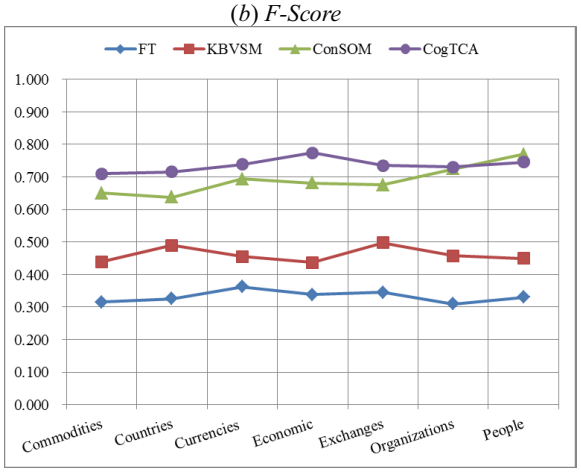
Track (1) Clustering with single domains (CSD)



(*a*) *Purity*

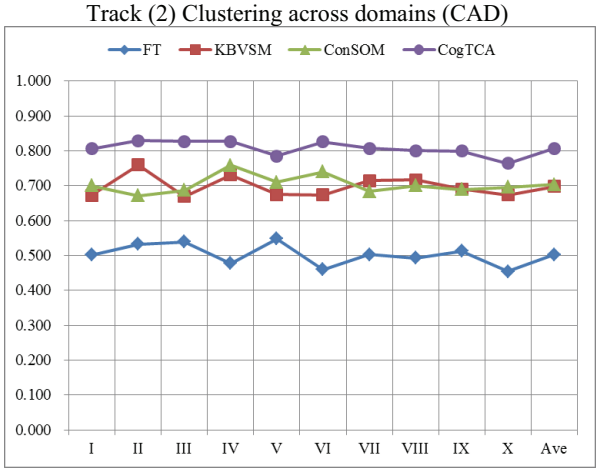


(*b*) *F-Score*

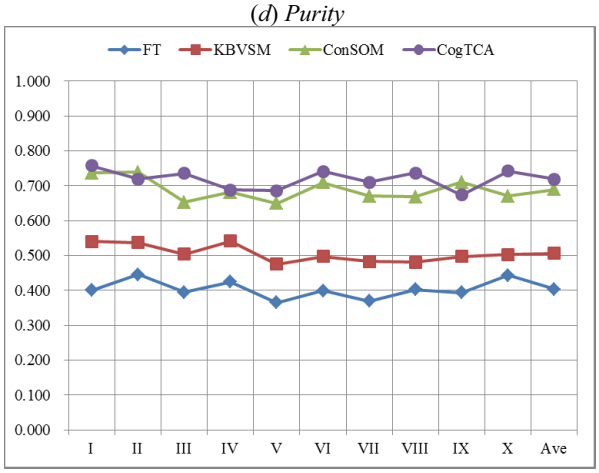


(*c*) *NMI*

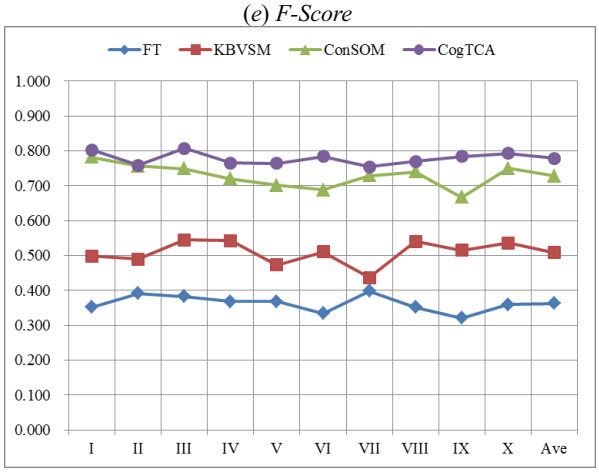
Track (2) Clustering across domains (CAD)



