Computers in Industry 78 (2016) 29–42

Contents lists available at ScienceDirect

Computers in Industry

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

Integrating a semantic-based retrieval agent into case-based reasoning systems: A case study of an online bookstore

Jia Wei Chang [a], Ming Che Lee [b,*], Tzone I Wang [a]

[a] Department of Engineering Science, National Cheng Kung University, Tainan, Taiwan
[b] Department of Computer and Communication Engineering, Ming Chuan University, Taoyuan, Taiwan

ARTICLE INFO

Article history:
Received 27 November 2014
Received in revised form 4 October 2015
Accepted 13 October 2015
Available online 26 October 2015

Keywords:
Short-text semantic similarity
Case-based reasoning
Case-based machine learning

ABSTRACT

Natural language search engines should be developed to provide a friendly environment for business-to-consumer e-commerce that reduce the fatigue customers experience and help them decide what to buy. To support product information retrieval and reuse, this paper presents a novel framework for a case-based reasoning system that includes a collaborative filtering mechanism and a semantic-based case retrieval agent. Furthermore, the case retrieval agent integrates short-text semantic similarity (STSS) and recognizing textual entailment (RTE). The proposed approach was evaluated using competitive methods in the performance of STSS and RTE, and according to the results, the proposed approach outperforms most previously described approaches. Finally, the effectiveness of the proposed approach was investigated using a case study of an online bookstore, and according to the results of case study, the proposed approach outperforms a compared system using string similarity and an existing e-commerce system, Amazon.

1. Introduction

Business-to-consumer (B2C) refers to a business model where companies and consumers trade digitally, which also describes a company that provides goods and services for consumers on the internet, one of the best examples of B2C e-procurement is the online bookstore of Amazon.com [1]. The focus of B2C e-procurement is to suggest enticing prospects to customers and to retain them and the share values they created [2], the final goal of which is to convert shoppers into buyers actively and constantly. Related studies have demonstrated that the recommendation quality of a commerce system directly affects customer satisfaction of a website [3–5]. However, most existing commerce systems use keyword searches, which have performance limitations because it is not possible for users to ask their question by using natural language and to acquire answers instantly by logically matching potential words that might be related. With the development of natural language processing (NLP), natural language search is a possible solution to the problems, thus allowing users to express what they want in their own words. In addition, natural language search can "read the minds" of users, enabling them to use internet technology more easily, thus improving the recommendation quality, reducing the fatigue associated with search engine use, and transforming the search experience into an effective, positive, and more human experience.

Existing recommender systems are mainly classified into collaborative filtering (CF) techniques and content-based (CB) methodologies. CF techniques recommended items by the known preferences of users. However, new items are not included because they have to be rated by many users before they can be recommended. This is called the cold-start problem, which limits their performance [6]. By contrast, CB methods recommend new items because recommendations are based on the descriptive characteristics of items, which rely on more specific information about items. However, they must overcome the problems of limited diversity and possible overspecialization. Recent studies have demonstrated that a hybrid approach can combine the advantages of both techniques, overcome the limitations of CF and CB, and improve the accuracy of recommendation [7,8].

Case-based reasoning (CBR) is a popular model that develops commerce recommendation systems [9], and is a framework with a high compatibility of combining CF [6,10,11] as well as CB [9,12,13]. In this technique, new problems are solved by utilizing or modifying the solutions of similar existing problems, the core of which is using similarity measure to quantify the differences that exist between objects [14] because CBR uses similarity measures to

* Corresponding author.
E-mail address: leemc@mail.mcu.edu.tw (M.C. Lee).

http://dx.doi.org/10.1016/j.compind.2015.10.007
0166-3615/

identify cases that are similar to the problem at hand, and most of measures that evaluate similarities of non-numeric properties are syntactic methods (such as brute force, longest common subsequence, and Levenshtein distance) and calculate the string similarity between two words [15–17] but often fail to match exactly when confronted with the words that have associated meanings. To overcome the limitations of syntactic approaches, semantic similarity measure can be used that is more suitable for describing non-numeric properties than syntactic similarities, which also can help system to communicate directly with users through natural languages and to eliminate the burden of query formulation and the enormous document load needed to find the answers. The NLP-based technologies facilitate the CBR model that can streamline the integration and creation of knowledge-based systems [18]. Recent business systems have evolved out of NLP techniques that demonstrated the potential in business workflow [19–23].

This study proposes a hybrid CBR model that combines the CB and CF mechanisms to accept natural language queries and increase the recommendation accuracy. Mainly, we designed a semantic similarity algorithm to analyze the content of user queries and goods, which plays the role of a case retrieval agent in the proposed system. The algorithm can more accurately understand users' intention to retrieve the most appropriate cases for reuse, which facilitates our CBR system in that it does not require a high-quality case base in the beginning, thus solving the cold-start problem of the CF technique. Furthermore, the proposed system includes the CF mechanism for recommendation ranking. This not only avoids the problems of CB, such as limited diversity and overspecialization, but also makes our system stronger. Finally, the effectiveness of the proposed system is illustrated using a case study of an online bookstore service, such as Amazon, but it can be developed for any domain. This study aimed to (1) use semantic similarity measurements instead of string similarity measurements to retrieve cases that can facilitate our system in having a more satisfactory recommendation at the beginning; (2) integrate CB and CF mechanisms into a hybrid CBR system to constantly improve the recommendation quality; and (3) propose a NLP-based CBR approach that can accept natural language queries to achieve user friendliness.

The paper is organized as follows. Section 2 describes the theoretical framework and related works of the CBR system, word-sense disambiguation, and short-text semantic similarity. Section 3 provides the details of the development of the proposed system and retrieval method. Section 4 describes the performance test of the proposed STSS algorithm by using established benchmarks. Section 5 describes the evaluation of the semantic measures proposed in this paper and the effectiveness of the proposed approach is illustrated using a case study on an online bookstore. Finally, Section 6 presents the conclusions.

2. Backgrounds

This section provides a focused introduction to the relevant foundations of CBR systems and word-sense disambiguation, as well as a review of short-text semantic similarity.

2.1. Case-based reasoning systems

For application-oriented projects, most systems are based on a CBR architecture [24–28], which indicates that CBR is a well-established model that facilitates methodology design in industry engineering and B2C commerce. CBR is a branch of artificial intelligence (AI), which is a method based on using past experience for problem solving and decision making, searching for the solutions to novel problems using previously solved problems, and reusing the existing solutions in new situations [29,30]. The outstanding characteristic of CBR is that it does not need to match the user's query exactly, such as in searching problems. These cases are usually similar to some extent. Because the basic assumption of CBR is that similar problems have similar solutions, even if the repository does not contain a solution that immediately addresses a user's problem, a similar answer can be available for use as a starting point. Similar solutions can then be adapted and at least provide some inspiration and guidance for the user. A CBR system usually consists of domain/expert knowledge, a case-base of past experiences, and a similarity measure for searching related cases [31]. Domain knowledge refers to knowledge about the features of the different entities and what is a "case". A case-base contains a set of cases, each of which describes a problem, a solution to the problem, and annotations about how the solution was derived. Similarity measures are developed according to the features of the case because the problem is typically defined in terms of specific features of objects, and those features can be numeric or non-numeric properties; the similarity measure is used to identify the cases that are most relevant to the problem. The CBR system responds to the query using a given algorithm and a similarity measure, which consists of translating and matching a query against a set of information objects. Finally, a similarity measure calculates the similarities that exist between objects. To realize automated reasoning, CBR systems basically have been formalized as a four-step process (Fig. 1) [32]:

Retrieve: Given a specific problem, retrieve similar cases from the case-base/repository to solve the problem.
Reuse: Choose the possible solutions from the retrieved cases. If the solutions are not able to be used directly, they need to be adapted for the new situation.
Revise: Modify the existing solution to the target problem, test the new solution for the problem, and if necessary, continue to revise.
Retain: Store the resulting new cases in the repository if the solution has been successfully applied to the target problem.

However, ambiguity often occurs in query design when users have no idea about the exact expressions and the related concepts they want to know, but they may have some contextual clues, such as the function of the objective. Therefore, traditional CBR uses syntactic similarity measurements, which are not intelligent enough for meaning-related searches. In contrast, because engineering designers usually need to construct a case-base for the CBR system and the task often needs many manual works and expert experiences, automatic knowledge acquisition has become an emerging issue for the development of CBR systems. Moreover, solutions from past cases may not directly be reusable; in these situations, they should be adapted to better fit the new problem.

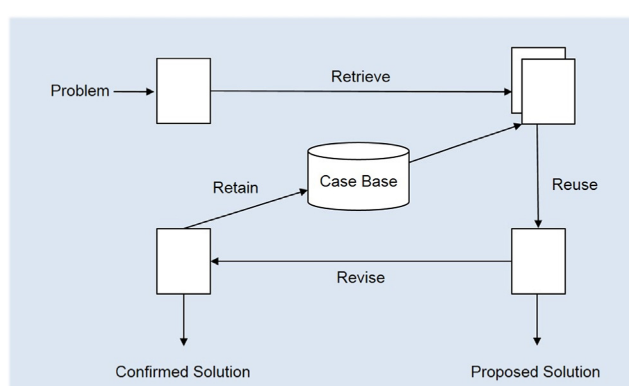


Fig. 1. The traditional CBR mechanism.

Finally, the revised case is retained to provide sustained learning, but this step usually needs the judgment of an expert. Collaborative filtering can be used to replace manual judgment, which is an automatic judgment method that predicts the decisions of a user by collecting the preferences from many users [33,34]; Facebook, YouTube, and Amazon are typical examples of collaborative filtering services [35]. A hybrid CBR approach, automatic knowledge acquisition, and collaborative filtering become interesting issues in the development of an intelligent commerce model.

2.2. Word-sense disambiguation

Ambiguity (also called word polysemy) is a crucial problem of NLP, which refers to a word or phrase with multiple meanings, and thus the word or phrase is difficult to parse. For example, the verb "to get" can mean "procure" (I will get the drinks), "become" (she got scared), "have" (I have got three dollars), "understand" (I get it), etc. Therefore, researchers began to study word-sense disambiguation (WSD) to solve the ambiguity problem. WSD is to identify which sense of a word (i.e. meaning) is used in a sentence when the word has multiple meanings. The WordNet-based method is a famous WSD that uses knowledge resources to infer the senses of words in context. The knowledge-based methods usually have the advantage of a broader coverage, thanks to the use of large-scale knowledge resources [36]. Since the introduction of computational lexicons, such as WordNet, a number of structural approaches have been developed to analyze and exploit the structure of the concept network made available in such lexicons.

