Extracting Various Classes of Data From Biological Text Using the Concept of Existence Dependency

Kamal Taha, *Senior Member, IEEE*

***Abstract*—One of the key goals of biological natural language processing (NLP) is the automatic information extraction from biomedical publications. Most current constituency and dependency parsers overlook the *semantic relationships* between the constituents comprising a sentence and may not be well suited for capturing complex long-distance dependences. We propose in this paper a hybrid constituency–dependency parser for biological NLP information extraction called EDC_EDC. EDC_EDC aims at enhancing the state of the art of biological text mining by applying novel linguistic computational techniques that overcome the limitations of current constituency and dependency parsers outlined earlier, as follows: 1) it determines the *semantic relationship* between each pair of constituents in a sentence using novel semantic rules; and 2) it applies a *semantic relationship extraction model* that extracts information from different structural forms of constituents in sentences. EDC_EDC can be used to extract different types of data from biological texts for purposes such as protein function prediction, genetic network construction, and protein–protein interaction detection. We evaluated the quality of EDC_EDC by comparing it experimentally with six systems. Results showed marked improvement.**

***Index Terms*—Text mining, information extraction, biological natural language processing (NLP), biomedical literature, dependency parsers.**

I. Introduction

NATURAL language processing (NLP) has become an important field within bioinformatics due to the rapidly growing biomedical literature [1]. The interest in biological NLP has been triggered by the vast amounts of biomedical publications available in databases such as PubMed [2]. One of the key goals of biological NLP is the automatic information extraction from these biomedical publications. Different NLP approaches have been applied to domains such as statistical analysis [3]–[5] and full syntactic analysis [6]–[8]. These methods include the extraction of various classes of data from biological texts, for purposes such as predicting protein function [9]–[14], detecting protein–protein interactions (PPIs) [15]–[17], and building disease-specific gene interaction networks for determining gene–disease associations [18]–[20].

Numerous parsers have been proposed to parse molecular biology data. These parsers can be divided in terms of their emphasis on grammatical and compositional elements into two categories: constituency and dependency [7], [21]. Constituency parsers perform syntactic analysis in a tree representation of the phrases comprising the sentence and the hierarchy in which these phrases are associated. The root node of the tree represents the sentence as a whole and the leaves of the tree represent words of the sentence. Constituency parsers aim at capturing the structural information for each sentence in the input corpus. In constituency parsing, lexical semantics aims at the analysis of meaning in the granularity of words, stems, suffixes, and prefixes [22]. Most of these parsers can recover structural information in accordance with an explicit grammatical theory [23]. Constituent trees have been used to solve problems such as pronoun resolution [24], labeling phrases with semantic roles such as RESULT, CAUSE, and EXPERIENCER [25], and the assignment of functional category tags such as MANNER and TEMPORAL [26].

Constituency parsers [21] overlook function tags when training. Dependency parsers aim at overcoming these problems. They analyze a sentence by determining the dependency between each pair of words. Each dependency has a type that reflects its grammatical function. They model language as a set of relationships between *words*, and build a graph for each sentence. A node in the graph represents a *word* and an arc represents a grammatical dependency connecting the words of the sentence to each other.

Constituency and dependency parsers have shown high level of sophistication. However, most of them suffer the following limitations: 1) they overlook the *semantic relationships* between the constituents comprising a sentence (*most current dependency parsers determine the structural relationships between terms*); 2) most current constituent parsers determine the relationships between individual words rather than between constituents defined as sequences of words *combined* to form units according to specific functional characteristics; and 3) both current constituency and dependency parsers may not be well suited for identifying complex phrases and long-distance dependences.

We propose in this paper a hybrid constituency–dependency parser for biological NLP information extraction called EDC_EDC (*E*xtracting *D*ata from *C*orpus using the *E*xistence *D*ependency *C*oncept). EDC_EDC overcomes the limitations of current constituency parsers and dependency parsers outlined earlier. It employs hybrid constituency and dependency parsing concepts. EDC_EDC applies novel linguistic computational techniques that overcome the limitations of current constituency and dependency parsers outlined previously, as follows: 1) it determines the *semantic relationship* between each pair of constituents in a sentence using novel semantic rules (*as opposed to most current constituency parsers, which determine the structural relationships between individual words*); and 2) it applies

Manuscript received August 5, 2014; revised November 22, 2014; accepted January 12, 2015. Date of publication January 19, 2015; date of current version November 3, 2015.