(*d*) *Purity*



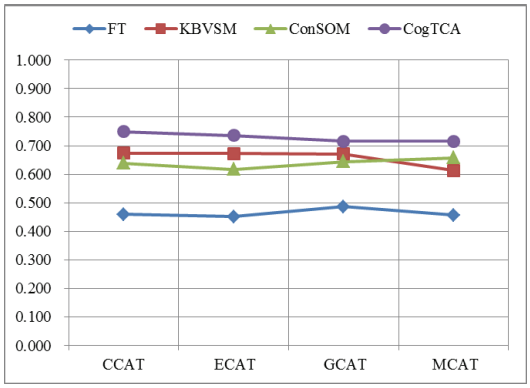
(*e*) *F-Score*



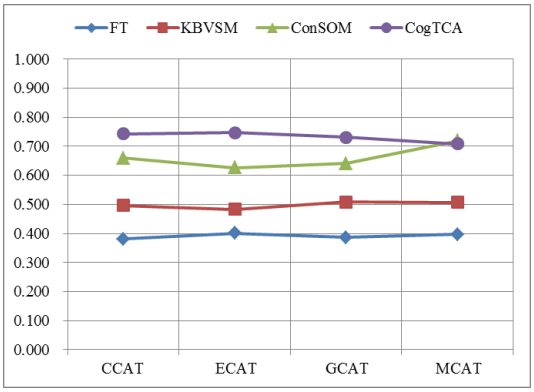
(*f*) *NMI*

Fig.4. Experimental results of CSD and CAD

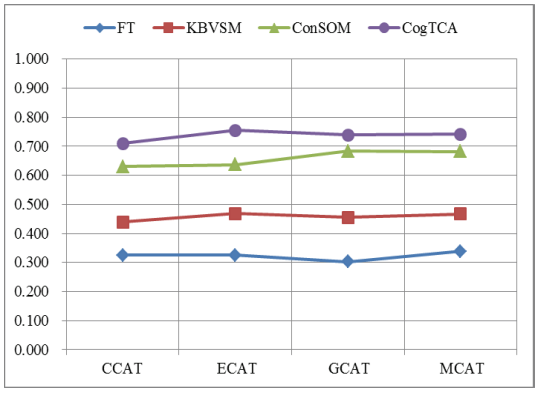
Track (3) Clustering with single groups (CSG)



(*a*) *Purity*

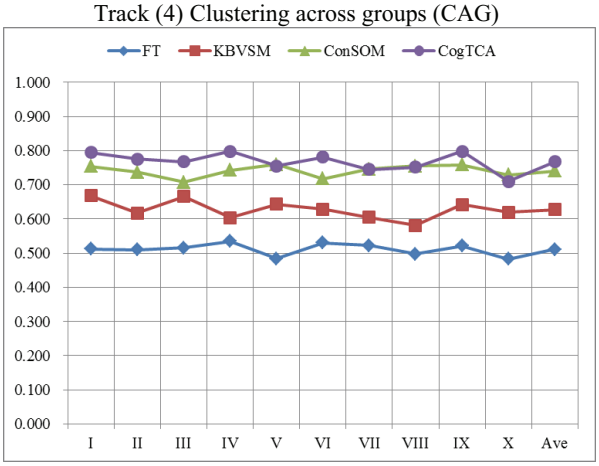


(*b*) *F-Score*

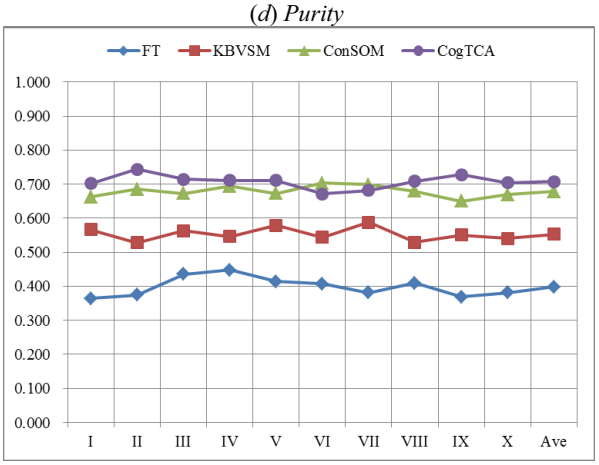


(*c*) *NMI*

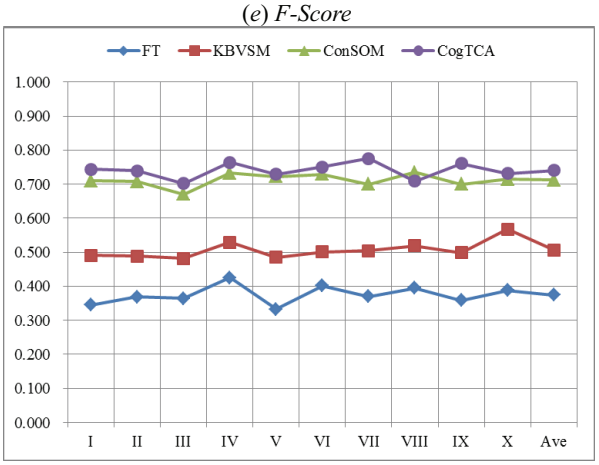
Track (4) Clustering across groups (CAG)