The latest version of WordNet is 3.0, which contains more than 155,000 words, 117,000 synsets and in particular of the semantic relationships among words and concepts. In WordNet, nouns, verbs, adjectives and adverbs are grouped into cognitive synonyms called "synsets", and each synonym expresses a distinct concept and has a descriptive gloss, and the lexicalized synsets of nouns and verbs are organized hierarchically via hypernymy/hypernymy and hyponym/hyponymy. These characteristics are able to reduce the ambiguity of words. Thus, Navigli constructed a general WSD framework based on the WordNet-based measures [36], which can disambiguate the word w_i in a text $T = (w_1, \ldots, w_n)$ by choosing the sense $\hat{S}$ of w_i which maximizes the sum (formula (1)). Given a sense S of the word w_i and if w_i has multiple meanings, the formula sums the contribution of the most appropriate sense of each word of T but $w_j \neq w_i$. The sense with the highest sum is chosen after all senses of w_i are computed.

$$\hat{S} = \underset{S \in Sense_D(w_i)}{\operatorname{argmax}} \sum_{w_j \in T : w_j \neq w_i} \max_{S' \in Sense_D(w_j)} score(S, S'). \tag{1}$$

In general, three types of WordNet-based measures were presented (detailed explanations are presented in [36,37]): (1) distance-based measures, such as PATH [38], WUP [39] and LCH [40], which are based on the distance between two senses and focused on hypernymy links and scaled the path length by the overall depth of the taxonomy. Distance-based measures use the hierarchical structure of WordNet (or any other taxonomy with a similar structure), and the path length between concepts can be used to measure the similarity between concepts. (2) Information content (IC)-based measures use a notion of information content shared by words in context; these measurements determine the specificity of the concept that subsumed the words in the taxonomy and are based on the more specific concept that subsume more words. IC-based measures, such as RES [41], JCN [42] and LIN [43], also use the hierarchical structures and hypernymy links, and if two concepts are more similar, they share more information content. (3) Gloss-based measures, an intuitional knowledge-based approach, such as LESK [44] and VECTOR [45], rely on the calculation of the gloss overlap between the meanings of two words; the two targets whose definitions have the highest overlap are assumed to be the correct ones.

However, the distance-based and IC-based measures very depending on the hierarchical structure, which is only available for nouns and verbs and completely deficient for adjectives and adverbs [46]; this weakness prevents these two types of measures from solving the polysemy problem for adjectives and adverbs, but gloss-based measures have potential to address word pairs of adjectives and adverbs [37]. The superiority of WordNet-based measures is the stable repeatability of performance that can be ensured because they are based on a strict ontology. Thus, we expect the WordNet-based word similarity to play an important role in the short-text semantic similarity measures.

2.3. Short-text/sentence semantic similarity measures

Short-text semantic similarity (STSS) is the key technique of natural language search and is widely used in social network analysis [47], Opinion mining to look for undiscovered knowledge [48], and personal assistants, such as Apple Siri, Google Now, Samsung S Voice, and Microsoft Cortana. The definition of STSS task assumes bidirectional graded similarity equivalence between pairs of short-texts (e.g., a vehicle and a car are more similar than a wave and a car), and which usually measures text similarity use lengths of 10–20 words and even incomplete grammar. Similar to spoken utterances, short-texts/sentences do not necessarily follow formal grammatical rule that is most difficult to master because of lack of information and its syntactic and semantic flexibility. In addition, STSS is more directly applicable to many NLP tasks, for example, some STSS approaches play the role of recognizing textual entailment (RTE) [37,49,50]. However, RTE is the task of determining whether the meaning of the text can be inferred from another text, which is a directional equivalence and binary decision [51]. For example, "A car is a vehicle, but a vehicle is not necessarily a car.", this means that RTE is an asymmetric task. However, STSS is a symmetric task given the semantic similarity between two natural language entities.

Mainly, the methodologies of STSS can be separated into three types: corpus-based, ontology-based and hybrid approaches. A typical corpus-based method, such as STS [49], removes stop words and builds an m by n similarity matrix of the meaningful words in two short-texts. STS obtains corpus-based word similarity and string similarity of the word pairs from the $m \times n$ matrix, sums up the maximum-valued matrix-elements and multiplies the sum by the reciprocal harmonic mean of m and n to obtain a balanced similarity score between 0 and 1, inclusive. The greatest difference in STS is the use of the longest common subsequence (LCS) to design the string matching algorithm, which can evaluate the proper nouns and improve the word similarity of the words with less statistical information. Although STS is an effective method but still has some problems because STS is a pair matching method with a high time complexity even it only computes the meaningful words for each word pair in the similarity matrix, and STS uses only corpus-based word similarity but cannot solve the problem of word polysemy.

Omiotis [50] is an algorithm based on WordNet to achieve text relatedness and WSD and uses the part-of-speech (POS) and various semantic relations (e.g., synonymy, antonymy, hypernymy, hyponymy, holonymy, meronymy, metonymy, etc.) between words. Omiotis improves STSS evaluation by reducing the ambiguity of a word pair and expanding the semantic relations to the word pair. However, Omiotis is a high time complexity method because it uses many semantic relations and redundant matching processes to compute word similarities. SyMSS [37] also uses

WordNet-based word measures and the parse tree to address WSD and evaluate STSS. SyMSS obtains the parse trees of short-texts through the grammar parser and depends on the structure of parse trees to compute the word similarities where the two words have the same syntactic role in the syntactic structures. The novel idea in SyMSS is to consider the syntactic information of short-texts and assign weights to different syntactic roles. It uses the syntactic information in WSD and reduced the word matching, which reduces the time complexity of SyMSS, but it is based on the structure of parse tree and roughly matches words, which leads to inaccurate results where the two short-texts have the same meaning but use different syntactic structures.

STASIS [52] produces the basis of a semantic vector space via the union of the words in two sentences; it combines WordNet-based and corpus-based word similarities to compute two semantic vectors by matching the basis. STASIS evaluates two vectors with vector space model (VSM) to obtain the semantic similarity between the sentences, which depends on the syntactic rule, word order, to design word order similarity measurements. STASIS combines VSM semantic similarity and word order similarity methods to the sentence similarity measurements. However, STASIS does not remove the stop words and meaningless words, resulting in inaccurate similarities. In sum, Omiotis and SyMSS reduce the ambiguity between words using the syntactic information, POS and parse tree, respectively, to match words with the same syntactic role. We can determine the syntactic information not only to help STSS measurements reduce the negative effects of word polysemy but also to improve the efficiency.

3. Methodology

To develop an effective natural language search for the proposed CBR systems, the first issue is transferring natural languages into semantic representations for knowledge acquisition and computer reasoning, which ensures that domain knowledge can be acquired in an easy and accurate way and be understood by both machines and humans. The system might not have a direct solution for new problems; thus, the second issue is developing an appropriate semantic-based retrieval method, which ensures that related knowledge can be found to solve the target problem. If the system did not have a stable case-base at the beginning, the third issue is improving the quality of the case-base constantly, which ensures that users' needs can always be satisfied. Among existing AI technologies, STSS is ideal for realizing natural language search. STSS has not only a powerful flexibility of knowledge acquisition but also satisfactory content analyzing. Thus, we adopt the WordNet lexicon and Stanford parser, as the means to acquire domain knowledge from natural language descriptions and use the proposed semantic measurement as the retrieval method. Consequently, the following sections focus on three major topics, the novel framework of the proposed CBR system, the proposed mechanism of a case retrieval agent, and the evaluation of semantic similarity. The entire mechanism of the proposed CBR system is shown in Section 3.1, the method of case retrieval is shown in Section 3.2, and the design of the STSS measure is presented in Section 3.3.

3.1. The framework of the proposed CBR system

To imitate the intelligent thinking of human beings, the case-base acts as "brain" in the proposed system, storing knowledge in the form of cases; the semantic-based case retrieval agent acts as a human being' understanding of natural language. RTE was used to obtain the existing cases, which caused the highly similar problems as those caused. If RTE cannot work well, STSS was used to search for possible solutions with high relevance to the target problem. Relevance ranking acts as human information filtering; and collaborative filtering learns new cases or enhances the reliability of cases in the same way as human beings learn. To improve the performance of the proposed system, a semantic-based case retrieval agent and a collaborative filtering mechanism were integrated. The entire mechanism of the proposed CBR system involves seven major steps (Fig. 2):

Step 1. Retrieve: Given the natural language description of the target problem, the case retrieval agent retrieves the most similar cases from the case-base to solve the target problem. However, if appropriate cases do not exist, the agent generates new cases by matching meaning-related solutions in the case base; the detailed retrieval process is shown in Section 3.2.
Step 2. Reuse: All of the retrieved cases that are past experiences and existing solutions from case-bases that provide possible solutions to the user immediately and even address the target problem directly. If the retrieved cases are not able to solve the target problem, a user can produce a new solutions based on the existing cases.
Step 3. Ranking: The case retrieval agent presents the retrieved cases. However, the cases do not have a suitable order before ranking. The ranking mechanism arranges the retrieved cases according to semantic similarity and collaborative filtering,

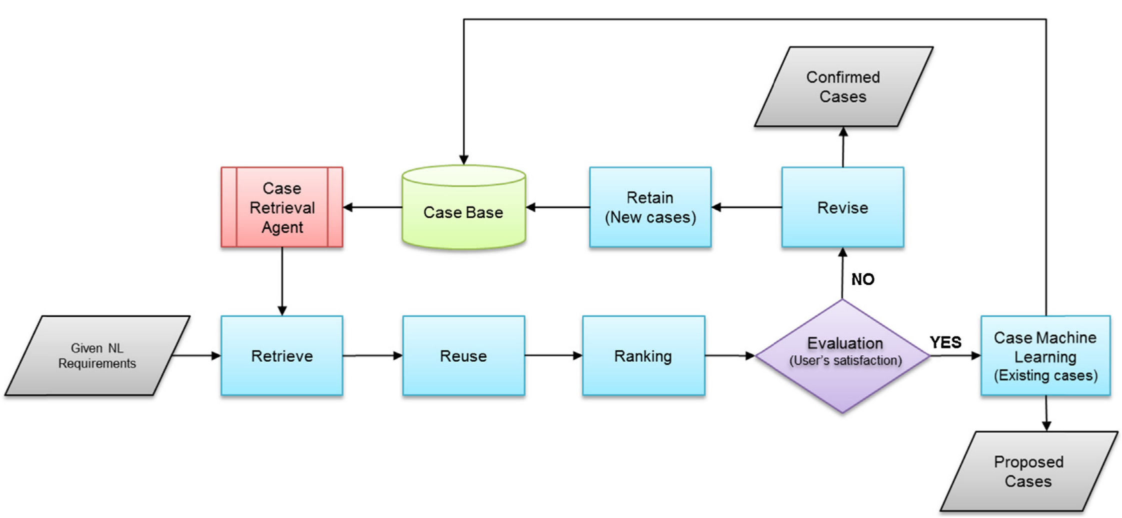


Fig. 2. The work mechanism of the proposed CBR system.

which causes the possible solutions to have a greater exposure rate and helps users achieve the desired answers in a short time.

Step 4. Evaluation: The user receives the ranking list of most similar cases and evaluates "Are the recommended solutions able to meet the user's needs?" If the answers are able to solve the target problem directly, the retrieved cases are observed as effective cases. If not, the user can find more suitable solutions after a revision step.

Step 5. Revise: An adapted solution to the target problem can be generated by modifying the existing cases if the recommended solutions cannot satisfy the user' needs. The most similar cases can be good references for revise if the repository does not contain a solution to the target problem.

