Available online at www.ijpe-online.com

**International Journal of Performability Engineering**

vol. 14, no. 6, June 2018, pp. 1099-1108
DOI: 10.23940/ijpe.18.06.p1.10991108

# Document Correlation Measurement based on Conceptual Dependent Forest

Gang Liu, Hanwen Zhang*, and Hanmo Zhang

*School of Computer Science and Technology, Harbin Engineering University, Harbin, 150001, China*

**Abstract**

The formal expression of natural language is the primary task of all natural language problems. In this paper, we propose the concept of conceptual dependency tree based on the conceptual dependency theory. Conceptual dependency differs from dependency analysis in that the performance at the grammatical and semantic level is more concerned with the conceptual hierarchy. Based on the conceptual dependency tree, a conceptual dependency forest model is defined, which provides a solution to the formalized representation of natural language. Based on the conceptual dependency forest model, the definition and calculation method of conceptual dependency strength and potential similarity are further proposed. The experiment proves that the conceptual dependency forest model proposed in this paper is reasonable and effective.

*Keywords*: conceptual dependency theory; conceptual dependency intensity; potential similarity; requirement dependency

(Submitted on March 1, 2018; Revised on April 21, 2018; Accepted on May 15, 2018)



## 1. Introduction

As an informal sentence, natural language sentences must be formalized in the production process. At the same time, in the process of formalization of natural language sentences, the relations among the morphemes of each sentence have become the major difficulties in formalization. At present, the most common method is to use morpheme processing in the formalized process of guiding the formalization of dependency theory. However, as a widely accepted guideline theory, the theory of dependency is far from being widely used to solve practical problems. Therefore, a new formal method is urgently needed.

## 2. Related studies

### *2.1. Research Status*

#### *2.1.1. Conceptual dependency Theory*

Conceptual dependency theory is also called the conceptual compliance theory [5]. The theory has two basic points: first, Schank believes that when people understand natural language, they rely on potential conceptual expressions rather than specific words or sentences. The second is to establish the axiom of conceptual dependency theory. Two sentences with the same meaning have the same internal representation and the same deep structure regardless of whether their words are the same or the order of words is the same. Conceptual subordination theory hopes to describe common sense in a systematic and specific way and use those basic actions to reason quickly so as to achieve an automatic understanding of the language.

#### *2.1.2. The proposition of Conceptual dependency Theory*

Conceptual dependency theory proposed by Schank is "conceptual hierarchy" and does not rely on the source language. But, it is obvious that the realization of this theory is difficult. So, it has not been more concretely applied. Article [7] proposed to use the dependency relation to abstract the specific knowledge domain and to organize and manage the knowledge in the specific domain with the conceptual dependency tree. It provides important theoretical support for the concrete application of the conceptual dependency theory in the representation of restricted Chinese sentences.

* Corresponding author.
*E-mail address*: 923074315@qq.com

In the conceptual dependency tree, each node in the tree represents a concept. Tree root is the most abstract concept, the leaf is the most specific concept, and the node between the root and the leaf are different levels of abstraction. They form a tree through the arc between nodes. Using the hierarchical relationship of the tree itself to represent the relationship between concepts, child nodes can inherit all the attributes of the parent node. In the concept of dependency tree, the more there are of the lower branches, the more detailed and more specific it is; on the other hand, the more there are of the upper floors, the broader and more abstract it is.

### *2.2. Existing problems*

#### *2.2.1. Theory Tree-base*

Since conceptual dependency trees are not labeled by sentence clauses, conceptual dependency trees do not constitute a true "tree bank". The tree-base is actually a "bank" of conceptual dependencies that resembles a database table, with each conceptual dependency tree representing a record in the table. Each conceptual dependency tree corresponds to a complete concept [2]. This structure does not require large thesaurus support, but the disorderly nature of the database tables makes it impossible to characterize the relationships between the different records in the table.

#### *2.2.2. Complete Conceptual Dependency Tree*

A major feature of the conceptual dependency tree is that the attributes of the parent node need not be labeled at the child node because the child node inherits all the attributes of the parent node. However, the payment base tree and the payment ratio tree all seem to belong to the payment tree. Many attributes of the concept of "payment", such as payment time, payment type, payment unit and so on are the attributes that "payment base" and "payment ratio" do not have. "payment" will not be the father of "payment base" and "payment ratio". In other words, if we want to construct a complete conceptual dependency tree, we must discard the inheritance relations between many attributes at the same time, which is a price we cannot pay.

