Decision Support Systems 141 (2021) 113465

Contents lists available at ScienceDirect

Decision Support Systems

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

Identifying comparable entities with indirectly associative relations and word embeddings from web search logs

Liye Wang [a], Jin Zhang [a,*], Guoqing Chen [b], Dandan Qiao [c]

[a] Department of Management Science and Engineering, School of Business, Renmin University of China, Beijing 100872, China
[b] China Retail Research Center, School of Economics and Management, Tsinghua University, Beijing 100084, China
[c] Department of Information Systems and Analytics, National University of Singapore, 117417, Singapore

ARTICLE INFO

Keywords:
Comparable entity identification
Web search logs
Indirectly associative relation
Semantic analysis

ABSTRACT

Comparable entity identification plays an essential role in the decision making of both consumers and firms in competitive environment. In contrast to traditional cooccurrence approaches, this paper proposes a novel method, namely, ICE (identifying comparable entities) for effectively identifying comparable entities from web search logs, which are online user-generated contents that reflect users' attention and preferences. ICE consists of two stages: the formulation of directly and indirectly associative relations, followed by a generative procedure that is designed for deriving a broad set of candidate entities that are indirectly associative with a specified focal entity; and a deep-learning-based semantic analysis with a word embedding procedure for measuring the similarities between entity profiles so as to target comparable entities from the candidate set. Extensive experiments show that ICE outperforms several baseline methods in the identification of accurate, broad and novel comparable entities with suitable rankings.

1. Introduction

The comparison of products or services plays a significant role in consumers' purchasing decision-making process, where they often resort to webpages, online reviews, and/or social media to obtain information regarding comparable entities. However, due to the bounded cognitive ability [22] and the information overload [29], consumers cannot effectively access the entire set of comparable entities in an effective manner. More importantly, such a comparison typically requires high-level domain knowledge of consumers. Thus, comparable entity identification, which aims at identifying a comprehensive and accurate set of entities that are comparable to a specified entity, is deemed desirable for helping consumers identify alternative products for consideration in their decision-making process [13].

Comparable entity identification is also vital for businesses in strategic and marketing management. Typically, managers can identify comparable entities by matching similarities and differences of entity categories and characteristics in their minds [22]. Due to limited cognitive abilities, they may only be aware of a small number of comparable entities, and entities that are out of sight will not be considered [3]. Although firms sometimes utilize paid profile services such as Hoovers (www.hoovers.com) and Mergent (www.mergentonline.com) to collect information regarding comparable entities, those services are provided by professionals for specified domains; thus, they may be costly and often suffer from scalability problems [32]. Moreover, such professional-based services cannot reach consumers' minds and fail to examine comparable entities from the perspective of users.

To overcome these limitations, some recent efforts have been made to automatically identify comparable entities or mine comparative relations from online user-generated contents (UGC) [1,29]. For comparable entity identification from the user perspective, extant methods are mainly conditioned on the premise that comparable entities have much higher cooccurrence in the same statements. However, this premise cannot be well applied in various types of UGC, such as web search logs, online product reviews, and tweets, where comparable entities appear less frequently in cooccurrence patterns [30], thereby leading to degraded performance.

To extend the premise, this study proposes a new perspective of comparable entity identification in terms of indirectly associative relation analysis. In various types of UGC, comparable entities not only directly appear in the same statements but also appear in an indirect form. The proposed indirectly associative relation analysis is a useful extension of previous efforts. It can improve consumers' and managers' exploration of various types of UGC and help them capture comparable

* Corresponding author.
E-mail addresses: wangliye@ruc.edu.cn (L. Wang), zhangjin@rmbs.ruc.edu.cn (J. Zhang), chengq@sem.tsinghua.edu.cn (G. Chen), qiaodd@nus.edu.sg (D. Qiao).

https://doi.org/10.1016/j.dss.2020.113465
Received 15 February 2020; Received in revised form 25 November 2020; Accepted 25 November 2020
Available online 3 December 2020

entities that are ignored by extant methods. In consideration of indirectly associative relations, a typical type of UGC, namely, web search logs, is selected as the research data of this study. In web search logs, comparable entities seldom appear in cooccurrence patterns [30]. This type of UGC demands for novel methods for identifying indirectly associative relations, which deviate from the traditional cooccurrence premise.

In this study, a method, namely, ICE (identifying comparable entities) is proposed for identifying entities that are comparable to a specified entity from web search logs. Entities in ICE refer to the objects (e.g., companies, products, and persons) that users care about and then query through search engines [1,23]. Comparable entities, such as BMW and Mercedes-Benz, are entities that share a common utility and meet the similar needs of consumers [29]. In ICE, the specified entity for which comparable entities must be identified by the method is called a focal entity [1,23]. For example, Ford is selected as a focal entity for managers in Ford Motor Company, and Ausnutria will be selected as a focal entity for consumers who want to buy milk powder. Two key issues must be addressed. First, as previously discussed, most comparable entities do not frequently appear concurrently in the same web search logs [30]. Second, due to data noise and short queries in web search logs [6], the accuracy of the identification results depends not only on the cooccurrence positions where entities appear. It is necessary to incorporate an effective semantic analysis between entities into the identification process. Therefore, ICE consists of two stages: the derivation of a broad set of candidate entities that are indirectly associative with the focal entity, which are linked by their related aspects, and the measurement of the similarities between the candidate entities and the focal entity to target comparable entities from the obtained candidate set. Data experiments are conducted to evaluate the performance of ICE in comparison to several baseline methods.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces the proposed method, whose algorithmic details are presented in Section 4. The experimental results are provided in Section 5. Finally, the work is concluded in Section 6.

2. Related work

Since the focus of this study is for identifying comparable entities, the mainstream studies of comparable entity identification are reviewed in Section 2.1. The second stage of ICE is to detect comparable entities from a set of candidate entities identified in the first stage, which is relevant to comparative relation mining. Thus, similar methods of comparative relation mining are reviewed in Section 2.2.

2.1. Comparable entity identification

The mainstream methods for comparable entity identification are based on the direct cooccurrence relation analysis. The basic assumption of these methods is that comparable entities more frequently cooccur in the same comparative sentences, such as “A is better than B” and “A versus B”. Some research efforts focused on identifying comparable entities straightly according to predefined comparative patterns in web pages [1,7]. Jain and Pantel [5] proposed a bootstrapping-based method for identifying comparable entities from web search logs and webpages. Li et al. [13] defined broader patterns and developed a pattern evaluation method for comparable entity identification in comparative questions. Jiang et al. [6] constructed a comparable entity graph based on pre-defined patterns, and then extracted broader comparison patterns and comparable entities in each sub-graph. Ruan et al. [24] defined comparative patterns in tables, lists and free texts of corporate prospectuses for identifying comparable entities. Li et al. [14] extended the cooccurrence patterns based on POS tags by including dependency relations and relative locations between comparable entities and comparison keywords.

The cooccurrence-based methods suffer from two defects. First, they are not always workable when the cooccurrence patterns are scarce or even non-existent [29] and cannot take advantage of indirectly associative relations that exist abundantly in UGC such as web search logs, online reviews, and tweets [29,30]. Second, there exist numerous noises in comparable entities identified merely by cooccurrence positions [5,13], which could affect the performance of these methods. In contrast, this study proposes a new method that identifies comparable entities with indirectly associative relations. The proposed method is an extension of existing cooccurrence-based methods and improves the comprehensiveness of comparable entity identification results. Moreover, in the proposed ICE method, comparable entities are identified based on the semantical meanings of entities. Thus, the results are more accurate compared with those identified by positions.

2.2. Comparative relation mining

Some research efforts accomplished comparative relation mining as a classification task. They utilized features extracted from different sources and applied machine learning models to inferring possible comparative relations among entities. For example, Ma et al. [17] explored structural attributes of the network derived from online news to infer the relation of two companies. Pant and Sheng [21] found that competing firms were likely to have similar web footprints that were represented by online isomorphism. Such online isomorphism can serve as good indicators in comparative relation classification. Choi et al. [2] evaluated the resource similarity and market commonality of two entities based on their structural features in the financial transaction network, by which three types of comparable entities were classified. Generative models are another type of methods used for comparative relation mining. Yang et al. [33] focused on both tweets and patent records and proposed a topical factor graph model to classify relations between companies or products. Tang et al. [26] proposed a dynamic probabilistic model to characterize topical evolution of entities in a patent network where topic-level comparative relations were extracted.

In addition, some studies analyzed comparative relations in terms of competitiveness. Wei et al. [30] proposed a competitiveness analysis method based on the bipartite graph built by conjoint attributes in web search logs. Valkanas et al. [29] defined the competitiveness between two entities based on their coverage on the product attributes and brand characteristics. Some research further analyzed the comparison directions of and the user preferences for comparable entities. Xu et al. [32] proposed a two-level conditional random field model to measure the interdependence between comparative relations in the same sentence and extracted comparison directions of comparative relations. Tkachenko and Lauw [27] developed a generative model for comparative sentences in online reviews to identify comparative directions of two entities and aspects they competed. Liu et al. [15] utilized sentiment analysis to mine competitive advantages of an entity compared to its comparable entities from the social media.

Although related, all the studies discussed above need a pre-given entity set and concentrate more on analyzing comparative relations between entities instead of detecting comparable entities for a particular entity. Moreover, most of the comparative relation mining methods ignore the semantical relations of entities and judge entity relations according to entities’ structural attributes [17,21,30], which will be invalid when the application scenario changes. Though some studies mined entity relations by their topic similarities through generative models [26,33], they are more suitable for professional-generated contents (PGC) like patents and news that are long documents with abundant and frequently-appear words. For the relatively shorter UGC, such as online reviews and web search logs, parameters of these generative model cannot be fully trained, thereby resulting in the poor performance in the UGC. The disadvantages of existing methods highly motivates us to propose a new comparable entity targeting procedure in this study. In the proposed ICE, the candidate entity sets are not pre-given. They are automatically identified through indirectly associative relations at the

Table 1
Sample set of Search Logs from Baidu.com

Index	K	N	Index	K	N
q_1	{BMW, X5}	27,322	q_4	{BMW,SUV, price}	740
q_2	{BMW, Business}	12,524	q_5	{BMW,car}	1330
q_3	{BMW, motor}	6430	q_6	{BMW,4S}	1414

* Note that all examples given in this paper are translated from Chinese web search logs of Baidu.com, otherwise indicated where necessary.

first stage of ICE. Moreover, a new comparable entity targeting procedure is proposed at the second stage of ICE based on the semantic analysis between entities. In the procedure, a concept of entity profiles that is suitable for short UGC is proposed and defined. With entity profiles, short UGC like web search logs can be aggregated into long documents, which could overcome the limitations of PGC-based generative models.