Step 6. Retain: If an adapted solution has been successfully applied to the target problem, it will become a new case. The suitable solutions are created by human beings, which are more reliable than computer reasoning; thus, the proposed system stores the resulting new cases in a repository that can improve the quality of the case base and the performance of case retrieving.

Step 7. Case-based machine learning: If a suggested case is applied to the target problem, indicating that the retrieved case is effective. The suggested case selected by most users that should have higher priority in the ranking step when other users ask the similar problems next time. Specifically, if a case was selected to solve the target problem by a user, the ranking score of this case has to add 1. Then, the higher score represents the higher priority in ranking step. Thus, this simple collaborative filtering mechanism can constantly improve the ranking result, and it is a useful mechanism for case-based machine learning.

3.2. Design of the proposed case retrieval agent

In CBR systems, the most crucial function is case retrieval because the case-base contains many previous cases. When a new task takes place, an effective system must provide solutions with high relevance to the target problem. Thus, the core issue of CBR is retrieving the most appropriate past cases, which is a crucial step determined mainly based on semantic similarity. However, the retrieved cases might not be similar enough so that inaccurate results were presented.

To address these problems, we designed a case retrieval agent that mainly relied on RTE supplemented by the STSS mechanism, in other words, the proposed case retrieval agent is based on a sentence-level semantic similarity measure, which has both RTE and STSS abilities. This process is shown in Fig. 3 and briefly explained as follows:

First, we used STSS to play the role of RTE, as described elsewhere [37,49,50]. The experiments of the approaches were in the form of RTE and were tuned an appropriate similarity threshold for achieving the most satisfactory accuracy. RTE is superior to STSS for use as the retrieval method to provide the most similar "existing" cases because the similarity scores of these existing cases are above an appropriate threshold. Thus, we infer that the problems of the cases are highly similar to the target problem. That is, it is able to retrieve the most similar existing cases by identifying "Does the inputted requirement exist in the repository in other forms with the same meaning?" If the inputted requirement (or called user query) is an existing problem that is identified using RTE, the system would return the solutions of existing similar problems. RTE finds that the similar requirement does not exist in the case base if the similarity scores of all cases are lower than the threshold, i.e., the requirement (user query) is a new problem. Thus, the case retrieval agent uses STSS to find and match potential solutions that are meaning-related with the user's natural language requirement. It also generates and returns new cases for the target problem.

3.3. The design of the POS-based short-text semantic similarity measure

We proposed a novel STSS algorithm to compute the maximum similarity of two short-texts, called POS-based STSS (P-STSS), which is based on the WordNet-based word measures. In the proposed method, natural language is first turned into a semantic representation for semantic evaluation. Each entry included a word set with a POS tag. The POS is able to decrease the ambiguity in context and to restrict the meaning for each word under the corresponding POS tag. The syntactic information supplies more clear meaning for each word in a context and make knowledge acquisition and representation more accurate [37]. In terms of case retrieval, this study emphasized that the proposed system is a natural language search, and the most appropriate cases can be searched according to the context and related information through semantic similarity, which ensures that a system can accurately understand users' intentions and provide correct solutions.

The proposed algorithm is divided into four major functions (shown as Algorithms P_1–P_4). The first part is the POS tagging by using the Stanford Parser [53], and POS simplifying. The POS tagset of the Stanford Parser is based on Penn Treebank, which included over 30 POS tags [54]. We simplified the original tagset for word similarity measures because WordNet has only nouns, verbs, adverbs, and adjectives [55]. Thus, the proposed comparison table of simplified tagset (η) is shown in Table 1.

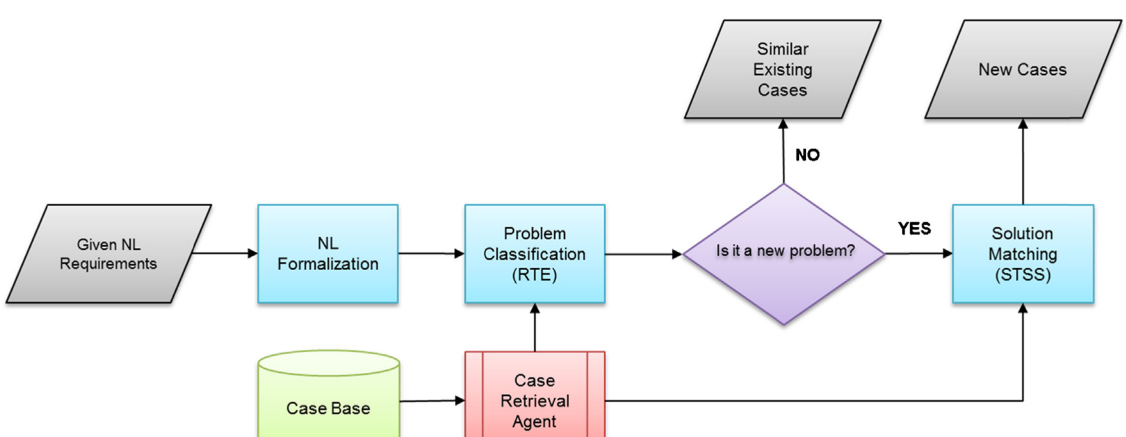


Fig. 3. The mechanism of case retrieval in the proposed CBR system.

Table 1
Simplified POS tagset.

Simplified POS	Penn Treebank POS
Noun (n)	NN, NNS, NNP, NNPS
Verb (v)	VB, VBD, VBG, VBN, VBP, VBZ
Adjective (a)	JJ, JJR, JJS
Adverb (r)	RB, RBR, RBS
Others (o)	CC, CD, DT, EX, FW, IN, LS, MD, PDT, POS, PRP, PRP$, RP, SYM, TO, UH, WDT, WP, WP$, WRB

Simplified POS Classifier (Algorithm P_1) accepts a short-text S and a simplified tagset (η), invokes the POS tagger to generate Penn Treebank POS tags, and finally returns a set of simplified POS. Word measures combined with the POS improved the performance because the POS tags matched the words, which are the same POS, in both short-texts.

Algorithm P_1. Simplified POS classifier (SPC).

INPUT: *SENT*, η
/* *SENT* is the input sentence, and η is a lookup table as Table 1*/
OUTPUT: $SimplifiedPOS_{SENT}$
$PennPOS_{SENT} \leftarrow Stanford\_Parser(S)$
FOR ALL $Tag_i \in PennPOS_{SENT}$
DO
$SimplifiedPOS_{SENT} \leftarrow LookupSimplifiedTag(\eta, Tag_i)$
END FOR
RETURN $SimplifiedPOS_{SENT}$

In *Semantic Similarity Optimization* (Algorithm P_2), words with POS of S_A and S_B can form a semantic matrix, called a POS-based coordinate matrix (PCM), which was shown as Fig. 4. In order to obtain the maximum similarity of two short-texts, we firstly assigned the row headers as the basis of the matrix. Then, we assigned the short-text with fewer words as the row headers and the other one as the column headers.

The elements of the matrix were computed using WordNet-based word measures when the two words were the same POS, in other words, we matched two words as a word pair if they had similar syntactic roles in their context, and each element represented the semantic similarity of the word pair, such as S_AW_1–S_BW_1 and S_AW_3–S_BW_1 were the word pairs of Noun (Fig. 5).

The corresponding word measures (*WordSimilarity*) then quantify the semantic information and extract semantics from these word pairs; WordSimilarity is referenced in [37]. WordSimilarity used PATH, WUP, LCH, RES, JCN and LIN to measure the similarity between nouns and verbs, and employed VECTOR to measure similarity between adjectives and adverbs. In addition,

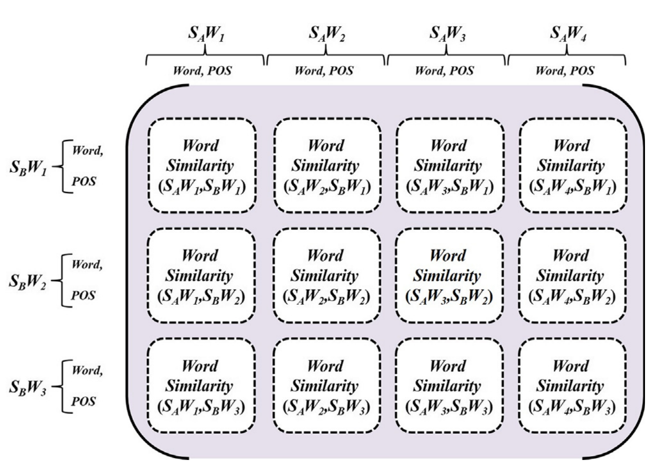


Fig. 4. Diagram of POS based coordinate matrix.

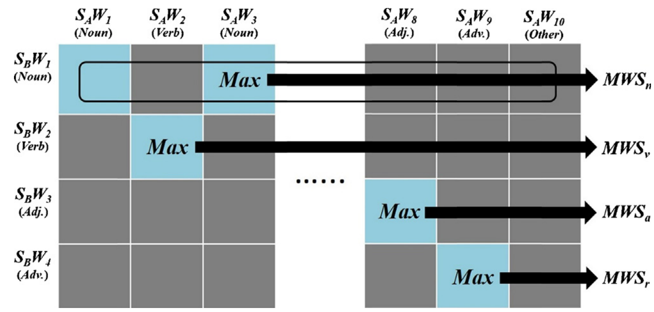


Fig. 5. Diagram of semantic similarity optimization.

we used the string similarity proposed in [49] to measure the similarity between other POSs.

When all elements were computed, the maximal element of each row was reserved. For example, S_AW_3–S_BW_1 represents the maximum word similarity (MWS) for its row (Fig. 5). The MWSs of other rows are followed by the same computation process as above. Thus, these MWSs denote a semantic array of the two short-texts. To gather the contribution of each MWS, we sum up all the MWSs to obtain the optimal semantic similarity, MWS_{SUM}.

Algorithm P_2. Semantic similarity optimization (SSO).

INPUT: $SimplifiedPOS_A$, $SimplifiedPOS_B$
/* Simplified POS sets of S_A, S_B */
OUTPUT: MWS_{SUM}
/* The Sum of Maximum Word Similarity of S_A, S_B */
$ROW \leftarrow MAX(SimplifiedPOS_A, SimplifiedPOS_B)$
$COL \leftarrow MIN(SimplifiedPOS_A, SimplifiedPOS_B)$
FOR ALL $c_x \in COL$**DO**
FOR ALL $r_y \in ROW$ **DO**
IF $c_x.pos$ **EQUAL** $r_y.pos$ **THEN**
$SA[x] \leftarrow MAX(SA[x], WordSimilarity(c_x.word, r_y.word, pos))$
END IF
END FOR
END FOR
FOR 0 **TO**$|COL|$
$MWS_{SUM} \leftarrow MWS_{SUM} + SA[x]$
END FOR
RETURN MWS_{SUM}

Once the semantic similarity of the two short-texts was optimized, *Semantic Similarity Normalization* (Algorithm P_3) was used to normalize the MWS_{SUM} to result in a STSS score of 0–1. The design of normalization function was using the reciprocal harmonic mean and the lengths of S_A and S_B. Then, Algorithm P_3 shows the whole process of normalization, and which finally returns a normalized coefficient (NC).