## 3. Conceptual dependency Forest Model

### *3.1. Definition Description*

#### *3.1.1. Conceptual dependency tree structure*

There is an essential difference between requirement dependency parsing trees and conceptual dependency trees:

- Requirement dependency parsing trees are constructed at the grammar and semantic level, while the conceptual dependency tree is constructed at the conceptual level. The tree structure of the requirement dependency parsing tree is constructed on the grammar level, and the dependency relationship is constructed on the semantic level. The conceptual hierarchy of conceptual dependency tree construction is a new level of different grammatical semantics. The conceptual hierarchy of conceptual dependency tree construction is a new level of different grammatical semantics [3].

- The requirement dependency parsing tree consists of four types of nodes, which are derived from four different parts of speech. The conceptual dependency tree consists of three types of nodes, which only come from the part of speech [2].

- The requirement dependency parsing tree characterizes the dependencies among keywords in the clause, whereas the conceptual dependency tree characterizes the subordination between the concepts in different clauses.

The following first gives the definition of requirement dependency parsing tree and conceptual dependency tree.

Definition (requirement dependency parsing tree): Set RK as a set of requirement keywords. The dependency parsing tree of RK is a set with two-tuple. DAtree=<D, R>, there is:

- DAtree is a kind tree.
- D is a collection of data elements with the same characteristics. $D \subseteq \{V,N,P,A\}$. There into V is the set of v elements, N is the set of n elements, P is the set of parameter elements, and A is the set of attribute element. $V \subset RK.N \subset RK.P \subset RK.A \subset RK$;

- If |D|=0, then R=Φ and DAtree is an empty depending parsing tree;
- If |D|=1, then R=Φ,D=N,P=Φ and DAtree is a single node independency parsing tree;
- If |D|≥2, then R is the set of certain binary relation H in D, namely R={H}. H obeys the grammar rules of context-free and the restrain of $V \to V \mid N \mid p$ and $N \to a$ . There into, $p \in P$ , $a \in A$ ,V and N are nonterminal symbols. P and a are terminators [1].

Definition (conceptual dependency tree): Set RK as a set of requirement keywords. The conceptual dependency tree of RK is a set with two-tuple. CDtree=<D, R>, there is:

- CDtree is a kind tree.
- D is a collection of data elements with the same characteristics. $D \subseteq \{N, C, A\}$， N is the set of concept elements, C is the set of classification elements, A is the set of attribute elements, $N \subset RK$， $A \subset RK$；
- If $|D|=0$， then $R=\Phi$， and CDtree is an empty depending parsing tree;
- If $|D|=1$， then $R=\Phi$， and $D=N, C=\Phi$， and CDtree is a single node independency parsing tree;
- If $|D| \geq 2$， then R is the set of certain binary relation H in D，namely $R=\{H\}$， H obeys the grammar rules of context-free and the restrain of $N \to C \mid a$ and $C \to N \mid n$ .There into $n \in N$， $a \in A$， N and C are nonterminal symbols，and n and a are terminators.

*3.1.2. Definition of Conceptual dependency forest*

With the definition of a conceptual dependency tree, it is convenient to propose the conceptual dependency forest.

Definition (conceptual dependency forest) Set CDtree to be a sentence conceptual dependency tree collection. The notion of a dependent tree structure on CDtree is a two-tuple CDforest = <D, R> there is:

- CDforest is a kind of forest，is a collection of $m(m \geq 0)$ conceptual dependency trees that are disjointed;
- D is a collection of data elements with the same characteristics. $D \subseteq CDtree$ ；
- If $|D|=0$， then $R=\Phi$， and CDforest is an empty conceptual dependency forest;
- If $|D|=1$， then $R=\Phi$， and $CDforest = CDtree$， CDforest is a single conceptual dependency tree；
- If $|D| \geq 2$， then R is a set of n-element relations H in D，and $R=\{H\}$. The n-element relationship H is a functional relationship on CDtree $H(w_1, w_2, \ldots, w_n)$ . Therefore, $w_i \in G$ ， $i=1,\ldots,n$ ， and the function $H(w_1, w_2, \ldots, w_n): CDtree \times G \to CDtree$

If it meets the above definition, the forest is a conceptual dependency forest. Different from the definition of a forest in the traditional sense, the trees in the conceptual dependency forest are inextricably linked and at the same time are bound by the forest operation. The unity of the growth and decay of trees in the forest is governed by the "law of nature" of forest operations. The conceptual dependency of forests is in accordance with the law of nature [4].