3. The ICE method

In this study, a two-stage method ICE is proposed for identifying comparable entities based on indirectly associative relations from the perspective of consumers. ICE is composed of two stages: the first stage is the discovery of a broad set of candidate entities based on indirectly associative relations of keywords in web search logs, and the second stage is a semantic analysis that is implemented based on keyword and document representations for the detection of comparable entities from the candidate entity set.

3.1. Preliminaries

Let Q be the web search log set within a time period, where a log in the web search log set is denoted as q. A web search log q is either a complete sentence or simply composed of several words. Depending on the language of web search logs, effective text segmentation tools, such as Jieba,[1] NLTK,[2] and Stanford Parser,[3] can be employed to preprocess the web search logs into segmentation units, which are denoted as keywords in this work. After preprocessing, each query log q can be represented as a tuple $q = (K, N)$, where $q.K$ is the set of search keywords in q and $q.N$ is the search volume of q. Table 1 presents a sample set of web search logs that contain "BMW" on January 5, 2018. Like entity "BMW", entities are a subset of keywords in web search logs. In ICE, the time period of the utilized web search logs can be set according to the practical requirements of users. For example, if users want to learn about the competition in the short term, they could set the time period as a week or a month. Otherwise, it can be set as a quarter or six months for a longer-term competition analysis. For an entity e, the web search log set that contains e can be directly downloaded from search engines through their free tools, such as Baidu Tuiguang and Google Adwords. All the obtained search log sets can serve as the web search log data Q used in this study.

Definition 1 (Directly associative relation)
For specified keywords k_a and k_b, if there exists a web search log $q = (K, N)$ in Q such that k_a and k_b cooccur, k_a and k_b are referred to as directly associative. In other words, letting (k_a, k_b) be a tuple that represents the relation between k_a and k_b, k_a and k_b being directly associative means that $(k_a, k_b) \in R^d = \{(k, k') | k, k' \in q.K, \forall q \in Q\}$, where R^d is the directly associative relation set.

Definition 2 (Indirectly associative relation)
For specified keywords k_a and k_c, if there exists another keyword k_b that is directly associative with both k_a and k_c, then k_a and k_c are referred to as indirectly associative. In other words, k_a and k_c being indirectly associative means that $(k_a, k_c) \in R^{ind} = \{(k, k'') | (k, k') \in R^d, (k', k'') \in R^d\}$, where R^{ind} is the indirectly associative relation set.

The keywords that are directly associative with a keyword k can be regarded as related aspects of it. For example, with the keyword "BMW", users may frequently search via queries such as "BMW 4S", "BMW SUV", and "BMW Germany". Keywords "4S", "SUV", and "Germany" could be regarded as related aspects of "BMW" from the perspective of users. Two indirectly associative keywords (for example, "BMW" and "Mercedes-Benz") may compete for consumersâ€™ attention on common aspects (for example, intermediate keywords, "4S" and "SUV"). Whether two entities are comparable is based on whether they compete for the attention of similar users [22]. Comparable entities are often searched together with intermediate keywords, which represent the related aspects that they share. In this regard, indirectly associative relations in web search logs can serve as an important basis for comparable entity identification. Moreover, the directly associative relations and indirectly associative relations between keywords are not mutually exclusive and may have overlaps. For instance, "BMW" and "Mercedes-Benz" can also be directly associative in the the query "BMW Mercedes-Benz". Effective methods have been proposed to eliminate the impact the overlaps of keyword relations, which will be introduced in the following sections.

3.2. Candidate entity generation

In the first stage of ICE, directly and indirectly associative relations are used to derive candidate comparable entities, as described in the following:

(1). For an entity e_f (focal entity), $Q_{e_f} = \{q | e_f \in q.K, \forall q \in Q\}$ is retrieved from the universal set Q. Q_{e_f} represents the set of web search logs that contain e_f.
(2). Starting with the focal entity e_f, ICE seeks all search keywords in Q_{e_f} that are directly associative with e_f and builds the intermediate keyword set $M = \{k_m | (k_m, e_f) \in R^d\}$.
(3). Via the same approach as in steps (1) and (2), for each intermediate keyword k_m in M, ICE extracts the web search log set of k_m and collects all keywords that are directly associative with it. These keywords are indirectly associative with e_f. The extraction outcomes for all intermediate keywords are merged to form the candidate entity set $E^{cand} = \{e_1^{cand}, e_2^{cand}, \cdots, e_{ncand}^{cand}\}$.

With these three steps, a set of candidate entities can be effectively derived based on the indirectly associative relations that are identified for the focal keyword e_f. Many intermediate keywords may be generated in step (2). Some keywords may cooccur with e_f but do not have practical meanings, let alone represent an aspect that is related to the keyword. Thus, the pointwise mutual information (PMI) [1] is used to measure the relevance between the focal entity e_f and the keywords that are directly associative with it, and is defined as:

$$PMI(e_f, k_m) = \frac{n(e_f, k_m)}{n(e_f)n(k_m)}, \quad (1)$$

where $n(e_f)$, $n(k_m)$ and $n(e_f, k_m)$ represent the search volumes of queries in Q that contain e_f, k_m, and both, respectively. Here, for a keyword k:

$$n(k) = \sum_{q \in Q, k \in q.K} q.N. \quad (2)$$

Based on PMI values, the keywords that are directly associative with e_f are ranked as $Kr = \{k_{(1)}, k_{(2)}, \cdots, k_{(l)}\}$, where l denotes the total number of directly associative keywords. A common entity has a limited number of meaningful aspects [28]. Thus, the top ranked keywords are selected as intermediate keywords for the generation of candidate comparable entities.

Furthermore, in the candidate set, an entity may appear in various

[1] https://github.com/fxsjy/jieba
[2] http://www.nltk.org
[3] https://nlp.stanford.edu/software/lex-parser.html

forms. Thus, entity normalization is needed to merge different forms of entities into the standard form. Facilitated by anchor texts in Wikipedia, which have been widely used in entity disambiguation and normalization [12], various expression forms of an entity can be collected. In Wikipedia, each link is associated with an anchor text. For example, "Mercedes" and "Benz" are anchored to "Mercedes-Benz". In ICE, the anchored entity (e.g.,"Mercedes-Benz") is regarded as the standard expression form for the normalization of entities that target it (e.g.,"Mercedes" and "Benz"). The normalization results can be manually checked and adjusted to ensure that all entities are in the proper form.

3.3. Comparable entity targeting

In the first stage of ICE, some outliers may be generated as candidates. For instance, "factory" may be included in the candidate entity set through the path "BMW"→"car"→"factory". Hence, a semantic analysis procedure is introduced into the second stage of ICE to eliminate the outliers and target comparable entities by measuring their similarities with the focal entity. If an entity e_i^{cand} in the candidate set is similar to the focal entity e_f, it could be regarded as a qualified comparable entity. Otherwise, it could be eliminated as an outlier. As discussed in Section 3.1, keywords that are directly associative with an entity e could be regarded as related aspects of it. Therefore, the entity profile of an entity e that is based on web search logs can be defined as follows.

Definition 3 (Entity profile)

Given a web search log set Q within a time period and an entity e, the profile of e, which is denoted as Profile(e), is defined as:

$$Profile(e) = \left\{ (w,n) | \forall (w,e) \in R^d, n = \sum_{\forall q \in Q, w \in q.K} q.N \right\} \tag{3}$$

Here, (w,n) is a tuple that contains the keyword w, which is directly associative with e, and the sum of its joint search volumes with e in all queries of Q. For example, for entity "BMW" in Table 1, *Profile*(*BMW*) is {(X5, 27,322), (Business, 12,524), (motor, 6430), (SUV,740), (price,740), (car, 1330), (4S, 1414)}. Search volumes can be regarded as the weights of keywords. A larger volume n for a keyword w in *Profile*(e) means that more consumers issue queries with e and w, which reflects that more consumers regard w as a significant aspect of e.

In brief, entity profiles can be regarded as detailed descriptions of entities from the perspective of consumers who use search engines. They represent consumers' top-of-mind concepts and attitudes toward an entity. Hence, similarities between entity profiles can be treated as similarities between entities, and comparable entities can be targeted based on the analysis of entity profiles.

Furthermore, to correctly measure the semantic similarities of entity profiles, a keyword embedding model, namely, word2vec, is implemented in ICE. From massive amounts of unstructured texts, the word2vec model can learn vector representations of keywords in a high-dimensional vector space [10,19]. The cosine distances of the learned vectors can effectively capture the semantic similarities of the corresponding keywords [10]. Since word2vec is a deep-learning-based model, it requires a large corpus for training the model [25]. Corpora of high quality that provide comprehensive information on entities and their related keywords can be utilized to train the word2vec model. The widely used corpora include Wikipedia data, search engine results, news, patent data or their combinations [8,19,25], the usage of which depends on the application scenarios (e.g. English or Chinese) and the corpus availability. In our experiment, Chinese web search logs were utilized as the experiment data; thus, the Chinese Wikipedia and top-100 search results of each candidate entity returned by Baidu.com were used for training the word2vec model, which will be explained in more detail later in Section 5.1.

