ISSN 1870-4069

The Fix-point of Dependency Graph – a Case Study of Chinese-German Similarity

Tiansi Dong[1], Armin B. Cremers[1], Juanzi Li[2], Peiling Cui[3]

[1] University of Bonn,
B-IT,
Germany

[2] Tsinghua University,
Department of Computer Science,
P. R. China

[3] University of Bonn,
Department of Chinese,
Germany

dongt@bit.uni-bonn.de, abc@iai.uni-bonn.de,
lijuanzi@mail.tsinghua.edu.cn, pcui@uni-bonn.de

Abstract. Applications using linguistic dependency graphs (LDG) produce exciting results in neural-NLP and machine translation, as it captures more semantic information. A case study is carried out to make *language dependency graphs*(LDG) closer to meaning representations. We designed graph-transformation rules, which remove some syntactic covers, and keep updating an LDG, till a fix-point is reached, which we name *spatial linguistic graph* (SLG). The formal definition of SDG is presented. An evaluation using SimRank-based method is conducted using paired German-Chinese sentences in a grammar book (totally 682 paired sentences). Results show that SDGs of paired sentences are more similar than that of LDGs – supported by 89.4% observations. After removing 66 invalid inputs, the support reaches to 99.03%. Comparison with related work is presented. Applications of SDG in word-embedding and Machine Translation are described.

Keywords: Linguistic/spatial dependency graph, fixed-point, SimRank-similarity.

1 Introduction

Recent NLP applications using linguistic dependency graphs (LDG) produce quite exciting and positive results. For example, in neural network computing, the word-embedding based on dependency graph is more accurate than sequence model, i.e., [19]; graph-based learning improves the quality in statistical machine translation, [2]; dependency-based machine translation demonstrates significantly better results than the phrase-based model, i.e., [28, 20]. The reason lies the fact that LDG captures long-distance word relations, and reveals more explicitly semantic relations.

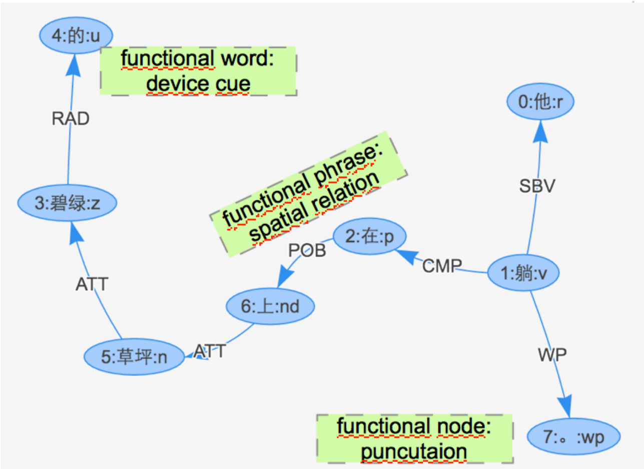
Fig. 1. A scene of a man lying on the grass.

The questions raised in this paper are: How close (or far away) is an LDG to the semantic representation? Can we make an LDG closer to its intended semantic representation?

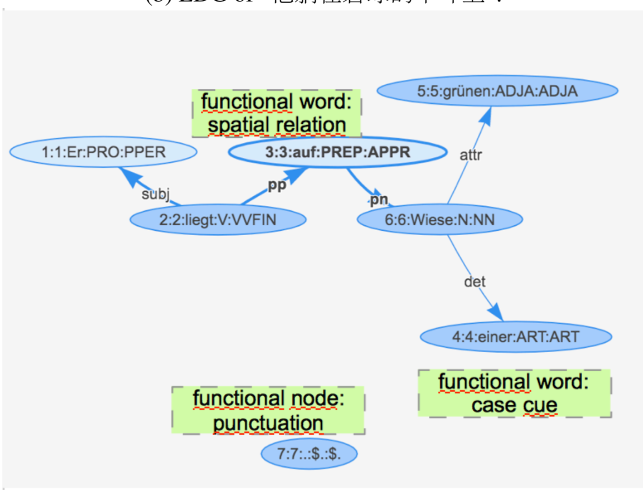
Let us start with a thought experiment is as follows: a bilingual speaker observes a scene, as illustrated in Fig 1, and describes the scene in two languages. He may give descriptions in two languages: For the Chinese description is "他/he 躺/lies在碧绿/green的草坪/lawn上。", its German description is " Er liegt auf einer grünen Wiese.". Underlined words are content words. How similar are the two dependency graphs? Can we make them more similar? We would assume that his descriptions have the same meaning. Their raw linguistic structures are shown in Figure 2 (a,b); the intended semantic structure shall be ideally very similar, as illustrated in Figure 3 (a, b).

The theoretical background can be found in the research of the relations between language and space. Language and thoughts are rooted in the knowledge of the space around us, [12]. Space structures linguistic descriptions, [31]; On the other hand, linguistic descriptions select and highlights some aspects in the space, i.e., [30]. Given a description about the same spatial layout, can we extract its spatial semantics? Research can be dated back to the work on relations between language form and meaning, in term of *cue*, i.e., [21]. Cross-linguistic studies showed that each language uses a particular set of cues: Italian extremely relies on agreement cues, German relies on both agreement cues and animacy cues, English relies overwhelmingly on word order, e.g., [22], Chinese relies on cues of word order, passive marker 被/by, animacy, object marker 把/hold, indefinite marker 一/one, i.e., [21].