*3.2. Construction and Updating Algorithm of conceptual dependency Forests*

The following is a formal description of the algorithm.

**Algorithm 1** Construction and Updating Algorithm of Conceptual Dependency Forests

**Input:** The concept root of the first concept tree in the dependency forest root，Into the forest node new node
**Output:** New conceptual dependency Forest First Concept Tree Root Node new root

1. New node is a complete concept description that extends into the forest node，comstr is the concept of common ground between the forest root and the node into the forest new node;
2. **If** comstr = root = new node, then it indicates the existence of the concept. The root value is given a new root, and if comstr = root, **then it** indicates the existence of the concept of the new concept of point. Repeat in the tree operation;
3. **If** the conceptual dependency node of root is not empty, the root value will be given to comstr. **If** comstr = new node, **then the** marked root is the new branch node;
4. Create a new classification node under the root and set the new node as the child node of the new root classification node. Create a new classification node and set the root of the original dependency node as the new classification node. Extract

new node sub-node attributes to the new node and extract new node attributes into the root. Otherwise, assign the root conceptual dependency node to the root;
5. Recursive makecd forest(root, new node). Otherwise, create a new classification node (new leaf node) under the new root conceptual dependency node. Set the new node as the dependency node of the new conceptual dependency node, and extract the new node attributes to the root conceptual dependency node. If comstr = new node, then it indicates the new root node.
6. Create a new node under the new node and set root as the node of new node. Extract the attributes of the root node to the new node, and give the root node the same root concept of the root;
7. Recursive makecd forest(root, new node). If comstr is not equal to the root and comstr is not equal to the new node, it indicates that there is no common ground. Give the same conceptual dependency the root node to the root. If the root is not empty, then recursive makecd forest (root, new node). Otherwise, the new node value is assigned to the new root. (Judged as a new concept, build a new tree);
8. **end**

### *3.3. Forest operation function*

#### *3.3.1. Induction of forest operations*

In the forest building and updating algorithm in Section 3.2 of this paper, many forest operations have been proposed, including: makecdforest (m, n) node n into forest m; expand (m) extended node m is a complete concept description; comstring (m, n) computes the commonalities between node m and node n; new classname (m) new classification node m; insertin (m, n) insert node n as node m child node; attrpickup (m, n) m) extract the attributes of node m to the parent node; new root (m) to m as the root node. These are all public operations of the forest, but public operations are not enough to describe the interplay of trees in a complex forest.

A similar operation is performed on all sentence clauses, and all concept operation functions in the conceptual dependency forest are extracted. The concepts of operational functions and forest public operations collection are a complete set of forest operations.

#### *3.3.2. Storage structure of forest operations*

Observe the direct successor to the root node and the root node of each verb subtree in the requirement dependency parsing tree. The root node of a verb subtree is equivalent to a concept operation function, and the direct successor of a root node happens to be a parameter of a concept operation function. By traversing all verb subtrees of the requirement dependency parsing tree, you get all of the concept manipulating functions [1,6].

#### *3.3.3. Forest operation extraction algorithm*

Based on the results of the requirement dependency parsing tree and the stored results of the forest manipulation functions presented above, the algorithm for extracting forest operations can be constructed as follows:

**Algorithm 2** Extracting Algorithm of conceptual dependency Forest Operation Function

**Input:** The root of the first requirement dependency parsing tree root
**Output:** Extracted forest operations new node

1. **If** the root is not empty or the tree is a noun tree, **then the** new node will take a null value. Otherwise, extract the root verb norm word and give the new node a new function name;
2. **When** the root of the concept of the conceptual dependency of the non-empty cycle extracts root conceptual dependency of normative words, give the new node a new concept parameter name. Root conceptual dependency belongs to the same node assignment root. (Concept parameter extraction);
3. **When** the root dependency verb list non-empty cycle extracts root dependent verbs norms, give the new node a new function parameter name. Root dependency verbs belong to the same root node assignment. (Function parameter extraction);
4. **When the** root dependent parameter list is not empty, cycle to extract the root dependent parameter specification word and give the new node a new name of the other parameters. Root subordinate parameters belong to the same node assignment root. (Other parameter extraction);
5. Root same subtree verb node assignment root. If the root is not empty, then recursively funcpickup (root);
6. Croot of the same forest verb node assignment root. If the root is not empty, then recursively funcpickup (root);
7. **end**

At this point, the conceptual dependency forest structure is completed. In essence, the concept dependent afforestation in the conceptual dependency forest is a tree formed by connecting the classification nodes after the nominal subtree of the requirement dependency parsing tree is normalized. Forest operations function in forests are two-level verb subtree collections of requirement dependency parsing trees.