In the trained word2vec model, each keyword w in the entity profiles is embedded into a vector $\omega = (s_1, s_2, \cdots, s_d)$, where d is the vector dimension in the word2vec model. To generate entity profile representations, each entity profile is modeled as a multivariate Gaussian distribution, from which the distributed representations of keywords that are contained in the profile are generated [20]. Namely, the distributed representation of a specific keyword w in *Profile*(e) is an independent and identically distributed sample that is drawn from a multivariate Gaussian distribution of *Profile*(e):

$$\omega \tilde{} \mathcal{N}(\mu_{\mathbf{e}}, \boldsymbol{\Sigma}_{\mathbf{e}}), \tag{4}$$

where $\mathcal{N}(\mu_{\mathbf{e}}, \boldsymbol{\Sigma}_{\mathbf{e}})$ is the multivariate Gaussian distribution of *Profile*(e), $\mu_{\mathbf{e}} = (u_{e,\ 1}, \ldots, u_{e,\ d})$ is the mean vector of the distribution, and $\boldsymbol{\Sigma}_{\mathbf{e}} = \begin{bmatrix} \sigma_{1,1}^2 & \sigma_{1,2}^2 & \cdots & \sigma_{1,d}^2 \\ \sigma_{2,1}^2 & \sigma_{2,2}^2 & \cdots & \sigma_{2,d}^2 \\ \vdots & \cdots & \cdots & \vdots \\ \sigma_{d,1}^2 & \sigma_{d,2}^2 & \cdots & \sigma_{d,d}^2 \end{bmatrix}$ is the covariance matrix with $\sigma_{f,\ g}$ being the covariance between the f-th dimension and g-th dimension of the distribution.

According to the maximum likelihood estimates, $\mu_{\mathbf{e}}$ and $\Sigma_{\mathbf{e}}$ are estimated by the sample mean and the empirical covariance matrix respectively. Thus, the mean vector $\mu_{\mathbf{e}}$ could be formulated as:

$$\mu_{\mathbf{e}} = \frac{\sum_{(w,n) \in Profile(e)} \omega \times n}{\sum_{(w,n) \in Profile(e)} n}. \tag{5}$$

Similarly, the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{e}}$ could then be formulated as:

$$\boldsymbol{\Sigma}_{\mathbf{e}} = \frac{\sum_{(w,n) \in Profile(e)} (\omega - \mu_{\mathbf{e}})(\omega - \mu_{\mathbf{e}})^{\mathbf{T}} \times n}{\sum_{(w,n) \in Profile(e)} n}. \tag{6}$$

To measure the similarity of each pair of entities, we compare the Gaussian distributions of their entity profile representations. For a candidate entity e_i^{cand}, its similarity score with the focal entity e_f is set equal to the cosine distance of their mean vectors of the Gaussian representation. Given that the mean vectors of the Gaussian distributions of *Profile*(e_f) and *Profile*(e_i^{cand}) are $\mu_{ef} = (u_{ef,\ 1}, u_{ef,\ 2}, \ldots, u_{ef,\ d})$ and $\mu_{ei^{cand}} = (u_{ei^{cand},\ 1}, u_{ei^{cand},\ 2}, \ldots, u_{ei^{cand},\ d})$ respectively, the similarity score can be denoted as:

$$sim\left(e_i^{cand}\right) = \frac{\mu_{e_f} \cdot \mu_{e_i^{cand}}}{\left\|\mu_{e_f}\right\| \left\|\mu_{e_i^{cand}}\right\|} = \frac{\sum_{s=1}^{d} u_{e_f,s} \times u_{e_i^{cand},s}}{\sqrt{\sum_{s=1}^{d} u_{e_f,s}^2} \times \sqrt{\sum_{s=1}^{d} u_{e_i^{cand},s}^2}}, \tag{7}$$

where d is the dimension of entity profiles' Gaussian representations. Candidate entities with low values of $sim(e_i^{cand})$ are regarded as having few commonalities with e_f and are labeled as outliers. ICE only retains top-ranked entities to constitute the comparable entity set for e_f, which is denoted as $E^{comp} = \{e_1^{comp}, e_2^{comp}, \cdots, e_{ncomp}^{comp}\}$, where *ncomp* is the number of comparable entities that users would like to know. In the main body of the experiments of this work, the top-20 ranked entities were identified as the comparable entity set. More details about the selection of the comparable entity identification set size and its influence will be explained and discussed in Section 5.2.2.

4. The algorithm

In this section, the algorithmic details and time complexity of ICE are analyzed to show the main factors affecting its computation time.

Algorithm 1 provides the pseudo-code of ICE. In the first stage,

HashMap data structure is adopted to map the directly associative relation by traversing over the web search log sets Q once. Supposing there are totally NQ queries in the web search log set Q, the time complexity for generating candidate entities is $O(NQ)$ (lines 3–5), which means the time complexity of this step is linearly related with NQ. By using the HashMap, directly associative keywords can be quickly obtained with the complexity of $O(1)$ (line 6). Assuming that the focal entity has NA directly associative keywords in Q on average, the time complexity to calculate the PMI from HashMap is $O(NA)$ (lines 7–10). Supposing the top NB ranked keywords are selected as the intermediate keyword set, the candidate entity set is generated and normalized with the time complexity of $O(NB + ncand)$ (lines 11–14), where $ncand$ is the number of the candidate entities derived. Therefore, the time complexity of the first stage is $O(NQ + NA + NB + ncand + 1)$, which means the computation complexity of the first stage is at the level of linear sum of the cardinality of four sets (i.e., NQ, NA, NB, and $ncand$).

In the second stage, profiles of the focal entity and the candidate entities is built with the time complexity of $O(ncand + 1)$ (lines 15–17). Assuming the top-m search results of each candidate entity are collected as the training corpus, the time complexity of building the training corpus of word2vec is $O(m \times ncand + 1)$(lines 18–20). The process of training the word2vec model and generating Gaussian distributions of entity profiles has the time complexity of $O(NT \times W + NT \times log_2 V + ncand)$ [18,19] (lines 21–24), where NT denotes the dimensions of word vectors, W denotes the size of windows and V represents the total number of words in the training corpus. The processes of similarity calculation and outlier labeling are at the cost of $O(ncand)$ (lines 25–27).

Algorithm 1 Algorithm of ICE.

Input: Focal entity e_f, comparable entity number $ncomp$, Wikipedia data $Wiki$, web search log set Q within a time period;
Output: $E^{comp} = \{e_1^{comp}, e_2^{comp}, ..., e_{ncomp}^{comp}\}$;

1: **Initialization:**
$HMap = \varnothing$; $E^{cand} = \varnothing$; $Qcorp = \varnothing$;
2: $Preprocessing(Q)$
3: **for** $q \in Q$ **do**
4: $Update\_HMap(q, HMap)$
5: **end for**
6: $Keyword = Generate\_set(e_f, HMap)$
7: **for** $k \in Keyword$ **do**
8: $PMI(k) = Calculate\_PMI(e_f, k, HMap)$
9: **end for**
10: $M = Generate\_rank(Keyword)$
11: **for** $k_m \in M$ **do**
12: $E^{cand} = Update\_candidate(k_m, HMap)$
13: **end for**
14: $E^{cand} = Normalize(E^{cand})$
15: $Profile(e_f) = Generate\_profile(e_f, HMap)$
16: **for** $e_i^{cand} \in E^{cand}$ **do**
17: $Profile(e_i^{cand}) = Generate\_profile(e_i^{cand}, HMap)$
18: $Qcorp = Qcorp + search\_result(e_i^{cand})$
19: **end for**
20: $Qcorp = Qcorp + Wiki$
21: $WM = Word2vec(Qcorp)$
22: $\mathcal{N}(\mu_{\mathbf{e_f}}, \mathbf{\Sigma_{e_f}}) = Generate\_distribution(WM, Profile(e_f))$
23: **for** $e_i^{cand} \in E^{cand}$ **do**
24: $\mathcal{N}(\mu_{\mathbf{e_i^{cand}}}, \mathbf{\Sigma_{e_i^{cand}}}) = Generate\_distribution(WM, Profile(e_i^{cand}))$
25: $sim(e_i^{cand}) = Similarity(\mathcal{N}(\mu_{\mathbf{e_f}}, \mathbf{\Sigma_{e_f}}), \mathcal{N}(\mu_{\mathbf{e_i^{cand}}}, \mathbf{\Sigma_{e_i^{cand}}}))$
26: **end for**
27: $E^{comp} = Generate\_rank(E^{cand}, ncomp)$

Therefore, the total time complexity of ICE is at a level of $O(NQ + NA + NB + (m + 4) \times ncand + NT \times W + NT \times log_2 V + 3)$. When the training corpus size becomes huge, m, W, $ncand$, and $NT \ll V$. Thus, the size of the training corpus of word2vec model is the dominant factor that affects the efficiency of ICE when dealing with large-scale web search logs, which means that users can control the running time of ICE by adjusting the corpus size.

Two computational experiments were conducted to verify the time complexity of ICE. Firstly, the running time for ICE and baseline methods to identify comparable entities of 30 focal entities were tested. These baselines and focal entities are the same as those in the subsequent experiments, and will be elaborated in Section 5. Then, by randomly selecting 20%, 40%, 60%, 80%, and 100% of the original web search log data and Wikipedia data as the experiment dataset, the ICE method was executed respectively, and the completion time was recorded accordingly. All the computational experiments were conducted on web services with 16 Intel Xeon Platinum 8000 series processors and 30GB RAM. Results of the two experiments are presented in Table 2 and Fig. 1 respectively, where the reported running time is the total time required by each method to perform the same computing task of identifying comparable entities for all the 30 focal entities.

Table 2
Running time of different methods to identify comparable entities of 30 focal entities.