The author is with the Electrical and Computer Engineering, Department, Khalifa University, Abu Dhabi, United Arab Emirates (e-mail: kamal.taha@kustar.ac.ae).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JBHI.2015.2392786

semantic relationship extraction models that extract information from different structural forms of constituents in sentences. This proposed approach disambiguates by comparing the likelihoods of plausible relationship between each pair of constituents in a sentence based on similarity to formal sematic rules. EDC_EDC can extract different types of data from biomedical abstracts, for purposes such as protein function prediction, PPI detection, and gene–disease association detection. We present novel deterministic techniques for semantic relationship extraction. A demo of EDC_EDC is available at: http://ecesrvr.kustar.ac.ae:8080/ (click on "Gene Ontology 1").

II. Definition of Key Concepts Used in This Paper

Typically, a sentence goes from one function to the next through a sequence of changes in word order as well as auxiliaries. EDC_EDC conceives a sentence as a hierarchical structure composed of clauses, groups, and units called *constituents*. A constituent is a sequence of words *combined* to form a unit according to a specific functional characteristic. EDC_EDC identifies constituents based on a set of rules specifying which syntactic composition can take place in a constituent and how they can combine to produce the constituent at hand. Each constituent has a label called a *head* that indicates the grammatical role and category of the constituent. Below are the list of constituents' heads and the definition of these constituents.

1) *Statement Clause:* It corresponds to the whole sentence.
2) *Dependent Clause:* It is a clause that cannot stand-alone as a sentence. It has a subject and a predicate but it does not express a complete thought. It often begins with connecting words and contains "relative pronouns."
3) *Adverbial Clause:* It is a clause that consists of two or more clause functions, one of them is adverbial and the other is the predicator. It usually contains conjunctive adverbs such as "whereas" and "while." It falls into three major subclasses: adjuncts, disjuncts, and conjuncts.
4) *Object Clause:* It is a clause that consists of two or more clause functions, one of them is the predicator and the other is a term that arises from a situation expressed by the predicator. It usually contains relative pronouns such as "that," "which," and "who."
5) *Conjoint Clause:* It is a constituent linked with another constituent by a coordination.
6) *Conjunct:* It is an adverbial that relates constituents in a sentence. It usually contains grammatical conjunctions such as "and" and "or."
7) *Compound Unit:* It consists of two or more conjuncts connected by one or more coordinators.
8) *Subject Group:* It is a portion of a sentence that consists of a subject(s) and one or more dependents.
9) *Object Group:* It is a portion of a sentence that consists of an object(s) and one or more dependents.
10) *Adverbial Group:* It is a portion of a sentence that consists of an adverb(s) and one or more dependents.

Each sentence in an abstract is treated as a tree called *Part Of Sentence Tree* (*POST*). A POST is an ordered rooted tree that represents the syntactic structure of a sentence according to the hierarchical dependency relations between the constituents comprising the sentence. Each node in POST represents a constituent. Branches are arcs represent the dependency relations between the constituents. Each arc represents a "part-of" dependency relation from a head to one of its dependents. A *part-of* relation has a specific meaning and a *part-of* relation would only be added between nodes *A* and *B* if *B* is necessarily part of *A*: wherever *B* exists, it is as part of *A*, and the presence of the *B* implies the presence of *A* [27].

Each node in POST is assigned a *type* that represents the *class of lexical characteristics* of the constituent represented by the node. Nodes that have the same class of lexical characteristics are said to have the *same type*. The following are the classes of lexical characteristics/types.

1) If a constituent contains a predicate (i.e., a main verb and auxiliaries) and the arguments of the predicate (e.g., subject and object noun phrases), the type of its node is *Clause Type* (*CLT*). Therefore, the nodes of the following constituents are assigned CLT: a) Statement Clause; b) Dependent Clause; c) Adverbial Clause; d) Object Clause; and e) Conjoint Clause. Thus, all these constituents have the same type, which is CLT.
2) If a constituent consists of a subject(s) and a dependent(s), the type of its node is *Subject Type* (*SUBT*). Thus, the type of a Subject Group constituent is SUBT.
3) If a constituent consists of an object(s) and one or more dependents, the type of its node is *Object Type* (*OBJT*). Thus, the type of an Object Group constituent is OBJT.
4) If a constituent consists of an adverb(s) and one or more dependents, the type of its node is *Adverb Type* (*ADVT*). Thus, the type of an Adverbial Group constituent is ADVT.
5) If a constituent serves to relate a sentence to a previous sentence, the type of the node is *Conjunct Type* (*CONT*). Thus, the type of a Conjunct node is CONT
6) If a constituent consists of two or more conjoint, the type of the node is *Compound Type* (*CMPDT*). Thus, the type of a Compound Unit constituent is CMPDT.

In this paper's figures, each constituent's node is assigned a color that denotes its type for easy reference. For example, in the POST in Fig. 1, the Statement Clause node is assigned a red color to denote that its type is Clause Type. Thus, nodes that have the same type are colored with the same color. For example, in Fig. 1, the two Compound Unit nodes are assigned the same color as an indicative they have the same type.