The work reported in this paper is a case study in the setting of Chinese and German – two languages that use completely different cues. We collected manually all paired-sentences (682 pairs) in a Grammar book, [29], so that different linguistic structures are covered, and transformed them into dependency graphs using existing tools. Then, we manually designed sub-graph transformation rules by walking through all these sentences, based on different cues in the two languages. We also proposed SimRank-based similarity measurement to evaluate the results. The experiment system transforms the structures in Figure 2 (a, b) into Figure 3 (a, b), which we call *spatial dependency graph* (SDG).



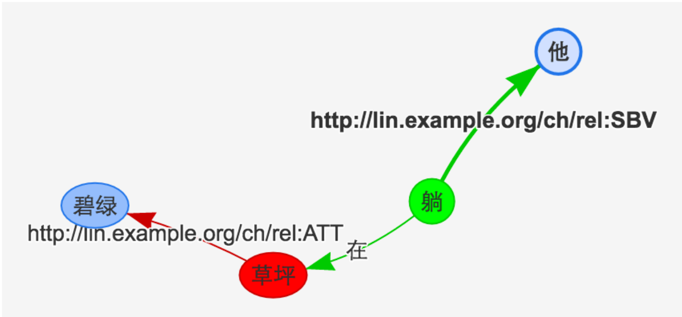
(b) LDG of “他躺在碧绿的草坪上”.



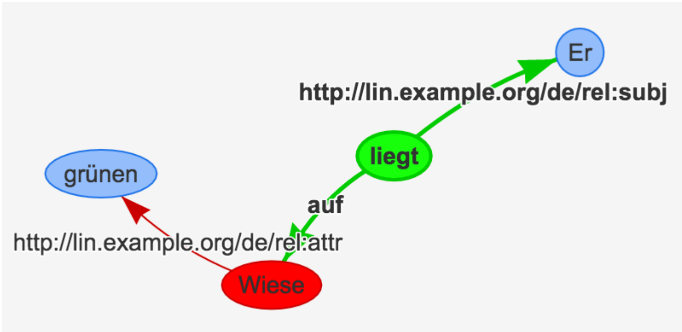
(b) LDG of “er liegt auf einer grünen Wiese”.

Fig. 2. The linguistic dependency graphs of a paired sentences. (a) is based on http://www.ltp-cloud.com/; (b) is based on http://kitt.cl.uzh.ch/kitt/parzu/.

The rest of the paper is structured as follows: section 2 defines the spatial dependency graph (SDG) as a fixed point of graph-transformations of LDG, and proposes the similarity conjuncture – if SDG is closer to semantic representation, SDGs of paired sentences shall be more similar than their LDGs; section 3 describes the SimRank-based method for graph comparison; section 4 show the experiment using paired sentences in a Chinese-German grammar book; section 5 listed some related work in the literature; section 6 presents two promising applications of SDGs: word-embedding, and machine translation.



(a) SDG of “他躺在碧绿的草坪上”.



(b) SDG of “er liegt auf einer grünen Wiese”.

Fig. 3. The spatial dependency graphs Fig 2 (a) and Fig 2 (b).

2 Spatial Dependency Graph and the Cross-linguistic Similarity Conjuncture

The starting point is language dependency graph (LDG) of a sentence, e.g., [18, 9]. Our work is to identify pure-syntactic structure (from a single node, to a sub-graph) in LDG, and repeatedly update them, till a fixed-point is reached. If it is a leaf node, we remove it, and set it as a feature of its parent node; if it bridges two nodes, we create a new edge from its parent node to its child node, and move it as one label of this node edge, as illustrated in Figure 2 and Figure 3. We formally define a LDG as follows.

Definition 1 A language dependency graph, LDG, is a *directed*, *labeled*, and *non-cyclic* graph $g = \langle V, E, \mu, \nu \rangle$, V is the set of nodes, E is the set of *directed*, *labeled*, and *non-cyclic* edges between nodes; $\mu : V \to \vec{F}$ is a function for node features – at least three feature: position, word form, and parts of speech; $\nu : E \xrightarrow{l} L$ is an edge-labelling function. POS is the set of parts of speech, POS_{cnt} is the set of parts of speech of content words, fun is the set of parts of speech of functional words, satisfying $POS = POS_{cnt} \cup POS_{fun}$ and $POS_{cnt} \cap POS_{fun} = \emptyset$. A directed edge from u to v, labeled with $l = \nu(u, v)$, non-cyclic ($u \neq v$) is written as $u \xrightarrow{l} v \in E$. Edges are *deterministic*. That is, for any u, v in V, there is only one edge from u to v labeled with l ($u \xrightarrow{l} v \in E$). Formally, $\forall u, v, v' \in V[u \xrightarrow{l} v \in E \wedge u \xrightarrow{l} v' \in E \Rightarrow v = v']$.

Our general observation on the data-set of deciding whether a node is a *cue* is to observe the parts of speech of the word in the node, and the number of children. If it has two or more than two children, it is not a *cue* node; if the node has a functional word, and one or less than one child node, it is a *cue* node, and shall be updated. An intuitive understanding is that a node with more than two children will served as a semantic frame on the relation among itself and the children. To identify whether a node is a *cue*, we need to consider *pattern* of, and around a node, as defined below.