(*d*) *Purity*



(*e*) *F-Score*



(*f*) *NMI*

Fig.5. Experimental results of CSG and CAG

The experimental result of track (2), Fig.4(*d*)-(*f*), states that, when processing mixed texts from various domains, FT still plays no better than any of other three algorithms (KBVSM, ConSOM and CogTCA). KBVSM and ConSOM still perform equally well on the measure of *Purity* in Fig.4(*d*), which might be brought on by application of knowledge bases in form of ontologies. ConSOM and CogTCA are even matched, while the latter one performs slightly better, on both measures of *F-Score* and *NMI* in Fig.4(*e*)-(*f*). Note that ConSOM takes an impact factor of 0.6, which has been proven as one of the best two impact factors (the other is 0.7). When comparing (*a*, *b*, *c*) and (*d*, *e*, *f*) in Fig.4, we see that all the four algorithms achieve better evaluation score in track (2). This phenomenon is believed due to the fact that the content deviations of single-domain sets are more prominent than those of multiple-domain sets.

The experimental result of track (3), Fig.5(*a*)-(*c*), indicates that, in a much larger testing corpus, four algorithms show no salience of group (or topic) sensitivity. FT remains its previous status, and KBVSM and ConSOM entangle with each other still on measure of *Purity*. ConSOM acts slightly inferior to CogTCA, except its upswing at the group of MCAT on the measure of *F-Score*.

The experimental result of track (4), Fig.5(*d*)-(*f*), reveals that, when processing much more mixed texts from various groups (top level topics). ConSOM and CogTCA are notable competitive to each other on all three measures. FT and KBVSM remain their rankings as expected. When comparing (*a*, *b*, *c*) and (*d*, *e*, *f*) in Fig.5, once again, we see that all the four algorithms are of better performance when processing "mixed" texts. This phenomenon confirms that it is of more challenge to divide similar texts into subsections of finer granularity.

"Extraction of Cognitive Situation Dimensions" (ECSD), a prerequisite processing module shown as Fig.3, plays an indispensable role of extracting four cognitive situation dimensions ("Protagonist", "Temporality", "Spatiality", and "Activity") for subsequent modules in CogTCA. The time cost of this module has direct impact on the whole clustering system. Therefore, the elapsed time (ET) is recorded as a supplementary index for each experiment when the four tracks are implemented. The average elapsed time of ECSD, shown as Fig.6, is calculated on the basis of per hundred thousand (10^5) words according to domains (Reuters-21578) or groups (RCV1-v2).

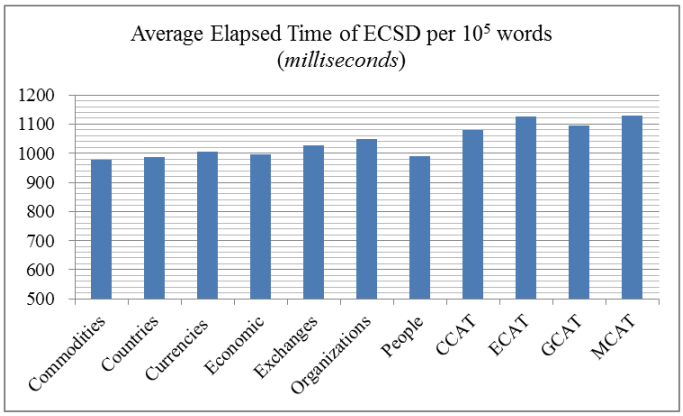


Fig.6 Average elapsed time of ECSD in four experimental tracks. (a) The elapsed time is not sensitive to domains or groups, ranging from 979ms to 1128ms. (b) Texts of RCV1-v2 costs slightly more time in average than those of Reuters-21578.