## 4. The application of conceptual dependency forest

### *4.1. Conceptual dependency measurement*

#### *4.1.1. Conceptual dependency strength calculation*

The degrees of subordination between different concepts and their parent concepts are different. Even the different sub-concepts of the same concept have different degrees of subordination to the concept. In order to describe the degree of subordinate degree between different concept nodes, the conceptual dependency strength is introduced.

Definition（conceptual dependency strength）Set $T \subseteq CDforest$ , $m \subset CDtree$ , $n \subset CDtree$ , and $m \in T$ , $n \in T$ , $ancestor(m,n) = true$ or $descendant(m.n) = true$ , the conceptual dependency strength between m and n Depint(m,n) is：

$$Depint(m,\ n) = 1 + \log \frac{2 \times length(comstr(expand(m), expand(n)))}{length(expand(m)) + length(expand(n))} \tag{1}$$

Where m and n are nodes in the same conceptual dependency tree, ancestor(m,n) calculates whether m is an ancestral node of n. descendant(m,n)calculates whether m is a descendant node of n. expand(m) expands on concept m and it expands m to a complete description of the standard concept. comstr(m,n) calculates common concepts for concepts m and n， and length(m) calculates the length of the description string m.

#### *4.1.2. Inside semantic similarity calculation*

In the calculation of dependency strength, the first emphasis is on the subordination between concepts. The two conceptual dependency nodes are indifferent of the "subordinate" strength of Equation (1). But, at the same time it can notice:

- The same concept affiliate tree brother node, cousin node, and nephew node do not subordinate. But, from the conceptual level they are very similar, after all, are generated by the same ancestor concept classification.
- The closer the kinship node, the greater the degree of similarity. The similarities between brother nodes are generally greater than the similarities between distant nephew nodes.

Due to the concept of independency, the definition of conceptual dependency strength is not suitable to describe the similarities between them. The conceptual semantic similarity introduced here is used to describe the similarity between concepts. The concept of semantic similarity, which describes the semantic similarity between key words and norm words, is called the inside semantic similarity.

Definition（inside semantic similarity): Set $T \subseteq CDforest$ , $m \subset CDtree$ ， $n \subset CDtree$ ， and $m \in T$ ， $n \in T$ ， the inside semantic similarity between m and n Inseside(m,n)is：

$$Insesidea(m,\ n) = 1 + \log \frac{2 \times length(comstr(expand(m), expand(n)))}{length(expand(m)) + length(expand(n))} \tag{2}$$

$$Insesideb(m,\ n) = \frac{depth(mincoan(m,n))}{\max(depth(m), depth(n))} \tag{3}$$

$$Inseside(m,\ n) = (Insesidea(m,n) + Insesideb(m.n)) \Big/ 2 \tag{4}$$

*m* and *n* are nodes in the same conceptual dependency tree，expand(m) expands concept m to include a complete description of the standard concepts，comstr(*m*, *n*) calculates common conceptual points for concepts m and n，length(m) calculates the description string length of concept m，mincoan(m,n) calculates the closest common ancestry of concepts m and n，and depth(m) calculates the depth of m in the conceptual dependency tree.

Definition (outside semantic similarity): Set domain specification vocabulary $T$， $n \in T$ ，define the degree of semantic similarity between m and n ouseside(m, n)，and $ouseside(m,n) \in [0,1]$ .

### *4.2. Potential similarity*

After constructing the conceptual dependency forest model, each concept exists as a conceptual dependency tree node. Each sentence keyword corresponds to the concept of belonging to the conceptual dependency tree, that is, the conceptual structure of the key words. Extract the corresponding conceptual dependency tree of each keyword in the sentence，and the resulting conceptual dependency tree sets should be sub-forests of the entire conceptual dependency forest.

Definition (Conceptual dependency sub forest): For the conceptual dependency forest CDforest = <D，R>，there are conceptual dependency trees $T_1, T_2, \ldots T_n$ . Make $D = \{T_1, T_2, \ldots T_n\}$ ，then for $T_i \in D$ ， $i \in [1, m]$ ， $m \le n$ ，there are $D' = \{T_1, T_2, \ldots T_m\}$ ， $D' \subseteq D$ ，if in $D'$ ，there are forest operation subsets $R' \subseteq R$ . Then, the binary group $< D', R' >$ is the conceptual dependency sub forest of CDforest.