Definition 2 Let $\vec{f}$ be a feature vector of node, at least one component has non-empty value, l be a non empty label. A pattern of $g = \langle V, E, \mu, \nu \rangle$ is recursively defined as follows: (1) $\mu(?v) = \vec{f}$ is a pattern, read as 'a node $?v$ has feature $\vec{f}$'; (2) $\nu(?u \rightarrow ?v) = l$ is a pattern, read as 'an edge from $?u$ to $?v$ having label l; (3) if c is a pattern, $\neg c$ is a pattern; (4) if c_1, c_2 are patterns, $c_1 \wedge c_2$, $c_1 \vee c_2$ are patterns; (5) patterns are only in these forms. $?v$, $?u$ are variables. Let c be a pattern, $var(c)$ is the set of all the variables in c, $node(c)$ is the set of all nodes in c.

After having identified a pattern in a graph, we need to bind variables in the pattern with the concrete nodes in the graph. Inversely, if we have a pattern, and a list of node assignment of its variable, we can construct a concrete graph from this pattern using these assignment.

Definition 3 Let c be a pattern, g be a graph has pattern c. The *binding* of pattern c with graph g is the assignment of each element in $var(c)$ by the corresponding node of g. A graph *construction*, based on pattern c, using bindings b, is to replace node variables in c with the bounded values defined in b.

A process of updating *cue* node is then a process of pattern identification within a graph, get binding to the graph, and construct a new sub-graph based on a new pattern.

Definition 4 A *transformation* $f_{c_1 \rightarrow c_2}$, satisfying $var(c_1) = var(c_2)$, is a function from graph to graph – to replace each sub-graph with c_1 pattern with a graph constructed by c_2 pattern using the binding of pattern c_1.

We strictly require that $var(c_1) = var(c_2)$, instead of $var(c_1) \supseteq var(c_2)$, so that inverse operation can be defined. For the transformation $f_{c_1 \rightarrow c_2}(g) = g'$, some nodes in g may be placed as label information in g', instead of being simply removed.

Definition 5 The *Identity transformation* f_{id} is defined as $f_{c \rightarrow c}$, for any c.

It is obvious that for any graph g, pattern c, $f_{c \rightarrow c}(g) = g$.

Definition 6 Let $f_{c_1 \rightarrow c_2}$ and $f_{c_3 \rightarrow c_4}$ be two transformations, its inverse transformation $f^{-1}_{c_1 \rightarrow c_2}$ is defined as $f_{c_2 \rightarrow c_1}$. The iteration, $f_{c_1 \rightarrow c_2} \circ f_{c_3 \rightarrow c_4}$, is defined as: for any g, $f_{c_1 \rightarrow c_2} \circ f_{c_3 \rightarrow c_4}(g) = f_{c_1 \rightarrow c_2}(f_{c_3 \rightarrow c_4}(g))$.

Theorem 1 Transformation is associative.

Proof 1 Let f_1, f_2, f_3 be three transformations, $(f_1 \circ f_2) \circ f_3(g)$= $(f_1 \circ f_2)(f_3(g))$ =$(f_1(f_2(f_3(g))))$ =$(f_1(f_2 \circ f_3(g)))$ = $(f_1 \circ (f_2 \circ f_3)(g))$.

If g has two disjoint patterns c_1 and c_3, $f_{c_1 \to c_2}$ and $f_{c_3 \to c_4}$ will operate on non-intersected sub-graphs of g. So, $f_{c_1 \to c_2} \circ f_{c_3 \to c_4}(g) = f_{c_3 \to c_4} \circ f_{c_1 \to c_2}(g)$. However, we do not set this restriction to the transformation operation. So, the transformation operation on graphs forms a non-commutative group (non-Abelian group).

Definition 7 Let f be an iteration of transformations $f_1 \circ \ldots f_n$, g be a graph. Define sequence $\{g_i\}$ as follows:

$$g_1 = g, \tag{1}$$
$$g_{i+1} = f(g_i). \tag{2}$$

If $g_{n+1} = g_n = f^{n-1}(g_1)$, g_n is the fixed point of graph g under f.

We call $f_{c_1 \to c_2}$ a em zooming-out transformation, if $node(c_1) \supset node(c_2)$. $f_{c_1 \to c_2}$ is *applicable* for graph g, if $g(c_1)$ is not empty.

Definition 8 Let f be an iteration of transformations $f_1 \circ \ldots f_n$, where f_i is zooming out. g be a language dependency graph. The spatial dependency graph (SDG) is the fixed point of g under f, written as $\mathcal{S}(g)$.

We need to prove the existence and the uniqueness of $\mathcal{S}(g)$.

Theorem 2 Let f be an iteration of transformations $f_1 \circ \cdots \circ f_n$, where f_i is zooming out. For any graph g, there is an m such that $f^m(g)$ is a fixed point.

Proof 2 If f is not applicable for g, then $g = f(g)$. Otherwise let f_m be the first transformation in $f_1 \circ \cdots \circ f_n$ which is applicable for g. Let $g' = f_m(g)$, as f_m is zooming out, f_m is not applicable for g'. So, after a maximum of n iterations, there will be no f_i applicable for $f^{n-1}(g)$. So, $f \circ f^{n-1}(g) = f^{n-1}(g)$.

Theorem 3 Let f be an iteration of transformations $f_1 \circ \cdots \circ f_n$, where each f_i is zooming out. For any graph g, there is only one fixed point under f.