6. Concluding remarks

This paper emphasizes the necessity of integrating human cognitive features into text processing procedures and advantages of artificial ant systems in solving hard combinatorial optimization problems, and presents the feasibility of converting a text clustering task into a shortest path searching problem.

Subsequently, this paper introduces a new text clustering

system, CogTCA, which represents a text in the form of cognitive situation matrix and searches for the shortest internal open routes with an optimized approach, the encounter ant colony system, within a set of texts. The final clustering result is delivered in a top-down splitting manner based on either number or diameter of expected clusters.

CogTCA condenses the cognitive situation models into four dimensions, which makes the computation with cognitive features applicable and practical. CogTCA processes texts at syntactical and semantic levels and complements research works heavily relying on statistics and probability.

CogTCA is examined with elaborately designed experimental tracks involving a broad range of sufficient texts of Reuters-21578 and RCV1-v2. The experimental results have testified the performance and effectiveness of CogTCA. The future research work will focus on the elevation of metrics in performance evaluation and the domain sensitivity of texts. Meanwhile, text processing always faces linguistic uncertainty in lexicology and semantics. The integration of solutions for linguistic uncertainty (Entropy, Fuzziness and Ambiguity[39]) into E-ACS is another potential research interest in no time.

Appendix A. Analysis of cluster internal distances

For clusters A1=$\{a_1, a_2, \ldots, a_m\}$ and B1=$\{b_1, b_2, \ldots, b_n\}$, suppose all the nodes in A1 and B1 satisfy:

$$\left\|a_j - a_k\right\| < \left\|b_i - a_k\right\|, \quad 1 \le j < k \le m, \quad 1 \le i \le n$$

If exchange a_p with b_q, A1→A2=$\{a_1, a_2, \ldots, a_{p-1}, b_q, a_{p+1}, \ldots, a_m\}$

$$\sum_{1\le j<k\le m}\left\|a_j - a_k\right\| = \sum_{1\le j<k}\sum_{j<k\le m}\left\|a_j - a_k\right\| = \sum_{\substack{1\le j\le p-1\\ j<k}}\sum_{j<k\le m}\left\|a_j - a_k\right\| + \sum_{p+1\le j<k}\sum_{j<k\le m}\left\|a_j - a_k\right\| + \sum_{p<k\le m}\left\|a_p - a_k\right\|$$

$$< \sum_{\substack{1\le j\le p-1\\ j<k}}\sum_{j<k\le m}\left\|a_j - a_k\right\| + \sum_{p+1\le j<k}\sum_{j<k\le m}\left\|a_j - a_k\right\| + \sum_{p<k\le m}\left\|b_q - a_k\right\|$$

If multiply a constant $\frac{1}{C_m^2}$ to either side of above inequation, we get

$$\frac{1}{C_m^2}\sum_{1\le j<k\le m}\left\|a_j - a_k\right\| < \frac{1}{C_m^2}\{\sum_{\substack{1\le j\le p-1\\ j<k}}\sum_{j<k\le m}\left\|a_j - a_k\right\| + \sum_{p+1\le j<k}\sum_{j<k\le m}\left\|a_j - a_k\right\| + \sum_{p<k\le m}\left\|b_q - a_k\right\|\}$$

which also means that

$$\text{ACID(A1)} < \text{ACID(A2)}$$

Similarly,

$$\text{ACID(B1)} < \text{ACID(B2)}$$

QED.

Appendix B. The numerical results of experimental tracks (1)-(4)

Track (1) Clustering with single domains (CSD)

Categories	FT	KBVSM	ConSOM	CogTCA
Commodities	0.479	0.734	0.708	0.780
Countries	0.512	0.662	0.717	0.758
Currencies	0.519	0.671	0.644	0.759
Economic	0.467	0.743	0.638	0.815
Exchanges	0.509	0.684	0.665	0.765
Organizations	0.453	0.702	0.686	0.784
People	0.471	0.661	0.718	0.801

(*a*) *Purity*

Categories	FT	KBVSM	ConSOM	CogTCA
Commodities	0.358	0.456	0.688	0.743
Countries	0.420	0.483	0.655	0.747
Currencies	0.376	0.509	0.640	0.711
Economic	0.366	0.547	0.610	0.711
Exchanges	0.351	0.532	0.668	0.747
Organizations	0.434	0.504	0.670	0.700
People	0.431	0.493	0.668	0.745