	ICE	CoCI	TFGM	Logistic	CMiner
Running time (hours)	9.3947	3.3984	3.3682	3.4248	3.3278

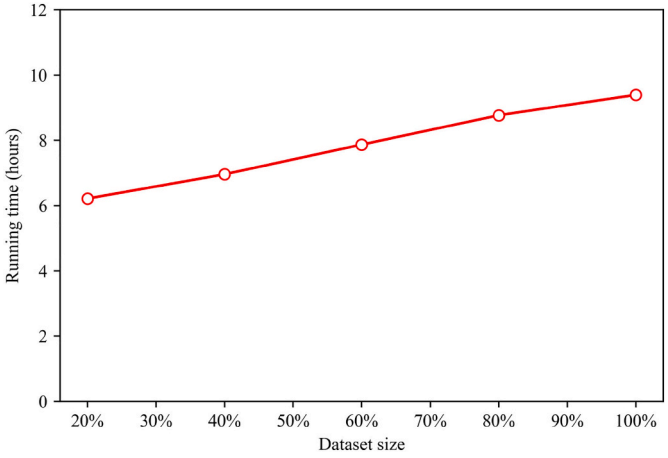

Fig. 1. Running time of ICE with datasets of different sizes.

As shown in Table 2, the computational efficiency of ICE is lower than that of CoCI, TFGM, Logistic and CMiner, since these baseline methods are either machine learning models (TFGM and Logistic) that are relatively easy to train or data mining models (CoCI and CMiner) that only need a few iterations to obtain the results. In contrast, ICE includes a deep-learning model that needs to update a large number of parameters during the model training. As expected, ICE took more time to identify comparable entities. It is worth noting that as an offline processing task, the running time of comparable entity identification is not sensitive in the practical application. Thus, the running time in Table 2 is acceptable when applying ICE to identifying more accurate and broader comparable entities in an offline environment.

Moreover, as illustrated in Fig. 1, the computing time of ICE increases with the size of the experiment dataset. As discussed above, the efficiency of ICE is dominated by the running time of the word2vec model. When the size of web search log data and the Wikipedia data grows, the size of the word2vec training corpus increases. Therefore, ICE's running time increases in an upward trend with the increase of the experiment data size.

5. Data experiments

This section aims to test whether ICE can identify accurate comparable entities and effectively cover the entities that users might compare. In addition, the effectiveness of ICE in comparable entity ranking is also validated.

Table 3
The list of focal entities.

Index	Focal Entity	Index	Focal Entity	Index	Focal Entity
1	Ausnutria	11	Galanz	21	Nikon
2	Baishaxi	12	GREE	22	PHILIPS
3	BMW	13	Hisense	23	Rejoice
4	Bosideng	14	Hodo	24	Saint Angelo
5	China Southern Airline	15	Home Inn	25	Sony
6	Dell	16	Johnson& Johnson	26	TISSOT
7	Dior	17	Luhua	27	Toshiba
8	Dove	18	M&G Chenguang	28	TTK Express
9	Forever bike	19	MG	29	Vinda
10	Fuanna	20	Nike	30	YONEX

5.1. Experimental setup

We conducted data experiments on web search logs that were collected from Baidu Tuiguang (http://e.baidu.com/), which provides a keyword tool for search engine advertising. Given a keyword, it supports the free download of the logs of web searches that contain the keyword issued by users of Baidu.com. The downloaded web search logs also include the search volume of each query within a time period and their bidding prices in search engine advertising. Such a tool can provide an effective mechanism for advertisers to obtain search logs that contain specified keywords and has been utilized for web search log analysis in previous studies [30].

As presented in Table 3, 30 focal entities were selected from the mainstream product/service category [31] of Taobao.com. From Baidu Tuiguang, web search logs for each focal entity and keywords that are directly and indirectly associative with it on January 5, 2018 were downloaded. On average, 18,411 web search logs were downloaded for each focal entity, and 1307 of them contained English words. Jieba, an effective Chinese segmentation tool [15], was employed to preprocess the web search logs into keywords. During the segmentation, word sequences were cut into several most suitable units according to the context. Proper nouns can be identified and segmented into one unit, even though they are composed of multiple words. Since entities are usually proper nouns, entities that are composed of more than one word are also segmented into one keyword instead of several separate keywords. In the segmentation results of web search logs, on average, 13 pieces out of 18,411 web search logs for a focal entity were not correctly segmented into one keyword. On top of that, Jieba can also preprocess English words by segmenting and tagging them into keywords. On this basis, function words (e.g., pronouns, conjunctions, and prepositions) were discarded, and only meaningful words were retained. In addition, the Chinese Wikipedia data were downloaded from an open-source website (https://dumps.wikimedia.org/zhwiki/latest/).

In the candidate entity generation stage, keywords that are directly associative with the focal entity were first mined. Existing UGC-based studies have found that an entity usually possesses 20 to 30 aspects on average [11]. Thus, the top-30 keywords that are most relevant to the focal entity in terms of PMI were utilized as the intermediate keyword set to further generate the candidate entity set. Then, the anchor texts in the Wikipedia data were used for the normalization of the candidate entities. To ensure accuracy, the normalization results were manually checked. It was found that some nicknames of entities cannot be normalized with Wikipedia and needed to be manually normalized. On average, 7 entities needed to be manually normalized for each focal entity and accounted for 0.65% of the candidate entities.

In the comparable entity targeting stage, the articles in Wikipedia were utilized to train the word2vec model. Since there may be a delay in the update of a few new entities in Wikipedia, to ensure that the trained word2vec model was domain-related to all the candidate entities, search engine results for each candidate entity were collected as a document and used for the word2vec training. In the main body of the experiments, the top-100 search results were included in the training corpus. The training corpus of the word2vec model was composed of 301,734 Wikipedia articles and 33,840 search result documents. Moreover, the effects of the search engine result size in the training corpus were tested, which will be presented in Section 5.2.2. Similar to the preprocessing of the web search logs, the texts in the training corpus were segmented into keywords with Jieba, and stopwords were removed, which resulted in a total of 1,019,360 unique keywords in the training corpus. Furthermore, to clarify the influence of keyword representation dimensions on ICE and choose the most suitable parameter, a pretest was conducted in advance. With keyword representation dimensions of 50, 75, 100, 125, 150, 175, and 200, different word2vec models were trained respectively, based on which comparable entities of each focal entity were identified. The differences between each pair of results generated with different parameters were measured by the Jaccard similarity coefficient. It was found that the Jaccard similarities between comparable entities identified with different dimensions were between 0.84 and 0.88, which indicated that ICE's comparable entity identification results remained stable among dimensions, and the change of keyword representation dimensions had little influence on the effectiveness of ICE. Thus, in the following experiments, the 100-dimensional keyword representations were used for comparison with other baseline methods. Through the word2vec model training, every keyword in the candidate entity profile was embedded into a 100-dimensional keyword vector. With the keyword vectors, mean vectors and covariance matrixes of entity profiles' multivariate Gaussian distributions were estimated, based on which the similarity scores between the candidate entities and the focal entity were calculated. In the main body of the experiments in this study, the top-20 ranked entities were identified as the comparable entity set, which was the same for every baseline method. In the following experiment, influences of the comparable entity set sizes were also tested, whose results will be discussed in Section 5.2.2.

5.2. Finding comparable entities

For comparable entity identification, four state-of-the-art methods were compared with ICE. In line with the common expression in design science research [4], all of these methods are called baselines in this study. The first baseline was a comparable entity identification method that was based on cooccurrence analysis [1,13], which we named CoCI. In light of the most recent studies in the field [13], three cooccurrence patterns (i.e., <$C/NN and $C/NN >, <$C/NN or $C/NN> and <$C/NN vs. $C/NN>) were utilized as initial pattern seeds to heuristically extract patterns and comparable entities. Each pattern is a word sequence, where $C represents comparable entities and /NN specifies that each entity must be a noun. Considering <$C/NN and $C/NN > as an example, this means that for a word sequence, if two nouns are connected by the word "and", these two nouns are identified as comparable entities. Meanwhile, three comparative relation mining methods were also selected as the baseline methods. In these methods, ICE's candidate entity sets identified by indirectly associative relations were utilized as their pre-given entity set. One method was adopted from the study of Yang et al. [33], which was a typical representative of comparative relation mining methods that are based on generative models. In this method, which is named TFGM, a topic model was trained on web search logs, and comparable entities were identified according to their similarities on topics. The third baseline method was a comparative relation mining method that is based on the classification model [17,21]. We mainly considered the method that was used in Pant and Sheng [21] and chose logistic regression as the classification model; we refer to this method Logistic. A keyword link network was constructed from web search logs, based on which in-link similarities, out-link similarities and text similarities were calculated as classification