Proof 3 Suppose there are two different fixed points: $g_{p+1} = g_p = f(g_{p-1})$, $g_{q+1} = g_q = f(g_{q-1})$, and $p > q$. It is obvious that $g_p = f(g_{p-1}) = f^2(g_{p-2}) = ... = f^{p-q-1}(g_{q+1}) = f^{p-q-2} \circ f(g_{q+1}) = f^{p-q-2} \circ f(g_q) = \cdots = g_q$.

The construction of spatial dependency graph can be understood as a process of repeatedly taking-off syntax parts of *cue*s in the way of moving them into edges, resulting in a graph of relations among content words [31]. Proposed the Schematization Similarity Conjecture as follows: *to the extent that space is schematized similarly in language and cognition, language will be successful in conveying space*. Put their conjecture in the cross-linguistic setting – if language is successful in conveying space, descriptions in different languages about the same space shall share some similarity in content and structure, and their semantic representation shall share more similarity after removing syntactic cloths. We call it the *cross-linguistic similarity conjecture*, which is used to examine the soundness of spatial dependency graph. In the next section, we will introduce a method to measure the similarity between dependency graphs.

3 Similarity Between Directed Graphs

Several methods in the literature address the similarity measurement of graphs. One method is purely based on structures: two graphs are similar if they are isomorphic, or one is isomorphic to a sub-graph of the other, or they have isomorphic sub-graphs, [26]. This isomorph-based method is too strong for our case, as most of LDGs of Chinese and German sentences shall belong to the non-isomorphic case – no sub-graphs of their LDGs are isomorphic.

A weaker version is to compute a *graph edit distance* – the similarity between two graphs is measured by minimal cost of transforming one of the two graphs into the other. Basic graph operations shall be defined, such as adding/deleting/substituting nodes/edges, or reversion of edges.

Each operation has an associated cost. Sequences of operations are searched to match one graph to the other. Both of the above methods set priority to the structural information, however, in our task setting, word-to-word matching affects structural matching – if the word x in graph A can only be mapped (translated into) the word y in graph B, the node x must be mapped with node y. In our setting, similarity between graphs is related with contents of nodes – either word-to-word translation suggests mapping of nodes, or mapped nodes affect mapping of unknown words.

A related graph similarity measurement is the SimRank method of [17]. Given two objects, a, b, SimRank method defines the similarity between a, b as follows:

$$s(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b)), \quad (3)$$

where, a, b are nodes of graph G_1 and graph G_2, respectively (G_1 and G_2 can be the same). $C \in (0, 1]$ is a constant, $I(x)$ is the set of in-neighbors of object x. The similarity between a and b is determined by similarities of its in-neighbors. If a or b does not have in-neighbors, $s(a, b)$ is defined as 0. The SimRank algorithm iterates many times over the nodes of G_1 and G_2, to compute the similarity values between the nodes, till these similarity values converge.

The SimRank method meets our setting for reasons as follow: (1) this method considers both content information of nodes and structural information of graphs; (2) it allows empty content information, and uses structural information to approximate similarities. The SimRank method does not meet our task setting in following aspects: (i) it sets the original similarity value to zero, if one of the nodes has no parent node; while in our task setting, we can use a word-to-word translation dictionary to set the original similarity value; (ii) it only considers in-neighbor nodes to make approximation, which is reasonable for the task setting of web-page similarity; while in our setting, similarity between nodes also affected by similarities of out-neighbors; (iii) it computes similarities between nodes of two graphs; while our task is to evaluate the similarity between two graphs. We need to somehow modify the SimRank method, to fit our task setting.

Let G_1 have N nodes $a_1, \ldots, a_N$, G_2 have M nodes $b_1, \ldots, b_M$; $node(G)$ be the set of all nodes of G; $init(a_i, b_j)$ is the in initial similarity value between a_i, b_j; $word(x)$ be a function which returns the word form of node x; $pos(x)$ be a function which returns the parts of speech of the word of node x; $DICT$ be a word-to-word translation dictionary from language L_1 into language L_2, that is, $DICT = \{(\text{word}^1_{L_1}, \text{word}^1_{L_2}), (\text{word}^2_{L_1}, \text{word}^2_{L_2}), \dots\}$.

We modify the SimRank method into our task setting in following aspects: (a) The original similarity value between two nodes is set to 1, if words of the two nodes exist in a word-to-word translation dictionary, otherwise set to 0:

$$init(a_i, b_j) = \begin{cases} 1, & \text{if } (a_i, b_j) \in DICT, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

(b) The Similarity value between two nodes, $s(a, b)$, are updated by considering similarity values of their both in-neighbors and out-neighbors, and the compatibility between their parts of speeches — we will multiply the sum by a constant value ($\lambda_{pos} \in (0, 1)$). λ_{pos} is understood as the confidence parameter for the similarity between two unknown words, if we only know their parts of speeches. $\lambda_{pos} = 0.36$ is used in this experiment:

$$s_I(a, b) = \begin{cases} \dfrac{C \sum\limits_{i=1}^{|I(a)|} \sum\limits_{j=1}^{|I(b)|} s(I_i(a), I_j(b))}{|I(a)||I(b)|}, & \text{if } |I(a)| * |I(b)| > 0, \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

$$s_O(a, b) = \begin{cases} \dfrac{C \sum\limits_{i=1}^{|O(a)|} \sum\limits_{j=1}^{|O(b)|} s(O_i(a), O_j(b))}{|O(a)||O(b)|}, & \text{if } |O(a)| * |O(b)| > 0, \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

$$s(a, b) = \begin{cases} \lambda_{pos} \frac{s_I(a,b) + s_O(a,b)}{2}, & \text{if } s(a, b) < 1 \wedge pos(a) = pos(b), \\ 0, & \text{if } s(a, b) < 1 \wedge pos(a) \neq pos(b), \\ 1, & \text{otherwise.} \end{cases} \tag{7}$$