(*b*) *F-Score*

Categories	FT	KBVSM	ConSOM	CogTCA
Commodities	0.315	0.440	0.650	0.709
Countries	0.326	0.490	0.637	0.715
Currencies	0.363	0.455	0.693	0.738
Economic	0.338	0.437	0.681	0.774
Exchanges	0.346	0.498	0.675	0.735
Organizations	0.309	0.458	0.725	0.731
People	0.331	0.450	0.770	0.746

(*c*) *NMI*

Track (2) Clustering across domains (CAD)

Experiment IDs	FT	KBVSM	ConSOM	CogTCA
I	0.502	0.673	0.700	0.806
II	0.533	0.760	0.671	0.829
III	0.539	0.669	0.687	0.827
IV	0.477	0.731	0.758	0.827
V	0.548	0.675	0.709	0.785
VI	0.459	0.674	0.740	0.826
VII	0.503	0.714	0.683	0.806
VIII	0.493	0.717	0.700	0.801
IX	0.512	0.690	0.689	0.799
X	0.454	0.674	0.695	0.764
Ave	0.502	0.698	0.703	0.807

(*d*) *Purity*

Experiment IDs	FT	KBVSM	ConSOM	CogTCA
I	0.400	0.541	0.736	0.757
II	0.445	0.537	0.739	0.719
III	0.394	0.504	0.653	0.736
IV	0.424	0.542	0.681	0.688
V	0.365	0.476	0.649	0.686
VI	0.399	0.497	0.709	0.741
VII	0.369	0.483	0.670	0.710
VIII	0.402	0.481	0.668	0.737
IX	0.394	0.497	0.709	0.673
X	0.443	0.502	0.671	0.742
Ave	0.404	0.506	0.689	0.719

(*e*) *F-Score*

Experiment IDs	FT	KBVSM	ConSOM	CogTCA
I	0.352	0.498	0.782	0.803
II	0.391	0.490	0.756	0.758
III	0.383	0.544	0.749	0.807
IV	0.368	0.543	0.720	0.765
V	0.368	0.473	0.701	0.763
VI	0.333	0.511	0.687	0.784
VII	0.398	0.436	0.728	0.754
VIII	0.351	0.540	0.740	0.770
IX	0.320	0.514	0.667	0.784
X	0.360	0.536	0.749	0.793
Ave	0.362	0.508	0.728	0.778

(*f*) *NMI*

Fig.B.1 Experimental results of CSD and CAD

Track (3) Clustering with single groups (CSG)

Codes for Categories	FT	KBVSM	ConSOM	CogTCA
CCAT	0.459	0.673	0.638	0.749
ECAT	0.451	0.672	0.617	0.736
GCAT	0.486	0.671	0.644	0.716
MCAT	0.457	0.613	0.658	0.715

(*a*) *Purity*

Codes for Categories	FT	KBVSM	ConSOM	CogTCA
CCAT	0.381	0.496	0.659	0.743
ECAT	0.401	0.483	0.626	0.747
GCAT	0.386	0.509	0.640	0.731
MCAT	0.396	0.507	0.719	0.708

(*b*) *F-Score*

Codes for Categories	FT	KBVSM	ConSOM	CogTCA
CCAT	0.325	0.440	0.630	0.709
ECAT	0.326	0.469	0.637	0.755
GCAT	0.303	0.455	0.683	0.738
MCAT	0.338	0.466	0.681	0.741

(*c*) *NMI*

Track (4) Clustering across groups (CAG)

Experiment IDs	FT	KBVSM	ConSOM	CogTCA
I	0.513	0.668	0.754	0.794
II	0.510	0.617	0.737	0.775
III	0.516	0.666	0.708	0.767
IV	0.535	0.603	0.742	0.798
V	0.484	0.644	0.760	0.754
VI	0.530	0.628	0.718	0.781
VII	0.523	0.604	0.746	0.745
VIII	0.497	0.581	0.755	0.752
IX	0.522	0.642	0.759	0.798
X	0.483	0.620	0.729	0.710
Ave	0.511	0.627	0.741	0.767