Algorithm P_3. Semantic similarity normalization (SSN).

INPUT: MWS_{SUM}
/* The Sum of Maximum Word Similarity of S_A, S_B */
OUTPUT: *NC*
/* Normalized Coefficient of S_A, S_B */
$Length_A \leftarrow Counting\_Words(S_A)$
$Length_B \leftarrow Counting\_Words(S_B)$
$NC \leftarrow (Length_A + Length_B)/(2{*}Length_A{*}Length_B)$
RETURN *NC*

The Algorithm P_4 depicts the details of the proposed algorithm, P-STSS. First, given two raw short-texts and a POS comparison table (η). The Algorithm P_1 was invoked to generate simplified POS tags for each short-text. Then, Algorithm P_2 then accepted the two short-texts with simplified POS tags, and returned the maximum correlation of the two short-texts, MWS_{SUM}. The Algorithm P_3 then provided a coefficient to normalize the MWS_{SUM}. Finally, we multiplied the MWS_{SUM} by the NC to obtain the STSS score ($STSS_{A,B}$) between 0 and 1.

Algorithm P$_4$. POS-based short-text semantic similarity measure.

INPUT: S_A, S_B, η
/* Raw short-text A, B and η (see Table 1) */
OUTPUT: $STSS_{AB}$
/* Short-text semantic similarity between S_A, S_B */
$SimplifiedPOS_A \leftarrow SPC\ (S_A, \eta)$
$SimplifiedPOS_B \leftarrow SPC\ (S_B, \eta)$
/* SPC (Algorithm P$_1$) returns the information of simplified POS. */
$MWS_{SUM} \leftarrow SSO\ (SimplifiedPOS_A, SimplifiedPOS_B)$
/* SSO (Algorithm P$_2$) returns the sum of maximum word similarity between S_A and S_B. */
$NC \leftarrow SSN\ (S_A, S_B)$
/* SSN (Algorithm P$_3$) returns the normalized coefficient between S_A and S_B. */
$STSS_{A,B} \leftarrow MSW_{SUM} * NC$
RETURN $STSS_{A,B}$

4. Performance test of the case retrieval agent

Our case retrieval agent combined STSS and RTE mechanisms for searching similar cases. Thus, we have to evaluate the performance of our P-STSS algorithm, for STSS and RTE tasks. Two datasets were used in the study. For STSS task, we used the dataset proposed by Li et al. [52] to enable comparison with other existing approaches. Also, we used a large dataset, Microsoft Paraphrase Corpus [56], in order to evaluate our method for RTE task.

The dataset described by Li et al. [52] contains 65 sentence pairs created from 65 noun pairs, which are defined in the Collins Cobuild dictionary. The definitions in the Cobuild dictionary are written in complete sentences. Thirty sentence pairs were then selected by Li et al. for evaluation. This dataset contains the average similarity scores given by 32 human judges, and the human similarity scores are provided as the mean score for each sentence pair. Li et al. used leave-one-out resampling to calculate the correlation coefficient for the judgments of each participant against the rest of the group. Thus, Li et al. pointed out the mean of human judgements is 0.825, and considered this as the upper bound. The following sentence pairs were extracted from the dataset.

1. "An automobile is a car."
"A car is a motor vehicle with room for a small number of passengers."

2. "Midday is 12 o'clock in the middle of the day."
"Noon is 12 o'clock in the middle of the day."

3. "A cemetery is a place where dead people's bodies or their ashes are buried."
"A graveyard is an area of land, sometimes near a church, where dead people are buried."

Microsoft Paraphrase Corpus [56] consists of 4076 training and 1725 test sentence pairs collected from thousands of news sources on the web over a period of 18 months. These pairs are labeled as *yes* (1) or *no* (0) by two human annotators, who determine whether the two sentences in a pair are semantically equivalent paraphrases. The agreement between human evaluators is approximately 83%, which can be considered an upper bound for an RTE task. The following sentence pairs were extracted from the dataset.

1. "The largest gains were seen in prices, new orders, inventories and exports."
"Sub-indexes measuring prices, new orders, inventories and exports increased."

2. "Ballmer has been vocal in the past warning that Linux is a threat to Microsoft."
"In the memo, Ballmer reiterated the open-source threat to Microsoft."

3. "At midnight on Wednesday, 68 percent of voters said "no" to the tax, with 97 percent of the votes counted."
"With 97 percent of precincts counted tonight, 68 percent of voters opposed the tax."

4.1. Experiments for STSS task

Pearson's correlation coefficient was used to evaluate the performance of the STSS task. First, the performances of the six measures of WordNet-based word semantic similarity, PATH, LCH, WUP, RES, JCN and LIN, were compared. We used the six measures in the P-STSS algorithm and then computed the similarities of the 30 sentence pairs proposed by Li et al. [52]. In addition, each Pearson's correlation coefficient was obtained by using the similarity scores of each method and human scores. Table 2 shows the Pearson's correlation coefficients of the six measures and the detailed results obtained by using six measures with the 30 sentence pairs. The results showed that the most satisfactory performance measure for the STSS task was PATH ($r = 0.83$), and this result is consistent with that reported by Oliva et al. [37].

Furthermore, Table 3 shows the human similarity scores along with Li [52], LSA [57], STS Meth. [49], SyMSS [37], Omiotis [50], and the proposed measure. The results shows that our proposed P-STSS using PATH achieves a high Pearson correlation coefficient of 0.83, then Li et al.'s measure achieves 0.82, LSA achieves 0.84, SyMSS achieves 0.76 and Omiotis achieves 0.86. The upper bound obtained by Li (0.82) and our method (0.83) are closest to the real upper bound (0.825). Thus, it is reasonable to say that our P-STSS algorithm involving PATH achieved excellent performance. In brief, our approach tried to identify and quantify the latent semantic

Table 2
Benchmark no. and the results compared with six WordNet-based measures of our approach.

No.	Human	PATH	LCH	WUP	RES	JCN	LIN
1	0.01	0.3	0.28	0.36	0.29	0.28	0.28
5	0.01	0.4	0.37	0.47	0.37	0.36	0.36
9	0.01	0.36	0.33	0.45	0.35	0.33	0.33
13	0.10	0.5	0.46	0.56	0.47	0.46	0.46
17	0.13	0.35	0.28	0.42	0.27	0.26	0.26
21	0.04	0.4	0.31	0.45	0.28	0.28	0.28
25	0.07	0.4	0.34	0.45	0.35	0.34	0.34
29	0.01	0.42	0.37	0.49	0.37	0.36	0.36
33	0.15	0.53	0.52	0.57	0.52	0.52	0.52
37	0.13	0.38	0.36	0.45	0.37	0.36	0.36
41	0.28	0.41	0.4	0.47	0.41	0.4	0.4
47	0.35	0.48	0.44	0.56	0.45	0.44	0.44
48	0.36	0.51	0.44	0.59	0.44	0.43	0.43
49	0.29	0.55	0.51	0.57	0.52	0.51	0.51
50	0.47	0.44	0.42	0.47	0.43	0.42	0.42
51	0.14	0.44	0.41	0.49	0.42	0.41	0.41
52	0.49	0.51	0.47	0.51	0.46	0.46	0.46
53	0.48	0.56	0.48	0.58	0.46	0.48	0.48
54	0.36	0.52	0.48	0.57	0.49	0.48	0.48
55	0.41	0.44	0.42	0.49	0.42	0.42	0.42
56	0.59	0.55	0.53	0.57	0.53	0.53	0.53
57	0.63	0.48	0.4	0.52	0.4	0.39	0.39
58	0.59	0.57	0.55	0.58	0.56	0.55	0.55
59	0.86	0.94	0.89	0.94	0.88	0.88	0.88
60	0.58	0.58	0.56	0.6	0.56	0.56	0.56
61	0.52	0.55	0.5	0.6	0.49	0.49	0.49
62	0.77	0.59	0.54	0.63	0.54	0.53	0.53
63	0.59	0.56	0.52	0.56	0.51	0.51	0.51
64	0.96	0.95	0.92	0.95	0.91	0.91	0.91
65	0.65	0.68	0.64	0.7	0.64	0.63	0.63
Pearson (r)	–	**0.83**	0.82	0.79	0.81	0.82	0.82

Table 3
Benchmark no. and the results compared with Li, LSA, STS Meth., SyMSS, Omiotis, and our approach.

No.	Human	Li	LSA	STS Meth.	SyMSS	Omiotis	Ours (PATH)
1	0.01	0.33	0.51	0.06	0.32	0.11	0.3
5	0.01	0.29	0.53	0.11	0.28	0.10	0.4
9	0.01	0.21	0.51	0.07	0.27	0.10	0.36
13	0.11	0.53	0.53	0.16	0.27	0.30	0.5
17	0.13	0.36	0.58	0.26	0.42	0.30	0.35
21	0.04	0.51	0.53	0.16	0.37	0.24	0.4
25	0.07	0.55	0.60	0.33	0.53	0.30	0.4
29	0.01	0.33	0.51	0.12	0.31	0.11	0.42
33	0.15	0.59	0.81	0.29	0.43	0.49	0.53
37	0.13	0.44	0.58	0.20	0.23	0.11	0.38
41	0.28	0.43	0.58	0.09	0.38	0.11	0.41
47	0.35	0.72	0.72	0.30	0.24	0.22	0.48
48	0.36	0.65	0.62	0.34	0.42	0.53	0.51
49	0.29	0.74	0.54	0.15	0.39	0.57	0.55
50	0.47	0.68	0.68	0.49	0.35	0.55	0.44
51	0.14	0.65	0.73	0.28	0.31	0.52	0.44
52	0.49	0.49	0.70	0.32	0.54	0.60	0.51
53	0.48	0.39	0.83	0.44	0.52	0.5	0.56
54	0.36	0.52	0.61	0.41	0.33	0.43	0.52
55	0.41	0.55	0.70	0.19	0.33	0.43	0.44
56	0.59	0.76	0.78	0.47	0.43	0.93	0.55
57	0.63	0.7	0.75	0.26	0.50	0.61	0.48
58	0.59	0.75	0.83	0.51	0.64	0.74	0.57
59	0.86	1	1	0.94	1	1	0.94
60	0.58	0.66	0.83	0.6	0.63	0.93	0.58
61	0.52	0.66	0.63	0.29	0.39	0.35	0.55
62	0.77	0.73	0.74	0.51	0.75	0.73	0.59
63	0.59	0.64	0.87	0.52	0.78	0.79	0.56
64	0.96	1	1	0.93	1	0.93	0.95
65	0.65	0.83	0.86	0.65	0.36	0.82	0.68
Pearson (r)	–	0.82	0.84	0.85	0.76	0.86	**0.83**

relationship among syntaxes and words; and our idea yielded satisfactory results for the STSS task.