(c) The similarity between two graphs is computed by averaging similarity values between nodes:

$$s(G_1, G_2) = \frac{\zeta \sum\limits_{a \in node(G_1)} \sum\limits_{b \in node(G_2)} s(a, b)}{|node(G_1)||node(G_2)|}. \tag{8}$$

ζ is a normalization constant. $\zeta = 1.5$ is used in this experiment.

4 Experiment and Evaluation

In this section, we describe an experiment to evaluate the similarity conjecture on Chinese-German, two languages with totally different grammatical structures.

We take paired Chinese-German sentences in a grammar book as the test set. Our target is to computationally evaluate whether paired sentences have more similar SDGs than their LDGs.

Table 1. RDF prefix of Chinese dependency graphs.

`@prefix`	`ch:`	<http://lin.ch2de.org/ch/>
`@prefix`	`word:`	<http://lin.ch2de.org/ch/word/>

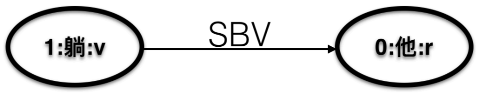


Fig. 4. A fragment of an LDG, where the main verb is taken as the predicate in the RDF representation.

4.1 Representing Graph in the RDF Format

To understand a text, we may need to interlink relations among words in neighborhood sentences, or need to integrate some background knowledge. This encourages us to use the RDF format, which is designed for interlinking knowledge across internet, and has been widely used in semantic web, e.g., [5, 3, 14]. An LDG contains knowledge of word features (represented as node feature), and knowledge of relations among words (represented as relations between nodes).

For example, in Figure 2(a), "`5:草坪:n`" contains three pieces of information as follow: (1) the form of the word: 草坪(lawn), (2) this word is located at the 6th position in the sentence (the first position is denoted by '0'), (3) its parts of speech is noun (`n`). For the RDF representation, we introduce a name stem <http://lin.ch2de.org/ch/>. The RDF form of a Chinese word should start with <http://lin.ch2de.org/ch/word>, and multiply 100 to the position number.

For example, the second word will be named as <http://lin.ch2de.org/ch/word/100>. We introduce <http://lin.ch2de.org/ch/form> as the predicate **has-form**, and <http://lin.ch2de.org/ch/pos> as the predicate **has-pos**. As LDG is represented in the RDF format, graph transformation will be carried out by SPARQL queries, e.g., [11]. Spatial dependency graph is the fixed-point of a linguistic dependency graph after a sequence of graph transformations. The RDF prefix is listed in Table 1.

An LDG is transformed into a set of RDF triples, each has the subject-predicate-object format. As we are aiming at semantic representation, we follow the approach used in the frame-net, i.e. [13], [12], and view verbs as semantic schema, which has category roles. For example, the `lie` schema has the actor role, which should be in the subcategory of living things. In Fig 4, the verb 躺(lie) is taken as the predicate with two arguments 他(he) and `SBV` (subject). Formally, we have:

$$\mathbf{word\_100}(\mathbf{word\_0}, \texttt{SBV}), \tag{9}$$

and the RDF form:

$$\texttt{word:0 word:100 rel:SBV}. \tag{10}$$

Table 2 illustrates a part of the RDF representation of Figure 2(a).

Table 2. RDF representation of the dependency graph in Figure 2 (a).

@prefix	ch	<http://lin.ch2de.org/ch/>
@prefix	word	<http://lin.ch2de.org/ch/word>
@prefix	rel	<http://lin.ch2de.org/ch/rel>
word:0	ch:form	他 `#he`
word:0	ch:pos	r
word:0	word:100	rel:SBV
word:100	ch:form	躺 `#lie`
word:100	ch:pos	v
word:100	word:dummy	ROOT

```
prefix de:      <http://lin.ch2de.org/de/>
prefix word:    <http://lin.ch2de.org/de/word/>
prefix rel:     <http://lin.ch2de.org/de/rel:>

ASK {
 ?n ?fuehren ?obj0 .
 ?fuehren de:base "fuehren" .
 ?regie ?n rel:app .
 ?regie de:base "Regie".
}
```

Listing 1.1. SPARQL query to fixed phrase pattern.

4.2 Rule-based Transformation Using SPARQL Queries

The advantage of using the RDF format is that graph transformation can be carried out by SPARQL queries. The basic operation is to search for a sub-graph pattern, and update this sub-graph, if found. Three kinds of sub-graphs can be distinguished: (1) sub-graphs for idioms and fixed expressions; (2) sub-graphs for grammatical rules; (3) sub-graphs containing nodes of functional words (f-node).

As each language has limited number of functional words, we have limited number of sub-graph patterns. Fixed phrases in one language may be expressed by just a word in another language. For example, the fixed phrase *Regie … führen* (serve as director) in German is expressed by one word 导演(direct) in Chinese. This pattern of fixed phrase can also be represented by SPARQL query as illustrated in List 1.1. We group components of fixed phrases together, and set the `rel:buddy` relation among them. For the above example, node *Regie* shall be moved to be a child of node *führen* with the relation `rel:buddy`, as illustrated in Figure 5.