(*d*) *Purity*

Experiment IDs	FT	KBVSM	ConSOM	CogTCA
I	0.364	0.566	0.662	0.702
II	0.375	0.528	0.685	0.743
III	0.436	0.563	0.672	0.715
IV	0.447	0.546	0.694	0.711
V	0.414	0.579	0.672	0.710
VI	0.408	0.544	0.704	0.671
VII	0.381	0.588	0.698	0.681
VIII	0.409	0.529	0.679	0.708
IX	0.369	0.550	0.650	0.727
X	0.381	0.540	0.669	0.704
Ave	0.398	0.553	0.678	0.707

(*e*) *F-Score*

Experiment IDs	FT	KBVSM	ConSOM	CogTCA
I	0.345	0.491	0.710	0.744
II	0.368	0.489	0.708	0.739
III	0.364	0.481	0.670	0.701
IV	0.425	0.529	0.733	0.764
V	0.333	0.485	0.722	0.729
VI	0.401	0.501	0.729	0.751
VII	0.370	0.504	0.700	0.775
VIII	0.394	0.520	0.735	0.709
IX	0.359	0.499	0.700	0.760
X	0.388	0.568	0.714	0.732
Ave	0.375	0.507	0.712	0.740

(*f*) *NMI*

Fig.B.2 Experimental results of CSG and CAG

References

1. B.C.M. Fung, K. Wang, M. Ester, Hierarchical document clustering, in: John Wang (Ed.), *The Encyclopedia of Data Warehousing and Mining* (Idea Group, 2005).
2. A.K. Jain, R.C. Dubes, *Algorithms for Clustering Data*, (Prentice Hall, Englewood Cliffs, NJ, 1988).
3. P. Bradley, U. Fayyad, C. Reina, Clustering very large database using EM mixture models, in: *Proceedings of 15th Intern. Conf. on Pattern Recognition* (ICPR-2000), 2000, pp.76-80.
4. K. Nigam, A.K. McCallum, S. Thrun, T.M. Mitchel, Text classification from labeled and unlabeled documents using EM, *Machine Learning*, **39**(2/3) (2000) 103-134.
5. T. Hofmann, Probabilistic latent semantic analysis, in: *Proceedings of Uncertainty in Artificial Intelligence* (UAI'99), 1999, pp.289-296.
6. M.H.C. Law, M.A.T. Figueiredo, A.K. Jain, Simultaneous feature selection and clustering using mixture models, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **26**(9) (2004) 1154-1166.
7. E.M. Voorhees, Implementing agglomerative hierarchical clustering algorithms for use in document retrieval, *Information Processing and Management*, ELSEVIER, **22** (6) (1986) 465-476.
8. A. El-Hamdouchi, P. Willett, Hierarchical document clustering using ward's method, In: *Proceedings of ACM Conference on Research and Development in Information Retrieval*, 1986, pp. 149-156.
9. D.G. Roussinov, H. Chen, Document clustering for electronic meetings: an experimental comparison of two techniques, *Decision Support Systems*, ELSEVIER, **27**(1-2) (1999) 67-79.
10. F. Beil, M. Ester, X. Xu, Frequent term-based text clustering, in: *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM Press, 2002, pp.436-442.
11. Y. Li, S.M. Chung, J.D. Holt, Text document clustering based on frequent word meaning sequences, *Data & Knowledge Engineering*, ELSEVIER, **64**(1) (2008) 381-404.
12. B.C.M. Fung, K. Wang, M. Ester, Hierarchical document clustering using frequent itemsets, in: *Proceedings of SIAM International Conference on Data Mining*, 2003, pp.156-162.
13. B. Choudhary, P. Bhattacharyya, Text clustering using semantics, in: *Proceedings of the 11th International World Wide Web Conference*, 2002, pp.326-332.
14. J. Sedding, D. Kazakov, WordNet-based text document clustering, in: *Proceedings of COLING-2004 Workshop on Robust Methods in Analysis of Natural Language Data*, 2004, pp.242-247.
15. A. Hotho, S. Staab, G. Stumme, Ontologies improve text document clustering, in: *Proceedings of the 3rd IEEE International Conference on Data Mining*, 2003, pp.541-544.
16. R. Saraçoğlu, K. Tütüncü, N. Allahverdi, A fuzzy clustering approach for finding similar documents using a novel similarity measure, *Expert Systems with Applications*, ELSEVIER, **33**(3) (2007) 600-605.
17. W. Tjhi, L. Chen, Possibilistic fuzzy co-clustering of large document collections, Pattern Recognition, ELSEVIER, 40(12) (2007) 3452-3466.
18. Y. Liu, X. Wang, C. Wu, ConSOM: A conceptional self-organizing map model for text clustering, *Neurocomputing*, ELSEVIER, **71**(4-6) (2008) 857-862.
19. L. Jing , M. K. Ng, J. Z. Huang, Knowledge-based vector space model for text clustering, *Knowledge Information Systems*, **25** (2010) 35-55.
20. T.A. van Dijk, W. Kintsch, *Strategies of discourse comprehension*, (Academic Press, 1983).
21. R.A. Zwaan, J.P. Magliano, A.C. Graesser, Dimensions of situation model construction in narrative comprehension, *Journal of Experimental Psychology, Learning, Memory, and Cognition*, American Psychological Association, **21** (1995) 386-397.
22. P.N. Johnson-Laird, *Mental models: Towards a cognitive science of language, inference, and consciousness*, (Cambridge, MA: Harvard University Press, 1983).
23. M. Srinivas and L. M. Patnik, *Genetic algorithms: A survey*, (Los lamitos: IEEE Computer Society Press, 1994).
24. M. Dorigo, G. Di Caro, The ant colony optimization meta-heuristic. In D. Corne, M. Dorigo, and F. Glover (Eds.), New ideas in optimization, London: McGraw-Hill, 1999, pp.11-32.
25. M. Dorigo, V. Maniezzo, A. Colorni, The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, **26**(1) (1996) 29-41.
26. T. Stützle, M. Dorigo, The ant colony optimization