Especially，if $m < n$ ，and $D' \subset D$ ， $R' \subset R$ ，then $< D', R' >$ is the conceptual dependency proper sub forest of CDforest. Forest operation subset $R'$ of conceptual dependency proper sub forest $< D', R' >$ is not complete on $D'$ ，but it is still valid.

Definition (Potential similarity): if there are conceptual dependency sub forest $CDforest_1 =< D_1, R_1 >$ and $CDforest_2 =< D_2, R_2 >$ ，where $D_1 \subseteq D$ ， $D_2 \subseteq D$ ，if $D_1 \bigcap D_2 \neq \Phi$ ，then conceptual dependency sub forests $CDforest_1$ and $CDforest_2$ are potentially similar.

If $D_1 \bigcap D_2 = \Phi$ ，then call conceptual dependency sub forests $CDforest_1$ and $CDforest_2$ dissimilar.

Because both D1 and D2 are conceptual dependency tree sets， $D_1 \bigcap D_2 = \Phi$ is equal to two sentence clauses without any conceptual level of similar keywords. According to definition 4.1，the similarity between two sentences is 0，which is not similar.

If $D_1 \bigcap D_2 \neq \Phi$ ，there is at least one concept dependent tree. The concept of the tree appears in the two clauses simultaneously. We can establish the similarity calculation by constructing the keyword correspondence. According to definition 4.4，the similarity value must be greater than zero. Whether the two clauses are similar or not depends on whether the finally calculated similarity value is greater than the specified threshold.

If $D_1 \bigcap D_2 \neq \Phi$ ，for every conceptual dependency tree in $D_1 \bigcap D_2$ , there is a key word corresponding to the establishment. If the corresponding keyword is the conceptual dependency tree, outside semantic similarity is used to indicate similar acceptance；Conversely, if the corresponding keywords are different nodes of the conceptual dependency tree，inside semantic similarity is used to indicate similar acceptance.

When two or more sentence keywords in two clauses correspond to the same keyword in another clause, the following principles should be observed:

- Specification of the word is prioritized. When a plurality of words have the same correspondence through the canonical words and the subordinates corresponding through the canonical words correspond, the corresponding relationships established by the canonical words are preferentially matched.

- Single corresponding is prioritized. However, there are many words, and some words correspond to multiple words of the opposite clause at the same time. Some words only correspond to one word in the opposite clause, and the keywords corresponding to unique words are preferentially matched.

- When the first two rules cannot be distinguished，calculate the value of different components Pi,Oi,Oj，and preferentially match the word pairs that form the maximum to ensure the similarity calculation results are maximized （where Oi,Oj is the outside semantic similarity between two sentences，P is the inside semantic similarity Equation (2,3,4) between the two sentences）

### *4.3. Sentence dependency*

There are a large number of elements 1 in the judgment matrix and the auxiliary judgment matrix constructed when calculating the criticality of sentences. Unless the two keywords are completely semantically equivalent, the importance of two different keywords in the sentence cannot be exactly equal.

The source of the above problem lies in the sentence independence assumption. It is assumed that the sentence clauses are linearly independent and there is no relation between the keywords of the sentences. In fact, as part of the text of a sentence, the semantic dependency between the key words of a statement describing the same or similar statements must exist.

#### *4.3.1. Dependency classification of sentences*

First, discuss the different types of sentence dependency:

The location of relevant words can be divided into Self-dependency and Co-dependency.
Because of the key words and similarities of sentences studied in this paper, they are all calculated between different sentence clauses. Dependency (Self-dependency) of key words in sentence clauses has no practical significance for the content of this paper. At the same time, it is easy to find that Co-dependency based on equivalence words can be converted to Self-dependency under the premise of equivalent Co-dependency. Therefore, the main problem of the research should be Co-dependency of equivalent and non-equivalent keywords, excluding the Co-dependency between Self-dependency and equivalence and non-equivalence keywords.

The relationship between related words is divided into direct and indirect dependency. The aforementioned dependencies are all directly related. Indirect dependency refers to the surface of the irrelevant keywords through the dependency of the transmission of the relationship.