III. Outline of the Approach and Its Applications

A. Outline of the Approach

The following is an overview of our approach in terms of the sequential processing steps taken by EDC_EDC to determine the relationship between two constituents based on their concurrences in biomedical abstracts:

1) Tokenizing an Abstract: In tokenization, texts are broken into words, constituents, and sentences. Then, each word is "tagged" with a Part Of Speech (POS). Sequences of words are grouped into phrases. Basis text chunks such as base NP are also identified. EDC_EDC is built on top of Stanford Tokenizer

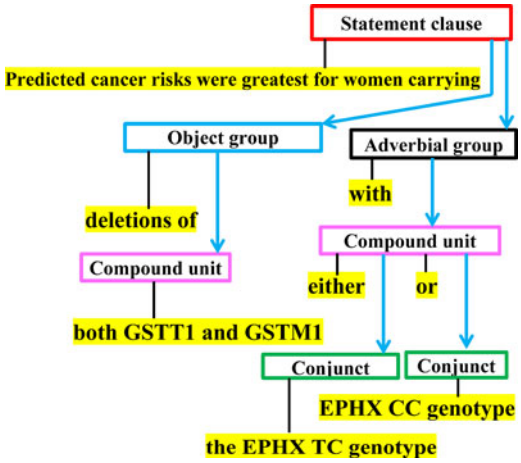

Fig. 1. POST for the sentence "Predicted cancer risks were greatest for women carrying deletions of both GSTT1 and GSTM1, with either the EPHX TC genotype or EPHX CC genotype." Arcs in the figure represent "part-of" relations between nodes and are represented by blue arrows. Black lines connect a node with a noun, verb, pronoun, modifier, etc. Each node in the figure is assigned a color that denotes its type for easy reference. Some of the specific constituents are merged in the figure for the sake of figure clarity.

and Part-of-Speech Tagger [8], for the purpose of tokenization and POS tagging.

2) Constructing POST: EDC_EDC identifies constituents in sentences by performing syntactic analysis to obtain a structure that specifies how words and phrases are related in the sentence. This is done after identifying the basic grammar that comprises the basic set of production rules such as sentence →"noun-phrase" "verb-phrase" and so on. All the allowable ways that each type of constituent can break down into parts are identified. This is done by building a dynamic programming matrix to construct the POST structure as follows. Each element holds the probability of a constituent. An array of back-pointers is set to contain the links between constituents. At the end of this step, POST is constructed along with the type/lexical characteristics of each node in the tree as described in Section II.

3) Determining Whether Two Nodes in POST Representing Constituent Are Semantically Related: EDC_EDC determines the *semantic* relationship between each two nodes in POST. That is, it determines the semantic relationship between each two constituents in a sentence. The semantic relatedness between two terms refers to the likeness of their meaning or semantic content. We determine the semantic relationship between nodes using the following observation. We observe that in order for the two nodes *A* and *B* in POST to be semantically related, the types of the nodes located in the path from *A* to *B* in POST should be unique. That is, in order for the two nodes *A* and *B* to be semantically related, there should not be two or more nodes in the path from *A* to *B* that have the same type. We validate this observation heuristically in Section IV.

4) Determining the Terms That Are Semantically Related to an Input Target Term Based on Their Co-occurrences in Abstracts: User submits to EDC_EDC a term T_i. EDC_EDC would return the significant set of terms that are semantically related to T_i based on the co-occurrences of the set and T_i in abstracts. Toward this, EDC_EDC applies novel semantic rule (recall step 3) to identify the terms that are semantically related to T_i in each sentence. EDC_EDC accepts keyword-based queries with the form Q("T_i," "R"), where:

1) T_i denotes an input target gene, gene product, disease name, or any biomedical term, and
2) R denotes the set of result terms that are semantically related to the input target term T_i, determined by applying the semantic rule outlined in step 3.

After inputting the term T_i, EDC_EDC first identifies the relevant named entities in the text using Stanford Named Entity Recognizer (NER) [28]. EDC_EDC is built on top of Stanford NER. NER tags biomedical terms such as protein, gene, or diseases names. The Co-reference Resolution connects occurrences of same proteins. Some of these occurrences are represented by terms such as "this protein," "it," "they," etc. Also, lexical peculiarities in protein names (such as symbols and numbers) are identified. Finally, EDC_EDC performs a domain analysis to identify the related entities as well as the nature of their relationships.