The present research focuses on sub-graphs in (2) and (3). We went through all paired Chinese-German sentences in a Chinese grammar book for German speakers, i.e. [29], and manually designed 17 sub-graph patterns for the German sentences, and 11 sub-graph patterns for the Chinese sentences.

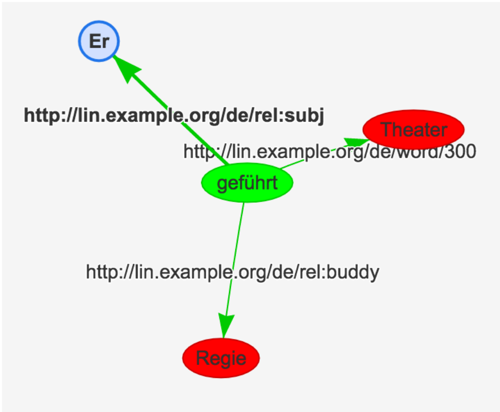


Fig. 5. The fixed phrase *Regie . . . führen* is identified and updated with a `rel:buddy` relation between node *Regie* and node *führen*.

4.3 Experiment Results and Analysis

Experiments are carried out by comparing the similarity between LDGs with the similarity of SDGs of 682 paired Chinese-German sentences in a Chinese grammar book for native German speakers, [29]. For Chinese sentences, we use LTP[1], i.e. [7]; for German sentences, we adopt Parzu[2], i.e. [27].

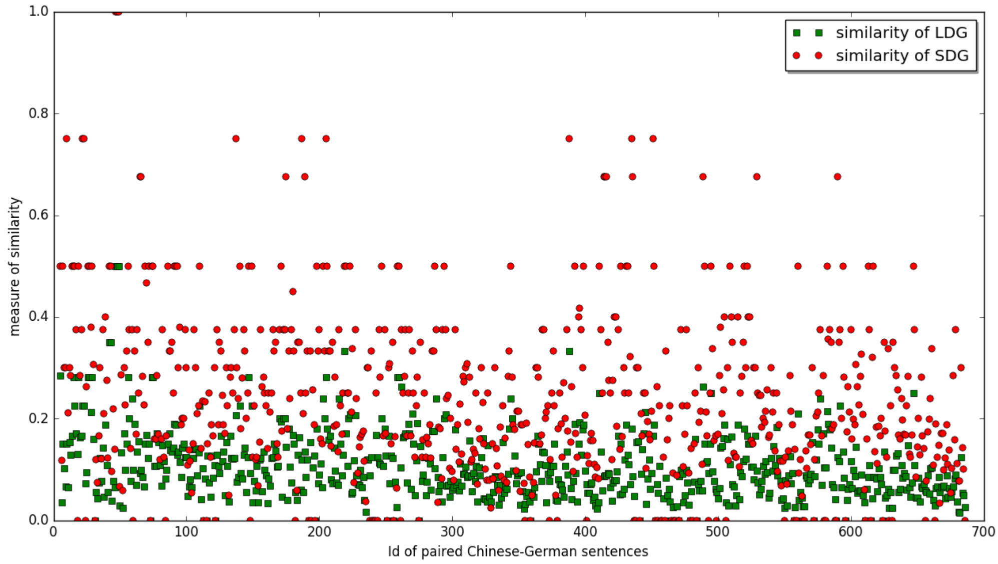
The result is illustrated in Figure 6: the SDGs similarity measures of 610 sentences are higher than the similarity of their LDGs. That is, 89.4% of the paired sentences support our hypothesis that SDGs of paired sentences shall be more similar than their LDGs. We examined the rest 10.6% paired sentences, and found that these 66 paired sentences have SDG similarity value 0.

That is, our dictionary does not have word-to-word translation of the content words for these 66 paired sentences, while functional words in paired LDGs appear in thedictionary. If we remove these 66 invalid inputs, our hypothesis will be supported by 610/(682-66) = 99.03% observations. The ratio between SDG similarity and LDG similarity is illustrated in Figure 6 (b).