metaheuristic: algorithms, applications, and advances, In F. Glover and G. Kochenberger (Eds.), *Handbook of metaheuristics* (Norwell, MA: Kluwer Academic Publishers, 2003), pp.251-285.

27. M. H. Aghdam, N. Ghasem-Aghaee, M. E. Basiri, Text feature selection using ant colony optimization, *Expert Systems with Applications*, ELSEVIER, 36 (2009) 6843–6853.
28. R.A. Zwaan G.A. Radvansky, Situation Models in Language Comprehension and Memory, *Psychological Bulletin*, American Psychological Association, **123**(2) (1998) 162-185.
29. W. Schneider, J. Körkel, The knowledge base and text recall: Evidence from a short-term longitudinal study, *Contemporary Educational Psychology*, **14** (1989) 382-393.
30. Y. Guo, Z. Shao, N. Hua, A Cognitive Interactionist Sentence Parser with Simple Recurrent Networks, *Information Sciences*, ELSEVIER, **180**(23), (2010) 4695-4705.
31. G.A. Miller, R. Beckwith, C. Fellbaum, D Gross, K.J. Miller, Introduction to WordNet: An on-line lexical database, *International Journal of Lexicography*, **3**(4) (1990) 235-312.
32. Y. Guo, Z. Shao, N. Hua, Automatic text categorization based on content analysis with cognitive situation models, *Information Sciences*, ELSEVIER, **180** (5), (2010) 613-630.
33. Reuters-21578, Distribution 1.0, http://kdd.ics.uci.edu/ databases/reuters21578/reuters21578.htm l. (2003).
34. D.D. Lewis, Y. Yang, T. Rose, and F. Li, RCV1: A New Benchmark Collection for Text Categorization Research, *Journal of Machine Learning Research*, **5** (2004) 361-397.
35. S. Zhong, J. Ghosh, Generative Model-based Document Clustering: a comparative study, *Knowledge and Information Systems*, SPRINGER, London Ltd., **8**(3) (2005) 374-384. (doi:10.1007/s10115- 004-0194-1)
36. A. Strehl, J. Ghosh, R.J. Mooney, Impact of similarity measures on web-page clustering, In AAAI Workshop on AI for Web Search, 2000, pp.58-64.
37. A. Strehl, J. Ghosh, Cluster ensembles - a knowledge reuse framework for combining partitions, *Journal of Machine Learning Research*, MIT Press, **3** (2002) 583-617.
38. G. Hirst, D. St-Onge, Lexical chains as representations of context for the detection and correction of malapropisms, (Fellbaum, 1998), pp.305-332.
39. X.Z. Wang, L.C. Dong, J.H. Yan, Maximum ambiguity based sample selection in fuzzy decision tree induction, *IEEE Transactions on Knowledge and Data Engineering*, IEEE Computer Society, (2011) 292-300. (doi:10.1109/TKDE.2011.67)