4.2. Experiments for RTE task

We used P-STSS as a RTE method, a training set to tune the most satisfactory similarity threshold score, and the test set to evaluate the performance of our method against this most satisfactory similarity threshold. To determine whether two sentences with identical meanings, we used different similarity thresholds ranging from 0 to 1 with an interval of 0.1. After training, we found that the most satisfactory score of the similarity thresholds of the six word measures was 0.6, and this similarity threshold (0.6) was used in the evaluation with the test set. According to the results (Table 4), our P-STSS using PATH achieves the most satisfactory accuracy 72.75% that outperforms other word measures, for example, the accuracies obtained using LCH, WUP, RES, JCN, and LIN were 72.12%, 70.67%, 71.94%, 71.65%, and 71.65%, respectively.

This experiment compares the performance in several categories by test dataset (see Table 4): (1) two baselines, a random selection and a VSM-cosine-based measure with TF-IDF weighting; (2) corpus-based approaches: the PMI-IR [58], the LSA [57], STS Meth. [49]; (3) lexicon-based approaches, including Mihalcea et al. [59], SyMSS (JCN and Vector) [37], Omiotis [50], and LG [60]; (4) machine-learning-based approaches, including Wan et al. [61], Zhang and Patrick [62], and Qiu et al. [63], which is a SVM approach [64]. The results in Table 4 showed that our algorithm outperformed most compared methods. However, it is not superior to the machine learning-based method reported by Wan et al. [61].

Overall, the results showed that our algorithm is an excellent method with a threshold of 0.6, which is a reasonable range to determine whether a sentence pair is a paraphrase. In addition, our P-STSS using the six word measures displayed their most satisfactory performance at a threshold of 0.6. This means that our P-STSS is a stable algorithm no matter which one WordNet-based word measure was used, the proposed approach had the most satisfactory result under the same condition.

4.3. Conclusions for performance test of the case retrieval agent

Two experiments were conducted to evaluate the performance of the case retrieval agent because the agent included STSS and RTE mechanisms. In the experiment for the STSS task, our P-STSS algorithm using PATH (a WordNet-based word measure) achieved an excellent Pearson correlation (r = 0.83) for 30 sentence pairs of the dataset of Li et al. In the experiment for the RTE task, our method surpasses the most of existing approaches and limits the best performance to a reasonable range of thresholds. We used 0.6 as the reasonable threshold for the six word measures (the most satisfactory accuracy of 72.75% was obtained using PATH) for the RTE task. The two experimental results showed that our P-STSS is a stable algorithm because (1) the six word measures had their most satisfactory accuracy at the threshold of 0.6 in the RTE task and (2) our algorithm had the most satisfactory performance when

Table 4
Results of the proposed and competitive methods on the Microsoft Research Paraphrase Corpus.

Category	Metric	Best threshold	Accuracy	Precision	Recall	F-measure
Corpus-based	PMI-IR	–	69.90	70.20	95.20	81.00
	LSA	–	68.40	69.70	95.20	80.50
	STS Meth.	0.6	72.64	74.65	89.13	81.25
Lexicon-based	SyMSS (JCN)	0.45	70.87	74.70	84.17	79.02
	SyMSS (Vector)	0.45	70.82	74.15	90.32	81.44
	Omiotis	–	69.97	70.78	93.40	80.52
	LG (WUP)	0.6	71.02	73.9	91.07	81.59
Machine learning-based	Wan et al.	–	75.00	77.00	90.00	83.00
	Z&P	–	71.90	74.30	88.20	80.70
	Qiu et al.	–	72.00	72.50	93.40	81.60
Baselines	Random	–	51.30	68.30	50.00	57.80
	VSM	0.5	65.40	71.60	79.50	75.30
Ours	**PATH**	**0.6**	**72.75**	**73.86**	**91.37**	**81.68**
	LCH	0.6	72.12	74.81	87.53	80.68
	WUP	0.6	70.67	71.07	94.25	81.03
	RES	0.6	71.94	74.79	87.18	80.51
	JCN	0.6	71.65	74.81	86.49	80.23
	LIN	0.6	71.65	74.81	86.49	80.23

P-STSS was used with PATH. From a practical viewpoint, our approach using POS information reduces word matching, and the time complexity obtained using our approach is lower than that obtained using existing methods.

5. Case study

In this section, a case study of an online bookstore based on the proposed system is given. Then, a number of experiments are organized to study the performance of the proposed CBR system.

5.1. Online bookstore

Enterprises provide large amounts of product information to meet the various needs of consumers and create more business opportunities. However, the increase in information and the rapid expansion of e-commerce creates an overload of information. Consumers have to spend a significant amount of time browsing online stores to find the product they need. One solution to overcome the problem is to develop an intelligent recommendation system that retrieves the information a consumer desires and helps him determine which one to buy. The famous e-commerce site, Amazon.com: books, gives details of the text and purchase information in a page for each book. Requests can be directly entered by customers, and the recommended books are provided to customers. However, we discovered that the recommendations from Amazon are not good when an inputted query is in the form of a complete sentence, e.g., "I want to go to beach, which place is the most popular in Spain?" Several combinations of keywords, {I, want, beach, most}, {I, want, go, most}, {I, want, go, beach} and {go, beach, most}, are separated from the natural language query, which enables the system to retrieve corresponding recommendations based on a keyword search (see Fig. 6).

Keyword matching is usually adopted when calculating the similarities between nonnumeric objects, which leads to very limited views of users' ideas if the inputted words are not exactly the same as the items you desired. The results are often not good. Thus, all of the necessary information must be given if the existing retrieval system is used. Otherwise, the case retrieval fails if some important information is missing or incomplete. To assuage

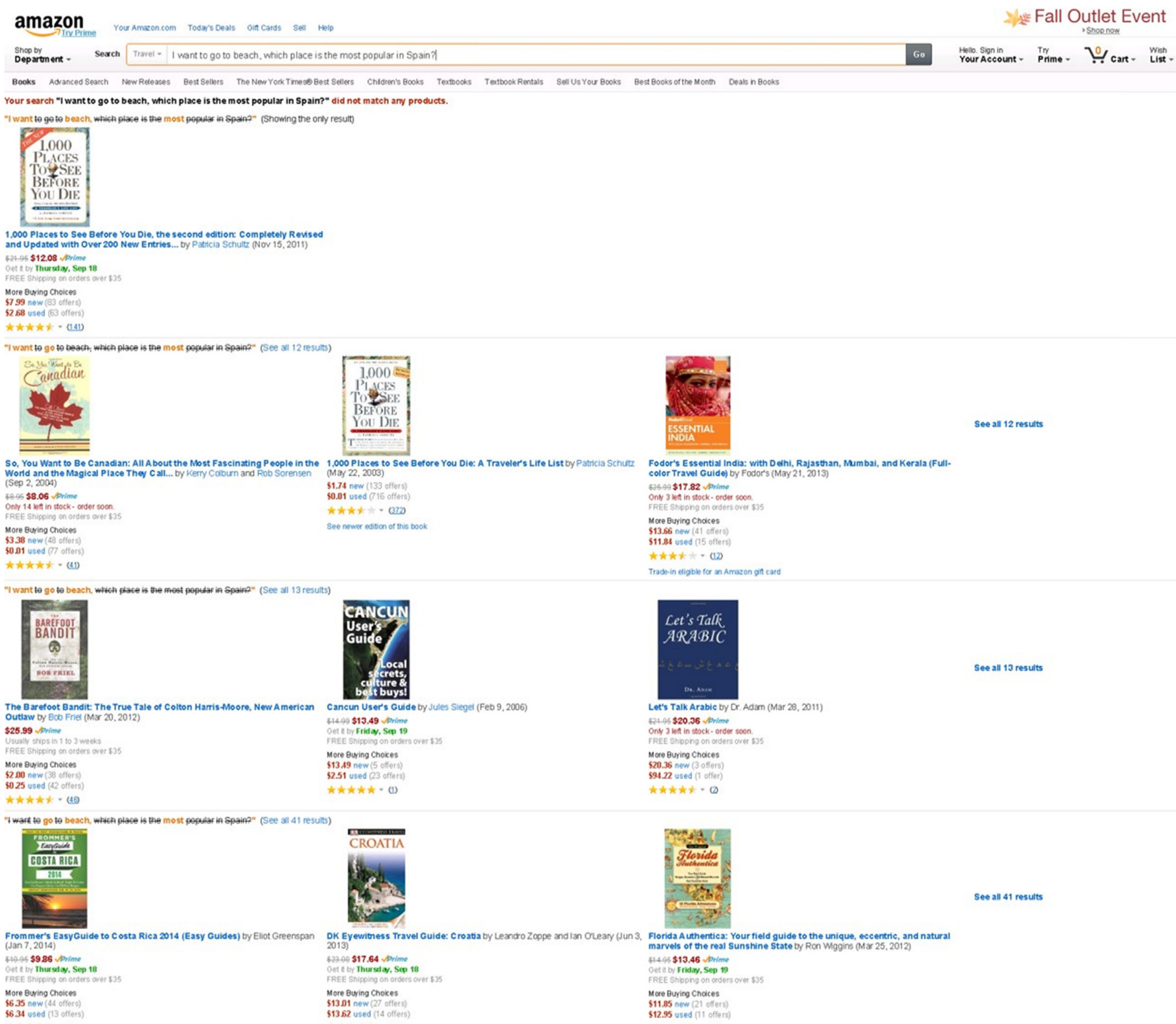


Fig. 6. An example of given a natural language query to Amazon.

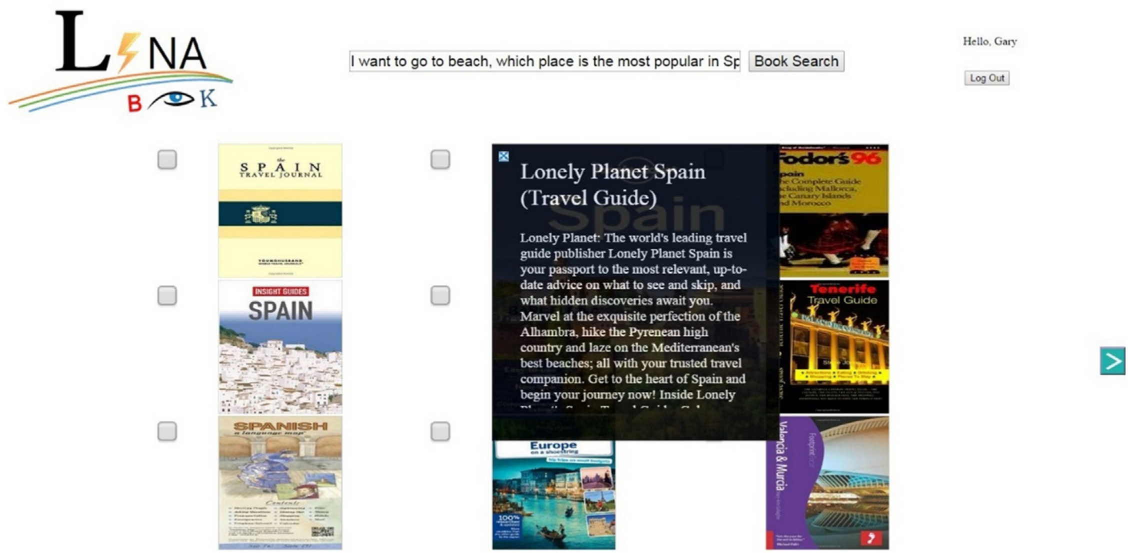


Fig. 7. The user interface of the proposed system.

customers' search engine fatigue and turn their search experience into an effective, positive, more human experience, Natural language search can be an effective solution that allows customers to express their requirements in their own words. They also have the ability to use the internet with reduced application conditions, which allows the system to conduct case retrieval without complete and exact information. Even if customers do not know detailed information about their objective and only know some relevant clues, they can still retrieve the desired cases.