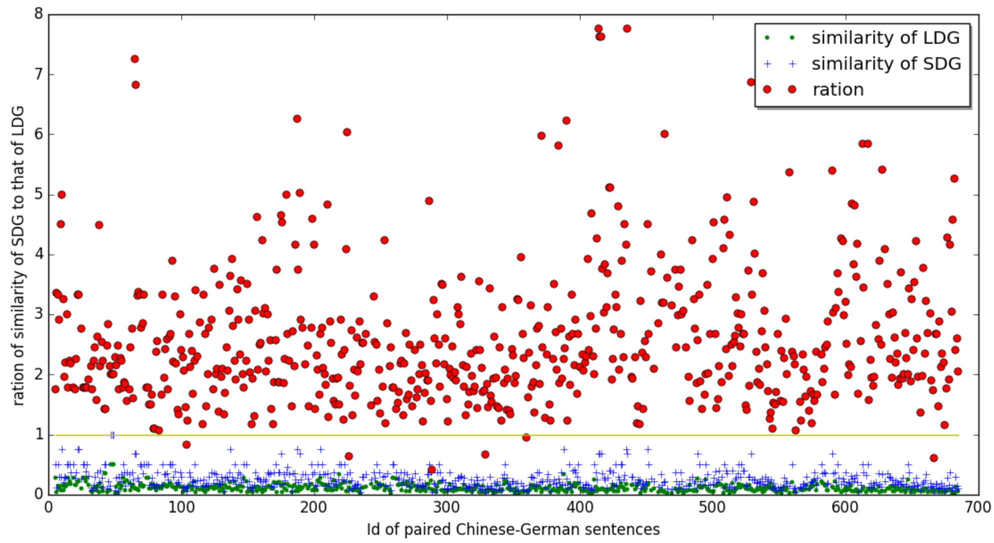
4.4 Error Analysis

There are six paired sentences, as listed in Table 3, whose similarity measures of the LDGs are larger than the non-zero similarity measures of their SDGs. Two main reasons are: (1) word-to-word level mapping are not found at dictionaries: for example, ‘5月’(‘May’) and ‘1日(‘the first’) do not have their German translation ‘Mai’ and ‘1.’ in the current dictionary we used; ‘放假’ (‘have vacation’) does not have the German translation ‘frei haben’ (‘have free’); (2) SDGs of paired sentences do have different structures, based on the current transformation method – in this experiment, we do not use sub-graph patterns of idioms and fixed phrases.

[1] http://www.ltp-cloud.com/
[2] http://kitt.cl.uzh.ch/kitt/parzu/



(a) Similarity comparison between LDGS and SDGs of paired Chinese-German sentences.



(b) Ratio of SDG similarity to LDG similarity.

Fig. 6. Similarity comparison after removing 66 non-valid observations. Below $y = 1$, there are six red points, which mean that similarity of LDGs is greater than the similarity of SDGs.

Table 3. List of the 6 paired sentences whose LDG similarity is larger than SDG similarity.

Id	Chinese sentence	German sentence
104	有一天晚上突然下了一场大雪。	Eins Abends hat es plötzlich stark geschneit.
226	这件事他几乎不知道。	Sie weiß fast nicht darüber.
289	我们这里从5月1日到5月7日放假。	Wir haben hier vom 1. Mai bis zum 7. Mai frei.
329	由于空气污染严重，人民难得见到蓝天和白云。	Wegen der schlimmen Luftverschmutzung sieht man selten blauen Himmel und weiße Wolken.
360	尽管这是一个小城，但是交通很方便。	Obwohl diese Stadt klein ist, sind die Verkehrsverbindungen sehr gut.
667	他身体还没有完全恢复，不过好多了。	Er ist noch nicht ganz gesund, aber es geht ihm viel besser.

5 Related Work

Our work is related with the research on semantic dependency graph, e.g., [15, 1]. The main difference is that our work uses node information of LDG as edge lables in SDG, instead of introducing semantic labels at the very beginning. Our method introduces the flexibility in semantic analysis: under concrete contexts or demands, we can introduce external semantic information, for example, from [15], or some emotional categories.

The idea of updating a structure until a fixed point is not new in the NLP literature. Many rule-based machine translation systems used this idea in grammar rewriting, e.g., [24, 6, 16]. Our work can be understood as rewriting rules into the direction of semantic representation. Most work on semantic representation is relied on manual annotation. An ambitious and popular semantic representation is the Abstract Meaning Representation (AMR), i.e. [4], and Chinese AMR is also recently released[3].

[15] is a semantic representation of German, which is based on a large set of over 10-years' long hand-annotated semantic dictionary. [25]'s Proposition Bank trained semantic-roles based on annotated corpus of Penn Treebank. [8] presents a unified neural NLP system for four common tasks: POS tagging, Chunk, name entity recognition, and semantic role labelling (SRL). The SRL task was trained by using convolution neural network.

Semantic representation is normally carried out as a supervised machine learning task: To get the meaning representation of a form, we first need to have meaning representations of other forms (by manual annotation), establish some (statistical or neural) relations between them, and apply these relations for the new form to get its meaning representation.

Different from the above work, this work proposed in this paper attempts to approach to semantic representation using linguistic cues, instead of annotated semantic corpus. The small scale experiment shows SDGs of paired Chinese-German sentences are more similar than their LDGs.

[3] http://www.cs.brandeis.edu/ clp/camr/camr.html

Our on-going work is to experiment with large data-sets, i.e. English-Chinese, English-German paired corpus used in machine translation.

6 Application of Spatial Dependency Graphs in Word-embedding, and Machine Translation

Word-embedding plays a key-role in neural-based NLP. [19] shows that word-embedding based on dependency relation outperforms the word-embedding based on neighborhood-relations in the raw corpus, i.e. [23], for the reason that semantic-relations between words are more precisely captured in the dependency graph than in the literal sequence. Following this reason, the word-embedding based SDG shall out-perform the word-embedding based on raw dependency graph.

The second application of SDG is in Cue-based Machine Translation, as introduced in [10]. As SDG is acquired based on cues of a language, the SDG representation of a pair sentence approximates an aggregated meaning representation. The alignment between components of paired SDGs shall outperform that of paired raw dependency graph, i.e. [28].

Acknowledgments. We are indebted to anonymous reviewers for their critical and valuable comments. Partial financial support from NSFC under Grant No. 61472177 is greatly acknowledged.