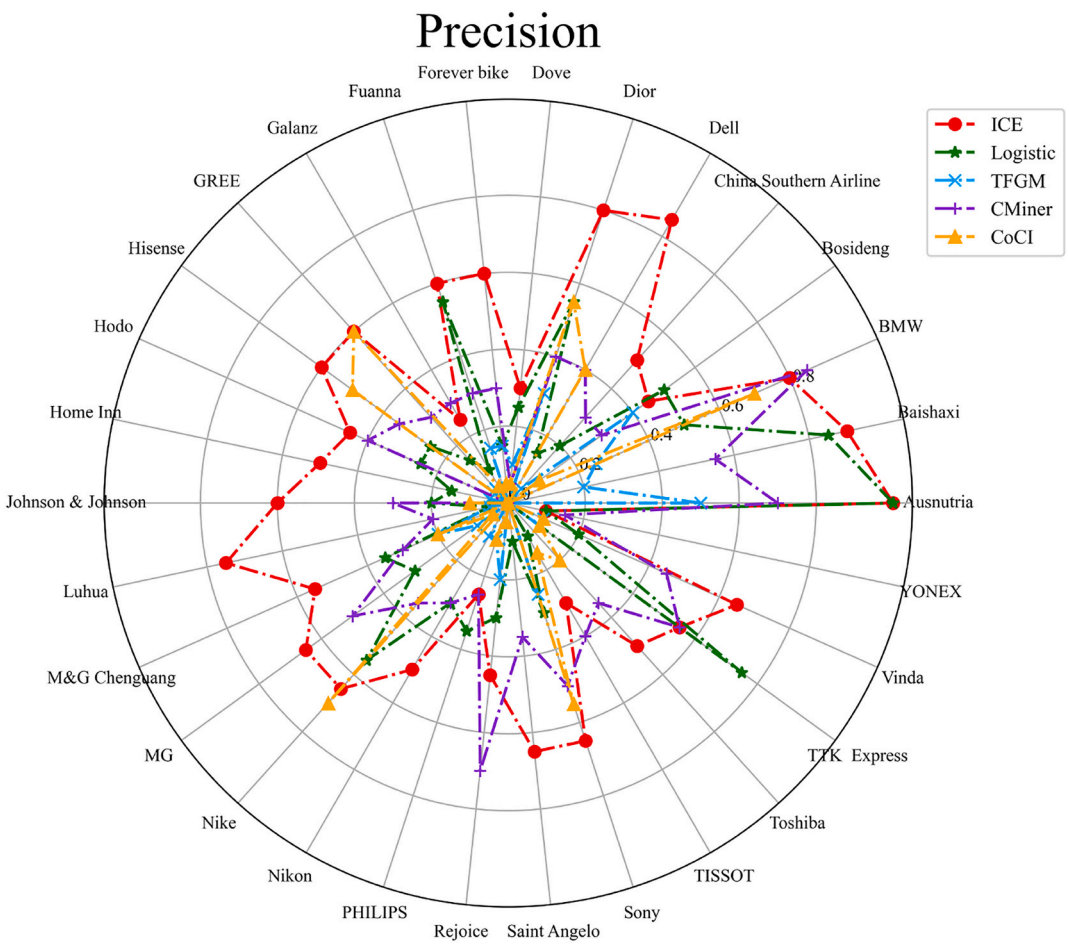

Fig. 2. Precision values of five comparable entity identification methods.

attributes. A total of 5000 entities were annotated manually as the training corpus for Logistic. Finally, a competitiveness analysis method that could also be used for comparable entity identification, namely, CMiner [29], was also included. In CMiner, the keywords that are directly associative with each entity were regarded as entity features (aspects), and the competitiveness between entities was calculated based on their overlap in entity features (aspects).

5.2.1. Evaluation data and metrics

For each comparative method, the top 20 identified comparable entities were selected as the result set of each focal entity. Fifteen human evaluators were employed to annotate comparable relationships between the identified comparable entities and the focal entities. To ensure the evaluation quality, all the employed evaluators were Ph.D. students from the department of management information systems and were familiar with the focal entities that were assigned to them according to their self-report. Each identified comparable entity was paired with its corresponding focal entity and was presented in random order to three evaluators. In total, the fifteen evaluators judged 7110 pairs of entities. For each entity pair (a focal entity and an identified comparable entity), the evaluators were asked to judge whether they were comparable in a practical business environment. Only the entities that were judged as comparable by at least two evaluators were regarded as having qualified comparable relationships.

Four metrics, namely, precision, recall, F_1-measure, and novelty, which are commonly used in the validation of comparable entity identification results [1,33], were considered in the evaluation. Precision is the fraction of qualified comparable entities in the result set, and could indicate the accuracy of a method in detecting qualified comparable entities. Given a focal entity, entities identified by method P_s are denoted as E_s, and TE_s is a subset of E_s containing all qualified comparable entities. The precision measure of the result generated by method P_s is formulated as:

$$Precision(P_s) = \frac{|TE_s|}{|E_s|}, \tag{8}$$

where $|TE_s|$ and $|E_s|$ represent cardinality of TE_s and E_s respectively.

Recall is the ratio of qualified comparable entities identified by a method over all qualified comparable entities in the market. For P_s, the recall measure is formulated as:

$$Recall(P_s) = \frac{|TE_s|}{\left|\cup_{j=1}^{B} TE_j\right|}, \tag{9}$$

where B is the total number of methods being compared.

F_1 metric, as the harmonic mean of Precision and Recall, is formulated as:

$$F_1(P_s) = \frac{2 \times Precision(P_s) \times Recall(P_s)}{Precision(P_s) + Recall(P_s)}. \tag{10}$$

Novelty, which evaluates whether a method could find novel qualified entities that other methods cannot detect, is formulated in the spirit of Lathia et al. [9]:

$$Novelty(P_s) = \frac{\left|TE_s \backslash \cup_{j=1, j\neq s}^{B} TE_j\right|}{|E_s|}, \tag{11}$$

where the numerator is the number of qualified comparable entities

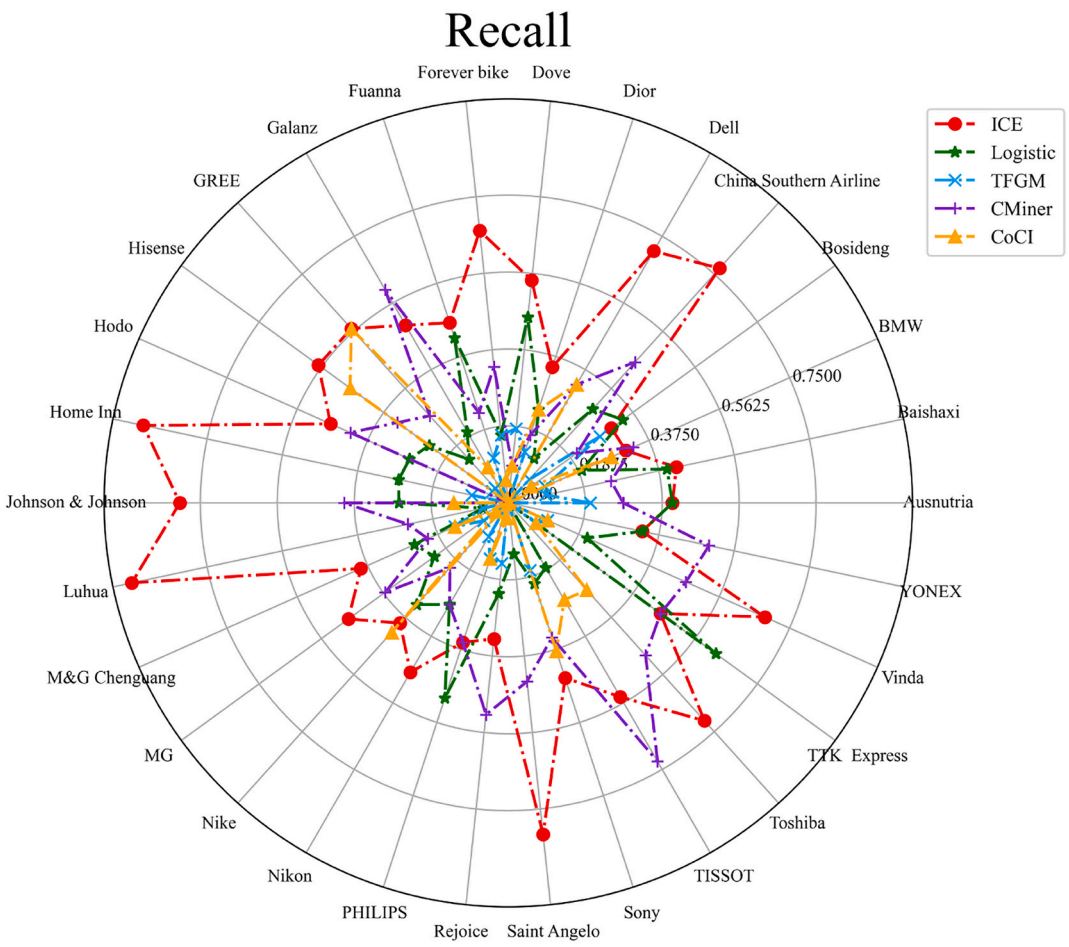

Fig. 3. Recall values of five comparable entity identification methods.

identified by P_s but not detected by other methods. Novelty can be regarded as the contribution of P_i in discovering novel and potential comparable entities lurked in the market for the focal entity.

5.2.2. Experimental results

Fig. 2 shows a radar chart of the precision values of the five methods, where the spokes represent the 30 focal entities that are presented in Table 3. For the 30 focal entities, ICE performed the best on 19 entities, and tied with baseline methods for the best performance on 2 entities. The larger precision values of ICE indicate that it could accurately identify more comparable entities than the baseline methods and could provide a more reliable comparable entity set.

Fig. 3 presents the radar chart of the recall values of the five methods on 30 focal entities, which shows that ICE outperformed the baseline methods on 19 of 30 focal entities, and performed as well as any other method on 2 entities. The larger recall values of ICE indicate that it could help detect a broader range of comparable entities than the other four methods, which is crucial for making decisions and monitoring potential threats in a changing market environment.

Fig. 4 presents the radar chart of the F_1-measure values of the five methods. Similar to precision and recall, ICE obtained better F_1-measure results on 19 entities and tied with other methods on 2 entities, thereby demonstrating that ICE could correctly and broadly identify comparable entities in the market and provide a more reliable comparable entity set than the baseline methods.

Table 4 presents the mean values of the precision, recall, and F_1-measure over 30 focal entities of ICE and the baseline methods. The paired t-test results demonstrate that at the confidence level of 99.9%, ICE significantly outperforms the four baseline methods in terms of precision, recall and F_1-measure. In terms of precision, ICE outperformed CoCI, TFGM, Logistic and CMiner by 39.0%, 45.2%, 24.8% and 19.5%, respectively. ICE also realized improved recall by 39.6%, 45.3%, 27.1% and 19.2% over CoCI, TFGM, Logistic and CMiner, respectively.