To provide a novel advanced recommendation system and objectively evaluate the performance of the proposed CBR system, this paper applied our mechanism in an online bookstore, where users can input their requirements in the form of a complete or incomplete sentences or keywords. Then, the system compares user's queries to the abstract of books and returns the recommended books that meet the user's needs. The user interface of the proposed system is shown in Fig. 7, and nine recommended books are provided for each page.

5.2. Experimental design

In the proposed CBR system, 1200 books were retrieved on June 8, 2015 from Amazon.com: books (category: "Travel" and language: "English" are the defaults). Two datasets were provided for experiments, and each dataset had 10 queries. The queries of the first dataset were described in natural language and were collected from three people; and the queries of the second dataset were keywords extracted from the search suggestions of the Amazon search engine. The queries of the two datasets are related to "Spain Travel". Thirty volunteers participated in the experiments, they are graduate students from department of engineering science of a research-based university in Tainan, Taiwan. To more accurately and objectively evaluate its performance, we also execute the same experiments using the string similarity proposed in [49] and then compare their performances with the proposed method. We set up the proposed system using semantic similarity as the experimental group (EG), which includes collaborative filtering. In addition, the proposed system using string similarity excludes collaborative filtering as control group 1 (CG1) and Amazon as the control group 2 (CG2). Ten people in the EG execute all queries of the two dataset in five rounds. In particular, the priority score of the selected cases has to update after EG finished each round. For example, the proposed system update the priority score of all selected cases according to the result of first round before second round test. Before third round test, the proposed system also update the priority score of all selected cases according to the result of first and second round, and so on. On the other hand, ten people in CG1 only partake in one test round, and the other ten people in CG2 test Amazon once. All participants input the 10 natural language queries (Table 5) and 10 keyword queries (Table 6) and, to avoid conscious and subconscious bias and obtain reasonable accuracy, we blinded participants. That is, they do not know which type of system they are using when they perform the experiments, except the Amazon group.

Two effective methods have been successfully used [65–68], and adopted to evaluate the information retrieval results, i.e., (1) top-k precision is the ratio between the correct top-k elements included in the returned top-k elements and k, which is the main

Table 5
Natural language queries for experiments of online bookstore.

No.	Natural language queries
1.	What kind of food is the most delicious in Barcelona?
2.	How to go to Madrid from Barcelona?
3.	I want to go to beach, which place is the most popular in Spain?
4.	When is running of the bulls?
5.	Where are Gaudi buildings in Barcelona?
6.	Do you have recommendations for Flamenco shows?
7.	Where are the famous wineries in Spain?
8.	What is the top place in Seville?
9.	Which hotel I can choose in Madrid?
10.	I want to realize the culture of Spain.

Table 6
Keyword queries for experiments of online bookstore.

No.	Keyword queries
1.	Spain travel guide
2.	Spain travel books
3.	Spain tour book
4.	Spain travel guide 2015
5.	Spain tourist guide
6.	Rough guide to Spain
7.	Spain travel kids
8.	Spain travel writing
9.	Spain travel family
10.	Spain travel memoir

quality indicator of the algorithm because it reflects the number of correct values obtained when obtaining a top-*k* list. (2) The mean average precision (MAP) assumes that the user is interested in finding many relevant documents for each query and summarizes rankings from multiple queries via the average precision scores.

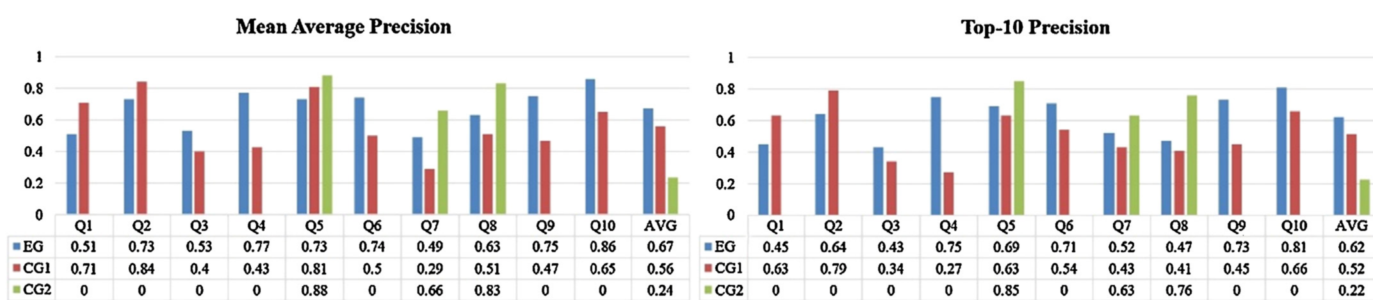
5.3. Experimental results of natural language queries

The results in Fig. 8 demonstrate that the proposed method outperforms the control group with string similarity and the e-commerce system of Amazon in one round test. MAP and top-10 precision more obviously show the advantages of the proposed system over the compared systems. The performance of the proposed method (EG, MAP: 0.67, top-10 precision: 0.62) was improved considerably over the compared systems, which are based on a string similarity method (CG1, MAP: 0.56, top-10 precision: 0.52) and the online bookstore of Amazon (CG2, MAP: 0.24, top-10 precision: 0.22). The proposed method outperforms the compared systems is because keyword matching is usually adopted to measure the similarity of conceptual entities, which leads to very limited concepts, and related words cannot be discovered because the inputted words are not exactly the same as the desired cases, which seriously limits the intelligence of CBR system.

The proposed system that uses collaborative filtering had a growth precision that improved after each of the five rounds, and the experimental results are shown in Fig. 9. The averages of MAPs of the five rounds were 0.67, 0.82, 0.87, 0.90, and 0.93; and the averages of the top-10 precision of the five rounds were 0.62, 0.79, 0.83, 0.85, and 0.88. Thus, the results showed a trend of improvement. The experimental results showed that the integrated collaborative filtering in the proposed system facilitated the system to improve its performance through case-based machine learning.

5.4. Experimental results of keyword queries

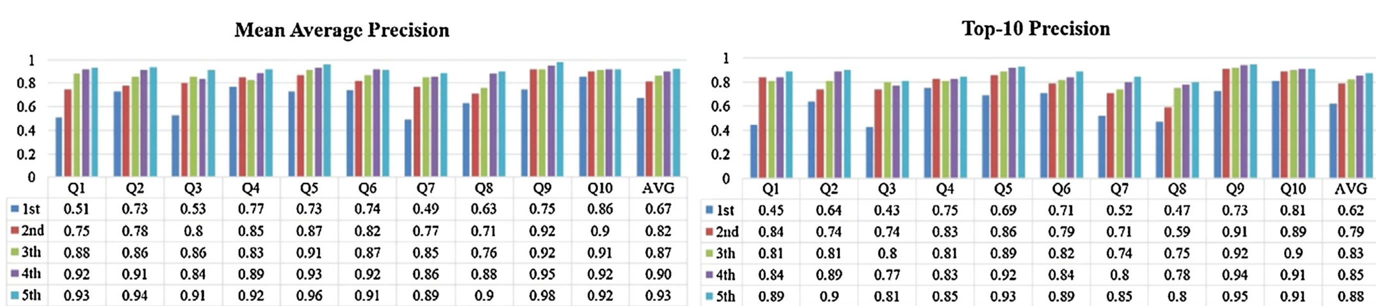
For a keyword search task, the results in Fig. 10 demonstrate that the online bookstore of Amazon (CG2, MAP: 0.66, top-10 precision: 0.62) outperformed our system with semantic similarity (EG, MAP: 0.61, top-10 precision: 0.57) and the proposed system with string similarity (CG1, MAP: 0.55, top-10 precision: 0.53). However, EG outperformed the CG1. Because the design of the Amazon search engine is a keyword-based search with a strong collaborative filtering mechanism, a large amount of rating data was collected for optimizing the recommendation ranking. Amazon outperformed EG and CG1 in the one-round test.



	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	AVG
EG	0.51	0.73	0.53	0.77	0.73	0.74	0.49	0.63	0.75	0.86	0.67
CG1	0.71	0.84	0.4	0.43	0.81	0.5	0.29	0.51	0.47	0.65	0.56
CG2	0	0	0	0	0.88	0	0.66	0.83	0	0	0.24

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	AVG
EG	0.45	0.64	0.43	0.75	0.69	0.71	0.52	0.47	0.73	0.81	0.62
CG1	0.63	0.79	0.34	0.27	0.63	0.54	0.43	0.41	0.45	0.66	0.52
CG2	0	0	0	0	0.85	0	0.63	0.76	0	0	0.22

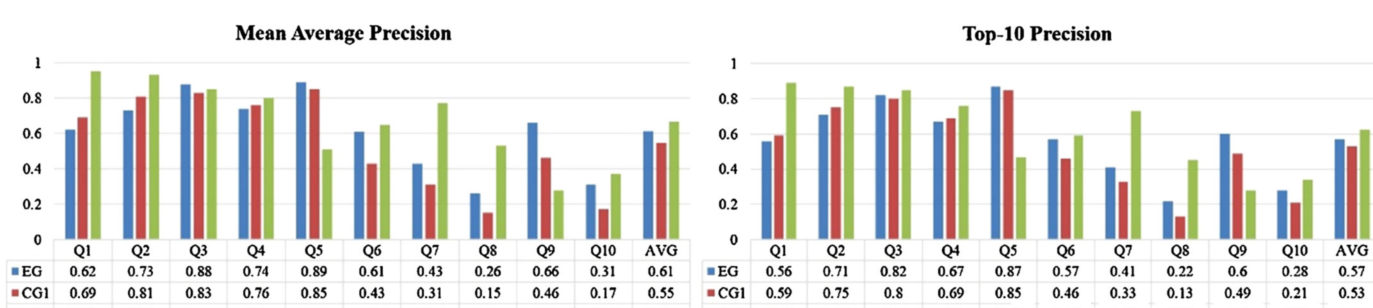
Fig. 8. MAP and top-10 precisions of the proposed method and two competitive systems when using natural language queries.