References

1. Agić, Ž., Koller, A., Oepen, S.: Semantic dependency graph parsing using tree approximations. In: Proceedings of the 11th International Conference on Computational Semantics, pp. 217–227 (2015)
2. Alexandrescu, A., Kirchhoff, K.: Graph-based learning for statistical machine translation. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 119–127 (2009)
3. Allemang, D., Hendler, J.: Semantic web for the working ontologist: Effective modeling in RDFS and OWL. Morgan Kaufmann Publishers Inc., Elsevier (2011) doi: 10.1016/C2010-0-68657-3
4. Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., Schneider, N.: Abstract meaning representation for sembanking. In: Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, pp. 178–186 (2013)
5. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American, vol. 284, no. 5, pp. 34–43 (2001)
6. Boitet, C.: The french national MT-project: Technical organization and translation results of CALLIOPE-AERO. Computers and Translation, vol. 1, no. 4, pp. 239–267 (1986) doi: 10.1007/BF00936424
7. Che, W., Li, Z., Liu, T.: LTP: A Chinese language technology platform. In: Coling 2010: Demonstrations, pp. 13–16 (2010)
8. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. Journal of Machine Learning Research, vol. 12, pp. 2493–2537 (2011)

9. De-Marneffe, M. C., Grenager, T., MacCartney, B., Cer, D., Ramage, D., Kiddon, C., Manning, C. D.: Aligning semantic graphs for textual inference and machine reading. In: Proceedings of the AAAI Spring Symposium at Stanford, pp. 658-476 (2007)
10. Dong, T., Cremers, A. B.: A novel machine translation method for learning Chinese as a foreign language. In: International Conference on Intelligent Text Processing and Computational Linguistics, vol. 8404, pp. 343–354 (2014) doi: 10.1007/978-3-642-54903-8_29
11. DuCharme, B.: Learning SPARQL: Querying and updating with SPARQL 1.1. O'Reilly Media, Inc (2013)
12. Feldman, J.: From molecule to metaphor: A neural theory of language. The MIT Press (2006) doi: 10.7551/mitpress/3135.001.0001
13. Fillmore, C. J.: Scenes-and-frames semantics: Linguistic structures processing. Fundamental Studies in Computer Science, North Holland Publishing, vol. 5, no. 59, pp. 55–88 (1977)
14. Grobe, M.: RDF, Jena, SparQL and the semantic web. In: Proceedings of the 37th annual ACM SIGUCCS Fall Conference: Communication and Collaboration, pp. 131–138 (2009) doi: 10.1145/1629501.1629525
15. Helbig, H.: Knowledge representation and the semantics of natural language. Springer (2006) doi: 10.1007/3-540-29966-1
16. Nakamura, J. I., Tsujii, J. I., Nagao, M.: GRADE: A software environment for machine translation. Computers and Translation, vol. 3, no. 1, pp. 69–82 (1988) doi: 10.1007/BF02057968
17. Jeh, G., Widom, J.: SimRank: A measure of structural-context similarity. In: Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Minings, pp. 538–543 (2002) doi: 10.1145/775047.775126
18. Klein, D.: The unsupervised learning of natural language structure. PhD thesis, Computer Science Department, Stanford University (2005)
19. Levy, O., Goldberg, Y.: Dependency-based word embeddings. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, vol. 2, pp. 302–308 (2014) doi:10.3115/v1/P14-2050
20. Li, L., Way, A., Liu, Q.: Graph-based translation via graph segmentation. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 97–107 (2016) doi: 10.18653/v1/P16-1010
21. Li, P., Bates, E., MacWhinney, B.: Processing a language without inflections: A reaction time study of sentence interpretation in Chinese. Journal of Memory and Language, vol. 32, no. 2, pp. 169–192 (1993) doi: 0.1006/jmla.1993.1010
22. MacWhinney, B., Bates, E., Kliegl, R.: Cue validity and sentence interpretation in English, German, and Italian. Journal of Verbal Learning and Verbal Behavior, vol. 23, no. 2, pp. 127–150 (1984) doi: 10.1016/S0022-5371(84)90093-8
23. Mikolov, T., Yih, W. T., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 746–751 (2013)
24. Nakamura, J. I., Tsujii, J. I., Nagao, M.: Grammar writing system (GRADE) of Mu-Machine translation project and its characteristics. In: 10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics, pp. 338–343 (1984) doi: 10.3115/980491.980560
25. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: An annotated corpus of semantic roles. Computational linguistics, vol. 31, no. 1, pp. 71–106 (2005) doi: 10.1162/0891201053630264
26. Pelillo, M.: Replicator equations, maximal cliques, and graph isomorphism. Neural Computation, vol. 11, no. 8, pp. 1933–1955 (1999) doi: 10.1162/089976699300016034

27. Sennrich, R., Schneider, G., Volk, M., Warin, M.: A new hybrid dependency parser for German. In: Proceedings of the Biennial GSCL Conference, pp. 115–124 (2009) doi: 10.5167/uzh-25506
28. Shen, L., Xu, J., Weischedel, R.: String-to-dependency statistical machine translation. Computational Linguistics, vol. 36, no. 4, pp. 649–671 (2010) doi: 10.1162/coli_a_00015
29. Song, J.: PONS grammatik chinesisch. PONS GmbH (2007)
30. Talmy, L.: How language structures space. Spatial Orientation, pp. 225–281 (1983) doi: 10.1007/978-1-4615-9325-6_11
31. Tversky, B., Lee, P. U.: How space structures language. Spatial Cognition, vol. 1404, pp. 157–175 (1998) doi: 10.1007/3-540-69342-4_8