ICE occasionally did not perform as well as the CoCI method or the Logistic method. ICE's relatively dissatisfactory performances in terms of precision, recall, and F_1-measure occurred on the same focal entities. Therefore, we conducted an error analysis for these cases and found that the quality of the training data for the word2vec modeling was the major reason. Considering "Rejoice" as an example, when utilizing the word2vec model to learn the semantic representations of "Rejoice" and its candidate comparable entities, 3070 keywords in the entity profiles had the frequencies of less than 3 in the training corpus. These keywords included "anti-hair loss", which is one of the requirements for the shampoo; "ketoconazole", which is an ingredient that could prevent dandruff in the shampoo; and "itchy head", which is a symptom that consumers want to control by using the shampoo. Due to their low frequencies in the training corpus, the vector representations of these keywords could not be effectively generated in the word2vec model, thereby leading to errors in the semantic representation of the candidate entity profiles and the similarity score calculation. Similar cases were also identified in the focal entity "Bosideng", which is a Chinese clothing brand. For "Bosideng", the semantic representations of 2102 keywords in the entity profiles were missing, which included "girlfriends outfit", "capless" and "leather sleeve" and so on. On average, in cases in which ICE generated results with high precision and recall, the average number of keywords in the entity profiles for which semantic representations could not be found in the word2vec model was 1681, while in the few

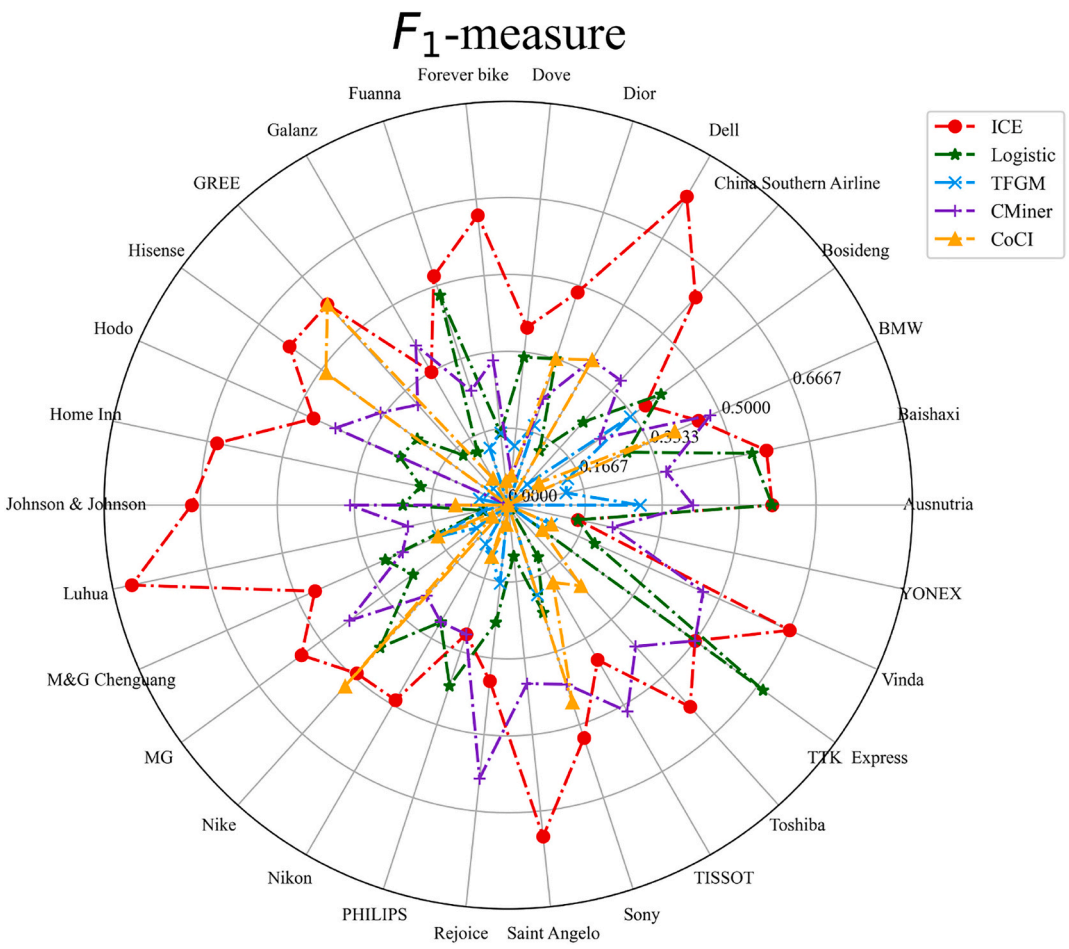

Fig. 4. F_1-measure values of five comparable entity identification methods.

Table 4
Mean values of Precision, Recall and F_1-measure.

	ICE	CoCI	TFGM	Logistic	CMiner
Precision	0.5667	0.1767***	0.1150***	0.3183***	0.3717***
Recall	0.5379	0.1418***	0.0850***	0.2665***	0.3463***
F_1-measure	0.5237	0.1504***	0.0931***	0.2726***	0.3378***

* Marks of "***" indicate ICE method's statistically significant improvement over other comparison methods according to the paired t-test at $p < 0.001$ level.

cases where ICE performed unsatisfactorily, the average number of missing word vectors was 1913, which was approximately 28% worse. The missing semantic representations of these important aspects or consumer requirements reduced the effectiveness of the entity representations and the similarity score measurement, thereby reduced the performance of ICE in comparable entity identification.

Table 5
Novelty values of ICE.

Focal Entity Index	Novelty	Focal Entity Index	Novelty	Focal Entity Index	Novelty
1	0.30	11	0.10	21	0.40
2	0.55	12	0.30	22	0.15
3	0.40	13	0.30	23	0.20
4	0.20	14	0.30	24	0.40
5	0.15	15	0.35	25	0.25
6	0.40	16	0.30	26	0.05
7	0.50	17	0.55	27	0.20
8	0.20	18	0.40	28	0.20
9	0.30	19	0.40	29	0.25
10	0.30	20	0.30	30	0.00
			Average: 0.29		

In addition, how the numbers of search engine results that are included in the word2vec training corpus affected ICE was tested. In this experiment, the corpus without any search engine results as well as the corpora including the top-25, top-50, top-75, top-100, top-125 and, top-150 search engine results of each entity were utilized for training the word2vec model respectively. From each trained word2vec model, vector representations of keywords were obtained. Then the multivariate Gaussian distributions of entity profiles were estimated. At last, the comparable entities of 30 focal entities were identified. Precisions of comparable entity identification results were measured. Results show that the precision trend of comparable entity identification results took a turn at 100, which means 100 was the most suitable number of search engine results in the training corpus of our experiment setting. From 25 to 100, with the increase of search results contained in the training corpus, the description of entities became more abundant and complete. However, as the quality and the relevance of low-ranked search results were getting worse, when the search engine result numbers exceeded 100, including more search results in the training corpus would introduce noise information and hence reduce the accuracy of vector representations.

The effect of the number of final identified comparable entities was also evaluated for ICE. With 1 to 30 comparable entities in the final identified set, ICE outperformed the baseline methods on each set size in terms of average precision, recall, and F_1-measure. As the size of the final set increased, the precision values of the identified comparable entities decreased since noise may be introduced into the ICE results. Moreover,

Table 6
nDCG values of three comparable entity ranking approaches.

Focal Entity Index	ICE	CMiner	BCQ	Focal Entity Index	ICE	CMiner	BCQ
1	**0.5855**	0.2923	0.3061	16	0.3249	**0.4257**	0.1383
2	0.7379	0.7993	**0.8328**	17	**0.6273**	0.3744	0.1374
3	0.4766	0.4378	**0.7057**	18	0.4579	**0.9095**	0.3803
4	0.6423	**0.9729**	0.5184	19	**0.8089**	0.5487	0.8020
5	**0.4515**	0.3836	0.2417	20	0.3974	**0.5916**	0.2418
6	0.4396	**0.8377**	0.2568	21	**0.3819**	0.2477	0.2202
7	**0.3983**	0.2620	0.2795	22	**0.5418**	0.4122	0.2293
8	**0.5120**	0.4246	0.4989	23	**0.6338**	0.6140	0.5262
9	**0.2984**	0.2212	0.2457	24	0.6310	**0.9982**	0.1656
10	0.6952	0.3405	**0.8490**	25	0.2238	**0.3471**	0.2057
11	**0.4366**	0.2218	0.1808	26	0.6154	**0.9486**	0.2714
12	0.5468	**0.7339**	0.2902	27	**0.3722**	0.2041	0.1821
13	0.3748	**0.4773**	0.3303	28	**0.6351**	0.6247	0.2798
14	**0.6557**	0.4332	0.4306	29	**0.7463**	0.4292	0.4769
15	**0.5407**	0.2191	0.2028	30	0.5220	**0.5388**	0.1498
Average	**ICE:0.5237**			CMiner: 0.5091		BCQ:0.3525	

the recall values increased with the size of the final set, namely, more comparable entities were included by ICE with an expanded final set. The F_1-measure of ICE increased with the size when the final set size was less than 20. After the size exceeded 20, the F_1-measure results remained relatively stable, which indicates that the comparable entity identification results reached a balance between accuracy and broadness after the size exceeded 20. Thus, in the experiments, the top-20 ranked entities of ICE and the baseline methods were identified as the comparable entity identification results to compare their performance.

Moreover, we evaluated ICE's performance in identifying novel comparable entities using the metric of novelty, as presented in Table 5. For most focal entities, the novelty values of ICE were larger than zero; hence, ICE could provide novel qualified comparable entities apart from the entities that were identified by other methods. The average novelty value was 29%; thus, 29% of the comparable entities that were identified by ICE could not be identified by other methods.

Considering all the above results, we conclude the following: ICE significantly outperformed the baseline methods in terms of precision, recall, F_1-measure and novelty, which is desirable for supporting high-quality decisions through the identification of comparable entities.

5.3. Comparable entity ranking

In this section, we further evaluate the effectiveness of ICE in comparable entity ranking. Two recent baseline methods were selected in the experiments. The first one (CMiner) was proposed by Valkanas et al. [29] that ranked comparable entities according to their competitiveness calculated on the coverage in feature space. The second method (BCQ) [30] measured the competitiveness on the strength of the bipartite graph that was developed from web search logs and ranked comparable entity accordingly.