	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	AVG
1st	0.51	0.73	0.53	0.77	0.73	0.74	0.49	0.63	0.75	0.86	0.67
2nd	0.75	0.78	0.8	0.85	0.87	0.82	0.77	0.71	0.92	0.9	0.82
3th	0.88	0.86	0.86	0.83	0.91	0.87	0.85	0.76	0.92	0.91	0.87
4th	0.92	0.91	0.84	0.89	0.93	0.92	0.86	0.88	0.95	0.92	0.90
5th	0.93	0.94	0.91	0.92	0.96	0.91	0.89	0.9	0.98	0.92	0.93

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	AVG
1st	0.45	0.64	0.43	0.75	0.69	0.71	0.52	0.47	0.73	0.81	0.62
2nd	0.84	0.74	0.74	0.83	0.86	0.79	0.71	0.59	0.91	0.89	0.79
3th	0.81	0.81	0.8	0.81	0.89	0.82	0.74	0.75	0.92	0.9	0.83
4th	0.84	0.89	0.77	0.83	0.92	0.84	0.8	0.78	0.94	0.91	0.85
5th	0.89	0.9	0.81	0.85	0.93	0.89	0.85	0.8	0.95	0.91	0.88

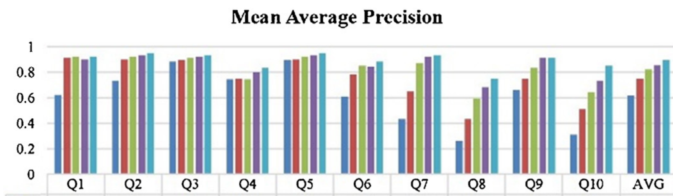
Fig. 9. MAP and top-10 precisions of the proposed method for five rounds when using natural language queries.



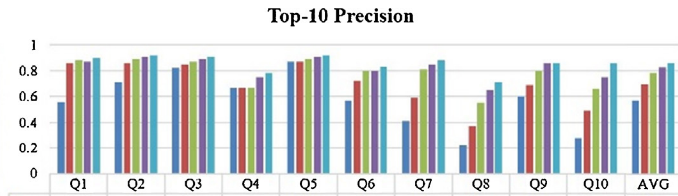
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	AVG
EG	0.62	0.73	0.88	0.74	0.89	0.61	0.43	0.26	0.66	0.31	0.61
CG1	0.69	0.81	0.83	0.76	0.85	0.43	0.31	0.15	0.46	0.17	0.55
CG2	0.95	0.93	0.85	0.8	0.51	0.65	0.77	0.53	0.28	0.37	0.66

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	AVG
EG	0.56	0.71	0.82	0.67	0.87	0.57	0.41	0.22	0.6	0.28	0.57
CG1	0.59	0.75	0.8	0.69	0.85	0.46	0.33	0.13	0.49	0.21	0.53
CG2	0.89	0.87	0.85	0.76	0.47	0.59	0.73	0.45	0.28	0.34	0.62

Fig. 10. MAP and top-10 precisions of the proposed method and two competitive systems when using keyword queries.



	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	AVG
1st	0.62	0.73	0.88	0.74	0.89	0.61	0.43	0.26	0.66	0.31	0.61
2nd	0.91	0.9	0.89	0.75	0.9	0.78	0.65	0.43	0.75	0.51	0.75
3th	0.92	0.92	0.91	0.74	0.92	0.85	0.87	0.59	0.83	0.64	0.82
4th	0.9	0.93	0.92	0.8	0.93	0.84	0.92	0.68	0.91	0.73	0.86
5th	0.92	0.95	0.93	0.83	0.95	0.88	0.93	0.75	0.91	0.85	0.89



	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	AVG
1st	0.56	0.71	0.82	0.67	0.87	0.57	0.41	0.22	0.6	0.28	0.57
2nd	0.86	0.86	0.85	0.67	0.87	0.72	0.59	0.37	0.69	0.49	0.70
3th	0.88	0.89	0.87	0.67	0.89	0.8	0.81	0.55	0.8	0.66	0.78
4th	0.87	0.91	0.89	0.75	0.91	0.8	0.85	0.65	0.86	0.75	0.82
5th	0.9	0.92	0.91	0.78	0.92	0.83	0.88	0.71	0.86	0.86	0.86

Fig. 11. MAP and top-10 precisions of the proposed method for five rounds when using keyword queries.

The experimental results of the keyword search are shown in Fig. 11, and the results are similar to those described in Section 5.3. The proposed system that used collaborative filtering had a growth precision that improved after each of the five rounds. The averages of MAPs of the five rounds were 0.61, 0.75, 0.82, 0.86, and 0.89; and the averages of the top-10 precision over the five rounds were 0.57, 0.70, 0.78, 0.82, and 0.86. Thus, the results showed a trend of improvement. The experimental results showed that the integrated collaborative filtering in the proposed system improved the system performance for the keyword search task. Although the Amazon outperformed our system at the beginning, our system continued to growth through collaborative filtering and outperformed Amazon after the second round.

5.5. Discussions for the case study of an online bookstore

In the two experiments, the proposed method outperforms the compared systems is because our system is adopted to measure the similarity of conceptual entities by semantic techniques, which leads to related words can be discovered because the inputted words are similar as the desired cases. The proposed method, P-STSS, allows the system to search for relevant concepts, which can make full use of past experiences stored in the case-base using semantic techniques to analyze the user's intentions and the content of products. Thus, the proposed system can work perfectly under conditions of incomplete queries and natural language requirements that exceeds the limitations of existing CBR systems. The reason that the proposed retrieval method outperforms the compared methods lies in the synthetically considered syntax and semantics is a good direction for improving short-text semantic similarity measurements, and the proposed combining design is appropriate. If a new problem is presented by the user and a case-base does not exist, a corresponding answer to the problem is produced by the proposed agent. Thus, the case retrieval agent can provide possible solutions using computer reasoning and semantic similarity. Compared with CG1, the proposed semantic-based retrieval algorithm solved a new problem or obtain a satisfactory start for the initial system (see Figs. 8 and 10).

In addition, the proposed mechanism with collaborative filtering can improve the performance of recommendation based on existing cases or the cases that are automatically generated by the proposed retrieval method. If a system does not have a self-learning mechanism, it cannot become more powerful. Although our case retrieval agent has the ability to determine possible solutions to the target problem, the system cannot improve its case retrieval performance and cannot provide better recommendations the next time or for other customers when the case-base is not strong enough or new problems are constantly introduced. To solve the problem, case-based machine learning can be realized through collaborative filtering, which increases the priority of the cases that are commonly selected by customers and increases the exposure rate of the cases to improve the ranking results. The proposed system that uses collaborative filtering has a growth precision that improves after each of the five rounds even though queries in natural language or keywords (see Figs. 9 and 11).

Furthermore, we observed that in the test data of keyword queries (Table 6), Q1–Q5 were considerably close keywords for searching tour books of Spain. When Q1 was input in our system, the performance was not satisfactory in the first round test (MAP/TOP-10 precision, EG was 0.62/0.56, but CG1 was 0.69/0.59 and CG2 was 0.95/0.89). When Q2, Q3, Q4 and Q5 were input in our system (EG and CG1) in the first round, a satisfactory precision was obtained, which was more satisfactory than Amazon (see Fig. 10). We discuss the satisfactory performance of Q1, Q2, Q4 and Q5 in the first round test, the result exceeded expectations. Our system learned new cases by collaborative filtering in the test of Q1 so that the learned cases contributed to other highly similar queries.

6. Conclusions

This paper presents a POS-based short-text semantic similarity (P-STSS) algorithm for natural language sentences, presents a case retrieval agent based on the P-STSS, and integrates the semantic-based retrieval agent into the proposed CBR system. The case retrieval method of existing CBR systems may not always determine perfect matching without obvious relationships or concept overlap between two natural language sentences because the string matching method cannot discover related words when the description of the desired items does not include the same inputted words. Some approaches address this problem via string similarity and keyword matching; however, they were hard to apply to natural language entities with ambiguous grammar structures and polysemous problems. The proposed retrieval method takes advantage of WordNet ontology and grammatical rules to overcome this problem.

The contributions of this work can be summarized as follows: (1) this study proposed the P-STSS algorithm which performed very well in both sentence similarity and paraphrase recognition; (2) the use of semantic and syntactic information that can determine the meanings of the natural language queries expressed user's intention more accurately; (3) to the best of our knowledge, the proposed system is the first approach that uses sentence-level semantic techniques to retrieve cases that helps CBR system have better recommendation quality at the beginning; (4) this study proposed a hybrid CBR system that involves a semantic-based case retrieval agent and a case-based machine learning mechanism, which outperforms the compared system; (5) our system allows the use of natural language queries for search. In particular, the system can help people who are unfamiliar with generating keywords. In addition, more specific or general questions more possibly can be answered from books.

Acknowledgements

This work was supported by the Ministry of Science and Technology, Taiwan, R.O.C., under contract no. NSC 102-2511-S-006-017-MY3 and MOST 103-2221-E-130-008.