The indirect dependency caused by such dependency has the following problems:

- Specification of the word is prioritized. When a plurality of words have the same correspondence through canonical words and the subordinates corresponding through the canonical words correspond, the corresponding relationships established by the canonical words are preferentially matched.

- Indirect dependency decreased rapidly with an increase of delivery times. If the first indirect dependency is 0.4, the second indirect dependency is 0.16, and the third indirect dependency is 0.064. When the dependency between the two keywords is 0.064, it is equivalent to saying that the independency between the two keywords is 0.936, which is almost equivalent to saying that the two terms are irrelevant.

To sum up the above analysis of the indirect correlation, we can get two conclusions:

- The calculation of indirect dependency abruptly weakened with the extension of the correlation chain, which has no practical significance or significance for the study of correlation degree calculation.

- In order to avoid the result similar to Dd (w4, w4) = 0.064, it is not advisable to measure the relevancy between keywords, and the measure of relevancy is only carried out between sentence clauses. So, for the relevant and irrelevant concepts, there is no concept of dependency.

#### *4.3.2. Dependency Definition under Conceptual Dependency Forest Model*

The conceptual dependency forest model is actually a dependency model. The dependency of the sentences to be discussed within the concept of the subordinate forest model is of practical significance. The various forest operations in the conceptual

depencency forest are in fact all related. According to Section 3.3 of this paper, the definition of the conceptual dependency forest is a two-tuple CDforest = <D, R>, where D is a conceptual dependency tree set, and R is a forest operation function set. Here's a set of sentence-dependency definitions based on conceptual dependency forest models:

Definition (dependency relation): Set conceptual dependency forest CDforest，Forest conceptual dependency tree set D ，forest operation function set R. If $c_1, c_2, \ldots, c_m \in D$ ， $f, f_1, f_2, \ldots, f_n \subseteq R$ ， $p_1, p_2, \ldots, p_r$ are the parameter set of R，make $f(c_1, c_2, \ldots, c_m, f_1, f_2, \ldots, f_n, p_1, p_2, \ldots, p_r) \subseteq R$ ， m,n,r are positive integer，then f is a dependency relation on CDforest.

Definition (direct dependency): Set conceptual dependency tree set D，forest operation function set R. If $f(c_1, c_2 \ldots) \subseteq R$ exists，where $c_1, c_2 \in D$ ， then $c_1, c_2$ is of direct dependency and the dependency relation between $c_1, c_2$ is of direct dependency relation.

Definition (Self-dependency): Set conceptual dependency tree set D，Statement clause S. If $c_1, c_2 \in D$ ， make $c_1, c_2$ direct dependency and $c_1, c_2 \in S$ ， then the dependency relation between $c_1, c_2$ is direct self-dependency. Call $c_1, c_2$ as Self-dependency.

Definition (Co-dependency): Set conceptual dependency tree set D，statement clause S1 and S2. If $c_1, c_2 \in D$ exists，make $c_1, c_2$ of direct dependency and if $c_1 \in S_1$ ， $c_2 \in S_2$ ， and $S_1 \bigcap S_2 = \Phi$ ， then the dependency relation between $c_1, c_2$ is of direct Co-dependency. Call $c_1, c_2$ as Co-dependency.

Definition (indirect dependency): Set conceptual dependency tree set D， forest operation function set R. If $f(c_1 \ldots, f_1(c_2 \ldots) \ldots) \subseteq R$ exists ， where $c_1, c_2 \in D$ ， then call $c_1, c_2$ as indirect dependency. The dependency relation between $c_1, c_2$ is of an indirect dependency relation.

Definition (weak dependency): Set conceptual dependency forest CDforest，forest conceptual dependency tree set D，forest operation function set R. If $c_1, c_2, \ldots, c_m \in D$ ， $f, f_1, f_2, \ldots, f_n \subseteq R$ ， $p_1, p_2, \ldots, p_r$ are the parameter sets of R，make $f(c_1, c_2, \ldots, c_m, f_1, f_2, \ldots, f_n, p_1, p_2, \ldots, p_r) \subseteq R$ ， and $f(c_1, c_2, \ldots, c_m, f_1, f_2, \ldots, f_n, p_1, p_2, \ldots, p_r) = null$ ， m,n,r are positive integers. Then, call f as a weak dependency of CDforest.