5.3.1. Evaluation data and metrics

Bao et al. [1] found that the numbers of web pages that were returned in the search engines by using comparative queries were satisfactory indicators of competitiveness between entities. According to Li et al. [13],"vs" was a strong indicator for comparison intent in web search queries. Therefore, we used the number of web pages that were obtained by searching "entity1 vs entity2" in a search engine to externally validate the competitiveness of two entities. For each focal entity e_f and its comparable entity e_i^{comp}, we collected the number of web pages that were returned by Baidu.com using a phrasal query "e_f vs e_i^{comp}"". The numbers of web pages that were returned in response to individually searching "e_f" and "e_i^{comp}" were also collected to eliminate the popularity impact of each entity. To further eliminate the dimensional effect, the benchmark competitiveness between e_f and e_i^{comp} is formulated as a dimensionless measurement:

$$Comp(e_f e_i^{comp}) = \frac{S(e_f \text{ vs } e_i^{comp})}{S(e_f)} \times \frac{S(e_f \text{ vs } e_i^{comp})}{S(e_i^{comp})}, \tag{12}$$

where $S(x)$ represents the number of webpages returned by searching x.

Based on Eq. (12), we could rank all comparable entities from the most comparable to the least, the result of which was regarded as the benchmark ranking. The entities were also ranked by ICE, CMiner and BCQ based on their competitiveness results. A typical metric, namely, nDCG, which has been widely used in recommendation systems, information retrieval and competitiveness analysis [30], was adopted to compare the benchmark ranking and the ranking of each method in terms of consistency. The metric collates the provided ranking with the benchmark ranking by computing cumulative gains. The ranking with a larger nDCG value is more consistent with the benchmark ranking. Given a focal entity e_f, the set of identified comparable entities is $E^{comp} = \{e_1^{comp}, e_2^{comp}, \ldots, e_{20}^{comp}\}$, nDCG of method P_t could be formulated as:

$$nDCG_t = \frac{1}{Z}\left[\sum_{i=1}^{20} \frac{2^{comp(e_f, e_i^{comp})} - 1}{log_2(1 + Ind_t(e_i^{comp}))}\right], \tag{13}$$

where $comp(e_f, e_i^{comp})$ represents the benchmark competitivenesss between e_f and e_i^{comp} calculated based on Eq. (12), $Ind_t(e_i^{comp})$ is the index for e_i^{comp} in the ranking result of P_t, and Z is the maximum possible discounted cumulative gain if entities are properly ranked, and is used as the normalization factor to ensure the result is between 0 and 1. Z is formulated as:

$$Z = \sum_{i=1}^{20} \frac{2^{comp(e_f, e_i^{comp})} - 1}{log_2(1 + BInd(e_i^{comp}))}, \tag{14}$$

where $BInd(e_i^{comp})$ represents the index assigned for e_i^{comp} in the benchmark ranking.

Moreover, a user experiment was conducted to evaluate the performance of comparable entity ranking at a finer granularity. Based on the results of comparable entity annotation in Section 5.2, we invited another ten evaluators to rate the competitivenss of each pair of comparable entities, namely, a qualified comparable entity and its corresponding focal entity. For each pair of entities, the evaluators were asked to label the competitiveness type between the entities, where 1 represented strictly comparable, 2 represented moderately comparable, and 3 represented loosely comparable. Each entity pair was randomly assigned to three evaluators, and only the entities that were labeled to be strictly comparable by the majority were considered to be vital comparable entities, which implies that they had a strongly comparable relationship with the focal entity and should be chiefly addressed in the decision-making process.

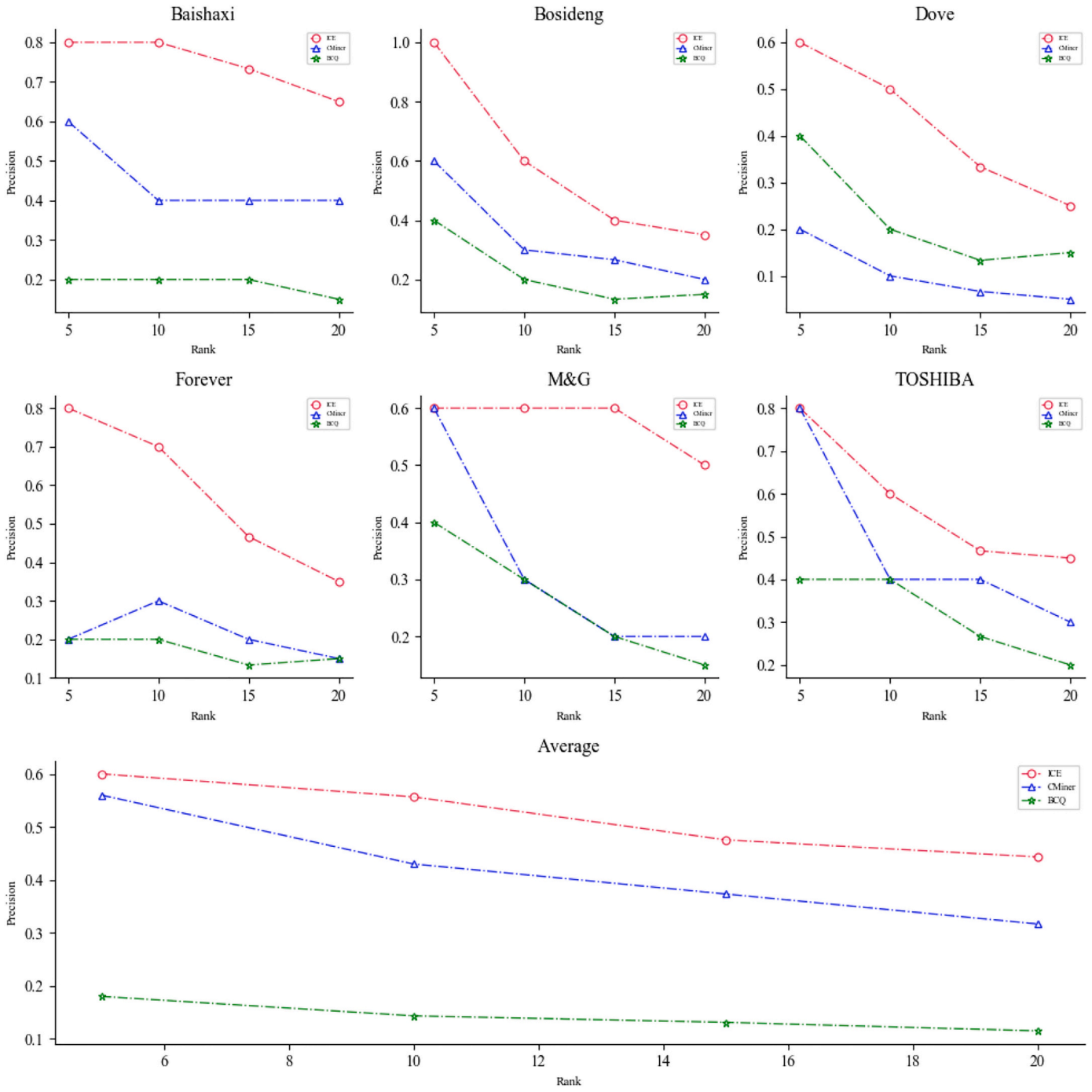

Fig. 5. Precision@k results of ICE, CMiner and BCQ.

To evaluate the ranking performance, we used the precision@k metric, which is often used to measure the quality of ranked lists [16]. A larger value of the precision@k metric corresponds to a higher performance of method P_t in ranking vital comparable entities at the top of the result list. Given a focal entity, its top-k ranked comparable entity results generated by method P_t is denoted by E_{tk}. KE_{tk} is a subset of E_{tk} and contains all vital comparable entities of the focal entity. Precision@k is defined as the fraction of vital comparable entities out of the top-k ranked entities, and is formulated as:

$$Precision@k(P_t) = \frac{|KE_{tk}|}{k}, \tag{15}$$

where $|KE_{tk}|$ is the cardinality of KE_{tk}. The variable of k was set as 5, 10, 15 and 20 in the experiments.

5.3.2. Experimental results

The nDCG values of the three methods across 30 focal entities are listed in Table 6, where the bold fonts indicate the best nDCG values. In terms of accuracy of comparable entity ranking, ICE outperformed the other two methods in most cases and had the best nDCG value for 16 focal entities. CMiner performed competitively with ICE and performed best for 11 focal entity entities. Among the three methods, ICE had the largest average nDCG values for all the 30 focal entities; hence, its ranking results were more consistent with the benchmark ranking and could provide a reliable comparable entity ranking. If the definition of Eq. (12) is modified to have only one term in the numerator, namely, if

only one $S(e_f vse_i^{comp})$ is used for the competitiveness calculation, the above findings still hold. In that case, ICE also obtained more accurate comparable entity ranking results than the other two baseline methods and had the best nDCG values for 18 focal entities. The average nDCG values of ICE, CMiner and BCQ were 0.5038, 0.4363 and 0.2859, respectively.

Fig. 5.3.2 shows the precision@k results of the three methods, where the horizontal coordinates represent the k values and the vertical coordinates represent the precision@k values, which ranges from 0 to 1. Limited by the space, the detailed results of 6 randomly selected focal entity entities are presented here, and the plot at the bottom illustrates the averaged precision@k across all the 30 focal entities. The results in Fig. 5.3.2 demonstrate that as k increased, the precision@k values of the three methods declined overall; thus, all these methods were relatively effective and could rank more vital comparable entities ahead. ICE realized the best precision@k performances with various values of k. The larger precision@k values of ICE show that it could rank more vital comparable entities ahead than the other two ranking methods; thus, it could provide a more accurate comparable entity ranking.

Overall, in terms of nDCG, ICE performed the best in most cases and had the largest average nDCG value; hence, the ranking results of ICE were more consistent with the benchmark ranking. In terms of precision@k, with the increase of k, all the three methods exhibited similar trends; thus, they were all effective in ranking vital comparable entities ahead. Moreover, ICE had the best precision@k values; therefore, compared with the other two ranking methods, ICE could rank more vital comparable entities ahead.