References

[1] M. Essig, U. Arnold, Electronic procurement in supply chain management: an information economics-based analysis of electronic markets, J. Supply Chain Manag. 37 (3) (2001) 43–49.
[2] Y.E. Zeng, H.J. Wen, D.C. Yen, Customer relationship management (CRM) in business-to-business (B2B) e-commerce, Inf. Manag. Comput. Secur. 11 (1) (2003) 39–44.
[3] V.Y. Yoon, R.E. Hostler, Z. Guo, T. Guimaraes, Assessing the moderating effect of consumer product knowledge and online shopping experience on using recommendation agents for customer loyalty, Decis. Support Syst. 55 (4) (2013) 883–893.
[4] T.P. Liang, Y.F. Yang, D.N. Chen, Y.C. Ku, A semantic-expansion approach to personalized knowledge recommendation, Decis. Support Syst. 45 (3) (2008) 401–412.
[5] T. Li, T. Unger, Willing to pay for quality personalization? Trade-off between quality and privacy, Eur. J. Inf. Syst. 21 (6) (2012) 621–642.
[6] J. Bobadilla, F. Ortega, A. Hernando, J. Bernal, A collaborative filtering approach to mitigate the new user cold start problem, Knowl. Based Syst. 26 (2012) 225–238.
[7] M. Al-Hassan, H. Lu, J. Lu, A semantic enhanced hybrid recommendation approach: a case study of e-government tourism service recommendation system, Decis. Support Syst. 72 (2015) 97–109.
[8] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749.
[9] Y. Guo, Y. Peng, J. Hu, Research on high creative application of case-based reasoning system on engineering design, Comput. Ind. 64 (1) (2013) 90–103.
[10] C. Hayes, P. Cunningham, B. Smyth, A case-based reasoning view of automated collaborative filtering, Lect. Notes Comput. Sci. 2080 (2001) 234–248.
[11] R. Burke, A case-based reasoning approach to collaborative filtering, Lect. Notes Comput. Sci. 1898 (2000) 370–379.
[12] M. Sohn, J.H. Yim, S. Lee, H.J. Lee, Ontology-based dynamic and semantic similarity calculation method for case-based reasoning, Intell. Autom. Soft Comput. 20 (1) (2014) 33–46.
[13] S. Montani, How to use contextual knowledge in medical case-based reasoning systems: a survey on very recent trends, Artif. Intell. Med. 51 (2) (2011) 125–131.
[14] M.M. Richter, Similarity, in: Proceedings of Case-Based Reasoning on Images and Signals, 2008, pp. 25–90.
[15] M. Crochemore, W. Rytter, Text Algorithms, Oxford University Press, New York, 1994.
[16] N. Singla, D. Garg, String matching algorithms and their applicability in various applications, Int. J. Soft Comput. Eng. 1 (6) (2012) 218–222.
[17] S.C. Sahinalp, M. Tasan, J. Macker, Z.M. Ozsoyoglu, Distance based indexing for string proximity search, in: Proceedings of 19th IEEE International Conference on Data Engineering, 2003, pp. 125–136.
[18] J.M. Alvarez-Rodríguez, J.E. Labra-Gayo, P.O. de Pablos, New trends on e-procurement applying semantic technologies: current status and future challenges, Comput. Ind. 65 (5) (2014) 800–820.
[19] T. Madhusudan, J.L. Zhao, B. Marshall, A case-based reasoning framework for workflow model management, Data Knowl. Eng. 50 (2004) 87–115.
[20] S. Kapetanakis, M. Petridis, Evaluating a case-based reasoning architecture for the intelligent monitoring of business workflows, in: Proceedings of Successful Case-based Reasoning Applications-2, 2014, pp. 43–54.
[21] S. Montani, G. Leonardi, Retrieval and clustering for business process monitoring: results and improvements, in: Proceedings of Case-Based Reasoning Research and Development, 2012, pp. 269–283.
[22] R. Bergmann, Y. Gil, Similarity assessment and efficient retrieval of semantic workflows, Inf. Syst. 40 (2014) 115–127.
[23] M. Minor, R. Bergmann, S. Gorg, K. Walter, Reasoning on business processes to support change reuse, in: Proceedings of 13th IEEE International Conference on Commerce and Enterprise Computing, 2011, pp. 18–25.
[24] N. Althuizen, B. Wierenga, Supporting creative problem solving with a case-based reasoning system, J. Manag. Inf. Syst. 31 (1) (2014) 309–340.
[25] S. Chattopadhyay, S. Banerjee, F.A. Rabhi, U.R. Acharya, A case-based reasoning system for complex medical diagnosis, Expert Syst. 30 (1) (2013) 12–20.
[26] S. Guessoum, M.T. Laskri, J. Lieber, RespiDiag: a case-based reasoning system for the diagnosis of chronic obstructive pulmonary disease, Expert Syst. Appl. 41 (2) (2014) 267–273.
[27] S. Begum, M.U. Ahmed, P. Funk, N. Xiong, M. Folke, Case-based reasoning systems in the health sciences: a survey of recent trends and developments, IEEE Trans. Syst. Man Cybern. 41 (4) (2011) 421–434.
[28] N. Mishra, S. Petrovic, S. Sundar, A self-adaptive case-based reasoning system for dose planning in prostate cancer radiotherapy, Med. Phys. 38 (12) (2011) 6528–6538.
[29] A. Aamodt, E. Plaza, Case-based reasoning: foundational issues, methodological variations, and system approaches, AI Commun. 7 (1) (1994) 39–59.
[30] J.L. Kolodner, An introduction to case-based reasoning, Artif. Intell. Rev. 6 (1) (1992) 3–34.
[31] C.K. Riesbeck, R.C. Schank, Inside Case-Based Reasoning, Psychology Press, 2013.
[32] A. El-Fakdi, F. Gamero, J. Meléndez, V. Auffret, P. Haigron, eXiTCDSS: a framework for a workflow-based CBR for interventional clinical decision support systems and its application to TAVI, Expert Syst. Appl. 41 (2) (2014) 284–294.
[33] E. Vargiu, A. Giuliani, G. Armano, Improving contextual advertising by adopting collaborative filtering, ACM Trans. Web 7 (3) (2013) 13.
[34] X. Luo, Y. Tu, J. Tang, C.K. Kwong, Optimizing customer's selection for configurable product in B2C e-commerce application, Comput. Ind. 59 (8) (2008) 767–776.
[35] G. Linden, B. Smith, J. York, Amazon.com recommendations: item-to-item collaborative filtering, IEEE Trans. Internet Comput. 7 (1) (2003) 76–80.
[36] R. Navigli, Word sense disambiguation: a survey, ACM Comput. Surv. 41 (2) (2009) 1–69.
[37] J. Oliva, J.I. Serrano, M.D. del Castillo, Á. Iglesias, SyMSS: a syntax-based measure for short-text semantic similarity, Data Knowl. Eng. 70 (4) (2011) 390–405.
[38] R. Rada, H. Mili, E. Bicknell, M. Blettner, Development and application of a metric on semantic nets, IEEE Trans. Syst. Man Cybern. 19 (1) (1989) 17–30.
[39] Z. Wu, M. Palmer, Verbs semantics and lexical selection, in: 32nd Annual Meeting of the Associations for Computational Lingustics, New Mexico, 1994.
[40] C. Leacock, M. Chodorow, Combining local context and WordNet similarity for word sense identification, in: WordNet: An Electronic Lexical Database, 1998, 265–283.
[41] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence, 1995, pp. 448–453.
[42] J. Jiang, D. Conrath, Semantic similarity based on corpus statistics and lexical taxonomy, in: Proceedings on International Conference on Research in Computational Linguistics, 1997, pp. 19–33.
[43] D. Lin, An information-theoretic definition of similarity, in: Proceedings of the International Conference on Machine Learning, 1998, pp. 296–304.
[44] M. Lesk, Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone, in: Proceedings of the 5th Annual International Conference on Systems Documentation, 1986, pp. 24–26.
[45] S. Banerjee, T. Pedersen, Extended gloss overlaps as a measure of semantic relatedness, in: Proceedings of the 18th International Joint Conference on Artificial Intelligence, 2003, pp. 805–810.
[46] L. Han, T. Finin, P. McNamee, A. Joshi, Y. Yesha, Improving word similarity by augmenting PMI with estimates of word polysemy, IEEE Trans. Knowl. Data Eng. 25 (6) (2013) 1307–1322.
[47] Y. Xu, X. Guo, J. Hao, J. Ma, R.Y. Lau, W. Xu, Combining social network and semantic concept analysis for personalized academic researcher recommendation, Decis. Support Syst. 54 (1) (2012) 564–573.
[48] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, M. Stede, Lexicon-based methods for sentiment analysis, Comput. Linguist. 37 (2) (2011) 267–307.
[49] A. Islam, D. Inkpen, Semantic text similarity using corpus-based word similarity and string similarity, ACM Trans. Knowl. Discov. Data 2 (2) (2008) 1–25.
[50] G. Tsatsaronis, I. Varlamis, M. Vazirgiannis, Text relatedness based on a word thesaurus, J. Artif. Intell. Res. 37 (1) (2010) 1–39.
[51] A. Fernández, Y. Gutiérrez, A. Chávez, H. Dávila, A. González, R. Estrada, Y. Castañeda, S. Vázquez, A. Montoyo, R. Muñoz, UMCC-DLSI: multidimensional lexical-semantic textual similarity, in: Proceedings of the First Joint Conference on Lexical and Computational Semantics, 2012, pp. 608–616.
[52] Y. Li, D. McLean, Z.A. Bandar, J.D. O'Shea, K. Crockett, Sentence similarity based on semantic nets and corpus statistics, IEEE Trans. Knowl. Data Eng. 18 (8) (2006) 1138–1150.
[53] M.C. De Marneffe, B. MacCartney, C.D. Manning, Generating typed dependency parses from phrase structure parses, Proceedings of LREC 6 (2006) (2006) 449–454.
[54] M.P. Marcus, M.A. Marcinkiewicz, B. Santorini, Building a large annotated corpus of English: the Penn Treebank, Comput. Linguist. 19 (2) (1993) 313–330.
[55] T. Pedersen, S. Patwardhan, J. Michelizzi, WordNet::similarity: measuring the relatedness of concepts, in: Demonstration Papers at HLT-NAACL 2004, 2004, 38–41.
[56] C. Quirk, C. Brockett, W. Dolan, Monolingual machine translation for paraphrase generation, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2004, pp. 142–149.
[57] J. O'Shea, Z. Bandar, K. Crockett, D. McLean, A comparative study of two short text semantic similarity measures, Lect. Notes Artif. Intell. 4953 (2008) 172–181.
[58] P. Turney, Mining the web for synonyms: PMI-IR versus LSA on TOEFL, in: Proceedings of the Twelfth European Conference on Machine Learning, 2001, pp. 491–502.
[59] R. Mihalcea, C. Corley, C. Strapparava, Corpus-based and knowledge-based measures of text semantic similarity, in: Proceedings of the Twenty First Conference on Artificial Intelligence, 2006, pp. 775–780.
[60] M.C. Lee, J.W. Chang, T.C. Hsieh, A grammar-based semantic similarity algorithm for natural language sentences, Sci. World J. 2014 (437162) (2014).
[61] S. Wan, M. Dras, R. Dale, C. Paris, Using dependency-based features to take the parafarce out of paraphrase, in: Proceedings of the Australasian Language Technology Workshop, 2006, pp. 131–138.
[62] Y. Zhang, J. Patrick, Paraphrase identification by text canonicalization, in: Proceedings of the Australasian Language Technology Workshop, 2005, pp. 160–166.

[63] L. Qiu, M. Kan, T. Chua, Paraphrase recognition via dissimilarity significance classification, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2006, pp. 18–26.
[64] V. Vapnik, The Nature of Statistical Learning Theory, Springer, 1995.
[65] E. Agichtein, E. Brill, S. Dumais, Improving web search ranking by incorporating user behavior information, in: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2006, pp. 19–26.
[66] G.V. Cormack, T.R. Lynam, Statistical precision of information retrieval evaluation, in: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2006, pp. 533–540.
[67] H. Zaragoza, H. Rode, P. Mika, J. Atserias, M. Ciaramita, G. Attardi, Ranking very many typed entities on Wikipedia, in: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, 2007, pp. 1015–1018.
[68] J. Weston, S. Bengio, S.N. Usunier, Large scale image annotation: learning to rank with joint word-image embeddings, Mach. Learn. 81 (1) (2010) 21–35.

Jia Wei Chang is a Ph.D. candidate in the department of engineering science at the National Cheng Kung University. He received his B.S. degree in computer and communication engineering from Ming Chuan University in 2010. In 2012, he graduated with a M.S. in engineering science from National Cheng Kung University. Currently, he is working toward his doctoral degree in engineering science. His research interests include natural language processing, artificial intelligent, data mining, and e-learning technologies.

Ming Che Lee received the B.S. degree in computer science from National Taiwan Ocean University and M.S. and Ph.D. degree from National Cheng Kung University, Taiwan, in 2002, 2003 and 2008, respectively. Since 2008, he has been a professor at computer and communication engineering of Ming Chuan University (Taoyuan, Taiwan). His research interests are in the fields of semantic web, ontology, artificial intelligence, algorithms, digital content, natural language processing, and e-learning technologies.

Tzone I Wang is a professor of engineering science at National Cheng Kung University, Taiwan. He held B.S. of engineering science at National Cheng Kung University and Ph. D. degree at Imperial College, University of London, UK. His major research areas include artificial intelligence, web and network services, distributed database, mobile agent based distributed systems and network applications, and e-learning technologies.