Definition (strong dependency): Set conceptual dependency forest CDforest，forest conceptual dependency tree set D ，forest operation function set R. If $c, c_1, c_2, \ldots, c_m \in D$ ， $f, f_1, f_2, \ldots, f_n \subseteq R$ ， $p_1, p_2, \ldots, p_r$ are the parameter sets of R，make $f(c_1, c_2, \ldots, c_m, f_1, f_2, \ldots, f_n, p_1, p_2, \ldots, p_r) \subseteq R$ ， and $f(c_1, c_2, \ldots, c_m, f_1, f_2, \ldots, f_n, p_1, p_2, \ldots, p_r) = c$ ， m,n,r are positive integers and $c \neq \phi$ . Then, call f as a strong dependency of CDforest.

*4.3.3. Sentence dependency calculation*

In the sentence clause, the property value is set to calculate the value itself. In these operations, such as the payment amount = payment base × payment ratio, the relationship between each operation can be expressed as the following forms:

$$c = f(c_1, c_2, \ldots, c_m, f_1, f_2, \ldots, f_n, p_1, p_2, \ldots, p_r)$$

The above meets the definition of a strong dependency. The keyword c in the sentence clause set depends on other m keywords in the sentence clause set. In a sentence operation equation, there is a dependency between the keywords on the left and right sides of the equation. This dependence is also of the strong dependency.

Do not measure the dependency between keywords. Dependency measures only the sentence clause. In this way, whether it is strong or weak, it shows that there is a dependency between two or more sentence keywords. The thing that affects the degree of dependency between keywords is only the acceptance of the key words, that is, the outside semantic similarity between keywords.

Therefore, we only need to extract these forest operation functions from the conceptual dependency forest operations. According to the appearance of keywords in function parameters, we can determine the existence of the dependency between keywords.

The following discusses the dependency of the correlation between statement clauses. Suppose the number of relevant keywords between sentences determines the dependency of Ddn and the dependency of each pair of related keywords determines Ddw，the key clause Dd of the statement clause can be calculated as

$$D_{dn} = \frac{N(W_i \cup W_j)}{N(W_i) + N(W_j) - N(W_i \cup W_j)} \tag{5}$$

$$D_{dw} = \frac{\sum_{i=1}^{N(W_i)} \sum_{j=1}^{N(W_j)} \frac{MIN(\gamma_i, \gamma_j) MIN(P_{rw}(w_{mi})\alpha_i, P_{rw}(w_{mj})\beta_j)}{MAX(P_{rw}(w_{mi})\alpha_i, P_{rw}(w_{mj})\beta_j)}}{\sum_{i=1}^{N(W_i)} \sum_{j=1}^{N(W_j)} MIN(\gamma_i, \gamma_j)} \tag{6}$$

$$D_d = D_{dn} \times D_{dw} \tag{7}$$

## 5. Experimental results and analysis

### *5.1. Complexity and Performance Analysis of Conceptual Dependency Forest Construction and Updating Algorithm*

The experimental data is shown in Table 1.

Table 1. Algorithm Execution Time Results

| Concept node | 100 | 207 | 420 | 892 | 1849 | 3834 | 7949 | 16483 |
|---|---|---|---|---|---|---|---|---|
| execution time | 4.86 | 9.72 | 19.68 | 40.96 | 87.23 | 175.45 | 378.61 | 809.53 |

As seen from Table 1, the execution time of the algorithm is basically linear with the number of ergodic nodes. It can be seen that although the algorithm consists of multiple recursive parts, the time complexity does not increase rapidly due to the introduction of the recursive process, and it remains within the controllable range.

According Table 1, the time complexity of algorithm 1 is about O (n). As a more complex formal formalism model, the conceptual dependency forest has the complexity of being close to linear time to construct and update it. This is a major advantage of the conceptual dependency forest model compared to formal formalism model of other statements.

### *5.2. Extraction Experiment and Result Analysis of Forest Operation*

The experimental corpus is from "Compilation of B City Medical Insurance Policy Documents" and the document contains 238 medical insurance policy documents promulgated by B City in 2001-2006. Choose to evaluate the performance of a formal representation using the two most commonly used PARSEVAL criteria:

Table 2. Parsing Results of Three Models

| Model | Length | ≤ 10 | ≤ 15 | ≤ 20 | ≤ 30 | ≤ 50 |
|---|---|---|---|---|---|---|
| Literature model | dependency accuracy rate(%) | 82.7 | 76.6 | 72.7 | 69.8 | 69.3 |
| | root accuracy rate(%) | 88.1 | 80.6 | 76.1 | 71.2 | 70.4 |
| Refinement requirement tree model | dependency accuracy rate(%) | 83.4 | 78.1 | 75.0 | 72.1 | 71.3 |
| | root accuracy rate(%) | 89.4 | 83.2 | 79.0 | 74.3 | 74.0 |
| Dependency parsing tree model | dependency accuracy rate(%) | 87.2 | 81.4 | 78.8 | 75.9 | 74.7 |
| | root accuracy rate(%) | 91.8 | 85.0 | 81.7 | 77.2 | 77.0 |

Table 2 lists the results of the work on a number of related models. In most of the results in Table 2, the recall rate is significantly lower than the accuracy rate. One of the important reasons is that the experimental corpus is too small in scale: "B City Medical Insurance Policy Compilation" only have 5,950 sentences, less than 1/7 of the size of the PTB treasury. The above experiments are basically conducted on sentences that contain only participles and no parts of speech.

For the experimental corpus, the total number of conceptual nodes is 892, and the algorithm identifies 880, accounting for 98.65% of the total; 734 of them are correctly identified, which is 82.29% of the total.

Since the syntactic analysis is carried out on sentences with part-of-speech marks, the experimental results in this paper are the most comparable to the DOP method. The concept-dependent forest model proposed in this paper can analyze and represent the relationship between concepts in the absence of large-scale corpus.

On the other hand, for the 5,950 sentences in the corpus, the number of conceptual nodes is only 892, and the number of nodes is only about 15% of the number of sentences. The 207 conceptual dependency trees formed by these nodes are only 3.48% of the sentences. Therefore, the establishment of a conceptual level of sentence description is a better method than the syntactic and dependency annotation of the sentence clause directly.

## 6. Conclusions

The innovative research results of this paper are as follows:

- Based on the conceptual dependency theory, we propose the definition and structure of the conceptual dependency tree, as well as the definition, construction and update algorithm of the conceptual dependency forest. The experimental results show that the algorithm has linear time complexity and collected the forest operation function. The algorithm solved the problem of conceptual dependency relation.

- Based on the conceptual dependency forest model, we proposed the definition of conceptual dependency strength and its calculation method. The definition and calculation method of inside sentence similarity put forward the definition and calculation method of potential similarity. The concept similarity relation between dependency and non-dependency nodes of a dependency node can be measured.

- We proposed sentence dependency issues and defined the dependency classification and correlation calculation method equations (5,6,7) based on a conceptual dependency forest model. The application example proves that the system of the consistency verification theory proposed in this paper is effective.

## Acknowledgements

This work is sponsored by the Postdoctoral Science Initial Funds of Heilongjiang under grant number LBHQ15031 and the Education Scientific Project of Heilongjiang under grant number GJC1215107.

## References

1. S. L. Beum, R. B. Barrett, “Automated Conversion from Requirements Documentation to An Object-Oriented Formal Specification Language,” In Proceedings of SAC 2012, Madrid, Spain, ACM, 2012: 23-32
2. G. R. Carlos, “On the Relation Between Dependency Distance, Crossing Dependencies, and Parsing,” Physics of Life Reviews, 2017
3. W. L. Chen, M. Zhang, Y. Zhang, and X. Y. Duan, “Exploiting Meta Features for Dependency Parsing and Part-Of-Speech Tagging,” Artificial Intelligence, 2016, 230
4. Y. C. Cheng, “Machine Learning-based Dependency Analyzer for Chinese”. Proceedings of the International Conference on Chinese Computing. Singapore. 2005: 66-73
5. R. Schank, “Conceptual Information Processing,” America Elsevier Publishing Company Inc, 1975: 187-201
6. Z. Z. Shukur and A. M. Ban, “A Tool for Translating a Natural Language Software Specification into Z. International Conference on Formal Methods and Software Engineering”. ICFEM 2012, 2012: 406-410
7. H. Yamada, “Statistical Dependency Analysis with Support Vector Machines,” in Proceedings of the 8th International Workshop on Parsing Technologies, Nancy, France, 195-206, 2013

**Gang Liu** is an Associate Professor and graduate student tutor from Harbin Engineering University. His main research areas are artificial intelligence, machine learning, and natural language processing.

**Hanwen Zhang** is a Master’s student from Harbin Engineering University. His main research areas are machine learning and natural language processing.

**Hanmo Zhang** as a graduate student admitted to Harbin Engineering University. His main research areas are natural language processing and machine learning.