6. Conclusions

Comparable entity identification is desirable for both consumers and managers in their decision-making processes. In this paper, a novel two-stage ICE method for comparable entity identification that is based on web search logs has been proposed from the perspective of consumers. In the first stage, a candidate entity generation process has been designed based on the indirectly associative relation of comparable entities that are linked by their shared related aspect information. Furthermore, using a word2vec model that was trained on Wikipedia data and web pages that were returned by search engines, a comparable entity targeting process in light of the semantic analysis of entity profiles has been developed for detecting qualified comparable entities from the candidate set. Moreover, data experiments have shown that the ICE method has performance advantages over the baseline methods for both comparable entity identification and ranking.

From a practical perspective, the method that is proposed in this study can support both consumers and managers in accurately and comprehensively understanding of comparable entities, which can further help them make well-informed and rational decisions. Consumers can choose the entity in which they are initially interested as the input to ICE and obtain a broad set of comparable entities from ICE. The result set, which effectively expands consumers' awareness of comparable entities in their purchase processes, can help them select the item that best suits their needs. Moreover, managers can select their products as the focal entities of ICE, which can identify the corresponding comparable entities. With these entities, a series of analyses could be conducted to support the managers in various ways, e.g., regarding market competition, product design, and pricing strategy. To realize these applications, companies with technical strength could choose to implement the proposed framework of comparable entity identification by themselves since all the data that are used in ICE are openly available to the public. Search engine companies can also implement the proposed ICE method and provide it as a free/paid online plug-in or software to managers and consumers.

Future studies can be focused in four directions. One is the further extension of the proposed comparable entity identification method to market structure analysis since many users query and obtain relevant information about products using search engines before making purchases. In addition, in the entity normalization of ICE, entities that are not present in the anchor text of Wikipedia, such as nicknames, cannot be automatically normalized and must be manually added to the normalization list. Future studies can incorporate a more effective entity normalization method for these entities to improve the comprehensiveness of the entity normalization result. Moreover, the inability to process entities that are composed of multiple words when they are incorrectly segmented by the text segmentation tools is another limitation in web search log segmentation of ICE. How to overcome this limitation is also a valuable future research direction. Another future direction is the extension of the proposed method to other related research subjects in web searches, such as query suggestions and query intent analysis, in which entity comparison is necessary.

Acknowledgements

The work was partly supported by the National Natural Science Foundation of China (71772177, 72072177), the MOE Project of Key Research Institute of Humanities and Social Sciences at Universities (17JJD630006), and the joint PhD scholarship of Renmin Business School.

References

[1] S. Bao, R. Li, Y. Yu, Y. Cao, Competitor mining with the web, IEEE Trans. Knowl. Data Eng. 20 (2008) 1297–1310.
[2] J. Choi, A. Tosyali, B. Kim, H.s. Lee, M.K. Jeong, A novel method for identifying competitors using a financial transaction network, IEEE Trans. Eng. Manag. (2019) 1–16, https://doi.org/10.1109/TEM.2019.2931660.
[3] B.H. Clark, D.B. Montgomery, Managerial identification of competitors, J. Mark. 63 (1999) 67–83.
[4] S. Gregor, A.R. Hevner, Positioning and presenting design science research for maximum impact, MIS Q. 37 (2013) 337–356.
[5] A. Jain, P. Pantel, How do they compare? Automatic identification of comparable entities on the web, in: 2011 IEEE International Conference on Information Reuse & Integration 228–233, IEEE, 2011.
[6] Z. Jiang, L. Ji, J. Zhang, J. Yan, P. Guo, N. Liu, Learning open-domain comparable entity graphs from user search queries, in: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, ACM, 2013, pp. 2339–2344.
[7] N. Jindal, B. Liu, Mining comparative sentences and relations, in: AAAI, 2006, p. 9.
[8] M. Kraus, S. Feuerriegel, Decision support from financial disclosures with deep neural networks and transfer learning, Decis. Support. Syst. 104 (2017) 38–48.
[9] N. Lathia, S. Hailes, L. Capra, X. Amatriain, Temporal diversity in recommender systems, in: Proceedings of the 33rd International Conference on Research and Development in Information Retrieval, ACM, 2010, pp. 210–217.
[10] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: International Conference on Machine Learning, 2014, pp. 1188–1196.
[11] T.Y. Lee, E.T. Bradlow, Automated marketing research using online customer reviews, J. Mark. Res. 48 (2011) 881–894.
[12] C. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, B.S. Lee, Twiner: Named entity recognition in targeted twitter stream, in: Proceedings of the 35th ACM SIGIR International Conference on Research and Development in Information Retrieval, ACM, 2012, pp. 721–730.
[13] S. Li, C.Y. Lin, Y.I. Song, Z. Li, Comparable entity mining from comparative questions, IEEE Trans. Knowl. Data Eng. 25 (2013) 1498–1509.
[14] X. Li, S. Zhang, B. Wang, Z. Gao, L. Fang, H. Xu, A hybrid framework for problem solving of comparative questions, IEEE Access 7 (2019) 185961–185976.
[15] Y. Liu, C. Jiang, H. Zhao, Assessing product competitive advantages from the perspective of customers by mining user-generated content on social media, Decis. Support. Syst. 123 (2019) 113079.
[16] S.L. Lo, R. Chiong, D. Cornforth, Ranking of high-value social audiences on twitter, Decis. Support. Syst. 85 (2016) 34–48.
[17] Z. Ma, G. Pant, O.R. Sheng, Mining competitor relationships from online news: a network-based approach, Electron. Commer. Res. Appl. 10 (2011) 418–427.
[18] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, 2013 arXiv preprint arXiv:1301.3781.
[19] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.
[20] G. Nikolentzos, P. Meladianos, F. Rousseau, Y. Stavrakas, M. Vazirgiannis, Multivariate gaussian document representation from word embeddings for text categorization, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, 2017, pp. 450–455.
[21] G. Pant, O.R. Sheng, Web footprints of firms: using online isomorphism for competitor identification, Inf. Syst. Res. 26 (2015) 188–209.

[22] M.A. Peteraf, M.E. Bergen, Scanning dynamic competitive landscapes: a market-based and resource-based framework, Strateg. Manag. J. 24 (2003) 1027–1041.
[23] D. Qiao, J. Zhang, Q. Wei, G. Chen, Finding competitive keywords from query logs to enhance search engine advertising, Inf. Manag. 54 (2017) 531–543.
[24] T. Ruan, Y. Lin, H. Wang, J.Z. Pan, A multi-strategy learning approach to competitor identification, in: Joint International Semantic Technology Conference, Springer, 2014, pp. 197–212.
[25] D. Shin, S. He, G.M. Lee, A.B. Whinston, S. Cetintas, K.C. Lee, Enhancing social media analysis with visual data analytics: a deep learning approach, Forthcoming at MIS Quarterly (2019) 1459–1492.
[26] J. Tang, B. Wang, Y. Yang, P. Hu, Y. Zhao, X. Yan, B. Gao, M. Huang, P. Xu, W. Li, et al., Patentminer: topic-driven patent analysis and mining, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2012, pp. 1366–1374.
[27] M. Tkachenko, H.W. Lauw, Comparative relation generative model, IEEE Trans. Knowl. Data Eng. 29 (2016) 771–783.
[28] P. Tsaparas, A. Ntoulas, E. Terzi, Selecting a comprehensive set of reviews, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2011, pp. 168–176.
[29] G. Valkanas, T. Lappas, D. Gunopulos, Mining competitors from large unstructured datasets, IEEE Trans. Knowl. Data Eng. 29 (2017) 1971–1984.
[30] Q. Wei, D. Qiao, J. Zhang, G. Chen, X. Guo, A novel bipartite graph based competitiveness degree analysis from query logs, ACM Trans. Knowl. Discov. Data 11 (2016) 21.
[31] H. Wu, G. Qiu, X. He, Y. Shi, M. Qu, J. Shen, J. Bu, C. Chen, Advertising keyword generation using active learning, in: Proceedings of the 18th International Conference on World wide web, ACM, 2009, pp. 1095–1096.
[32] K. Xu, S.S. Liao, J. Li, Y. Song, Mining comparative opinions from customer reviews for competitive intelligence, Decis. Support. Syst. 50 (2011) 743–754.
[33] Y. Yang, J. Tang, J. Keomany, Y. Zhao, J. Li, Y. Ding, T. Li, L. Wang, Mining competitive relationships by learning across heterogeneous networks, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, ACM, 2012, pp. 1432–1441.

Liye Wang is currently pursuing her PhD degree in the Department of Management Science and Engineering, School of Business, Renmin University of China. Her research interests include competitive intelligence, e-commerce and text mining. Her work has been published in the journal of *Frontiers of Business Research in China*.

Jin Zhang is an associate professor in the Department of Management Science and Engineering, School of Business, Renmin University of China. He received his PhD degree in the Department of Management Science and Engineering from the School of Economics and Management at Tsinghua University. His current research interests include data mining, business intelligence, and web search. His work has been published in journals such as *MIS Quarterly*, *INFORMS Journal on Computing*, *Decision Support Systems*, *Information & Management*, and *IEEE Transactions on Neural Network and Learning Systems*, etc.

Guoqing Chen received his PhD from the Catholic University of Leuven (K.U. Leuven, Belgium) and now is Professor of Information Systems at the School of Economics and Management, Tsinghua University, Beijing, China. His research interests include information systems management, business analytics and decision support systems. His work has been published in journals such as *MIS Quarterly*, *Journal of Management Information Systems*, *Journal of Association for Information Systems*, *Decision Sciences*, *INFORMS Journal on Computing*, *Decision Support Systems*, *ACM Transactions on Knowledge Discovery from Data*, etc.

Dandan Qiao is an assistant professor in the Department of Information Systems and Analytics at the National University of Singapore (NUS). Prior to joining NUS, She earned her Ph.D in the Department of Information Systems from Tsinghua University. Her current research interests lie in the intersection of information systems, behavioural science, and data mining. Her work has been published in journals such as *MIS Quarterly*, *Information Systems Research*, *Information & Management*, *ACM Transactions on Knowledge Discovery from Data*, etc.