5) Assigning a Score to Each Candidate Result Term: Each candidate result term T_c is assigned a score reflecting the degree of its relationship (i.e., dependency) with the input term T_i. The score is denoted by $\mathrm{Score}(T_i, T_c)$. The value of the score is determined based on: 1) the co-occurrence frequency of T_i and T_c in abstracts; and 2) the semantic relationships between T_i and T_c in sentences. Let $f_{(T_i,T_c)}$ be the frequency of T_i and T_c being semantically related, and let $f'_{(T_i,T_c)}$ be the frequency of T_i and T_c being semantically unrelated. $\mathrm{Score}(T_i, T_c)$ is calculated using:

$$\mathrm{Score}(T_i, T_c) = f_{(T_i,T_c)} - f'_{(T_i,T_c)}. \quad (1)$$

$f_{(T_i,T_c)}$ and $f'_{(T_i,T_c)}$ are calculated using the contingency table shown in Table I, where:

1) $f_{(T_i,T_c)}$ is calculated by normalizing the sum of the squared deviations between the observed frequencies O_{11} and O_{12} and the theoretical frequencies e_{11} and e_{12}, where T_i and T_c are semantically related and may or may not co-occur in the same sentence. Thus, $f_{(T_i,T_c)} = \left(\frac{(O_{11}-e_{11})^2}{e_{11}} + \frac{(O_{12}-e_{12})^2}{e_{12}}\right)$.
2) $f'_{(T_i,T_c)}$ is calculated by normalizing the sum of the squared deviations between the observed frequencies O_{21} and O_{22} and the theoretical frequencies e_{21} and e_{22}, where T_i and T_c are semantically unrelated and may or may not co-occur in the same sentence. Thus, $f'_{(T_i,T_c)} = \left(\frac{(O_{21}-e_{21})^2}{e_{21}} + \frac{(O_{22}-e_{22})^2}{e_{22}}\right)$.

TABLE I
SEMANTIC RELATIONSHIP AND CO-OCCURRENCE FREQUENCY DISTRIBUTION FOR THE TWO TERMS/NOUNS T_i AND T_c

T_i and T_c co-occur in the same sentence	yes	no	Total
T_i and T_c are semantically related			
yes	O_{11}	O_{12}	R_1
no	O_{21}	O_{22}	R_2
Total	C_1	C_2	N

3) O_{11} Denotes the frequency where T_i and T_c co-occur in the same sentence and are semantically related.
4) O_{12} Denotes the frequency where T_i and T_c are semantically related and occur in two different sentences connected by a sentence connector (such as moreover, however, otherwise, therefore, etc.). In this case, the two sentences are represented by one POST and one root node.
5) O_{21} Denotes the frequency where T_i and T_c co-occur in the same sentence and are semantically unrelated.
6) O_{22} Denotes the frequency where T_i and T_c do not co-occur in the same sentence and are semantically unrelated.
7) $e_{ij} = \frac{R_i \times C_j}{N}$, where *N* is the grand total of observed frequencies (recall Table I for R_i and C_j).

As can be seen from (1), the greater the value of $\mathrm{Score}(T_i, T_c)$, the greater the semantic relatedness between T_i and T_c. A negative value indicates that the frequency of T_i and T_c being unrelated is greater than the frequency of being related. An acceptable value should be greater than a *heuristically determined threshold*.

B. Applications of the Approach

EDC_EDC can be used to extract various classes of data from biological texts, based on the biological classification domain of the input target term T_i in the query Q("T_i," "R"). The following are some of the applications of EDC_EDC.

1) *Predicting protein functions:* In this case, T_i represents an unannotated protein and R represents the set of annotated proteins that are semantically related to T_i based on the co-occurrences of R and T_i in abstracts. Set R is determined using the five sequential processing steps described in Section III-A, which EDC_EDC follows for determining the semantic relationships between terms. T_i will be assigned the functional category *f*, if the following is satisfied:
   The occurrences of set R in abstracts associated with proteins whose functional category is f is statistically significantly different than its occurrence in abstracts associated with proteins that belong to all other functional categories.
   EDC_EDC determines the significance of the occurrences of set R in abstracts associated with proteins belonging to different functional classes using Z-score. That is, it uses Z-score to determine the differences of the occurrences of *each* protein in set Rin abstracts associated with proteins that belong to different functional classes. The Z-score for a protein $T_c \in R$ and a protein whose functional class is *f* is the distance between the raw score for T_c and the population mean in units of the standard deviation [29]

$$Z - \mathrm{score} = \frac{x - u}{\sigma}. \tag{2}$$

   a) x: The raw score for T_c. It is calculated by dividing the number of abstracts that contain T_c and associated with proteins annotates with the function *f* by the total number of abstracts associated with proteins annotated with *f*.
   b) u: The population mean. It is calculated by dividing the number of abstracts that contain T_c and associated with each protein annotated with a function $f'' \neq f$ by the total number of abstracts associated with proteins annotated with f''.
   c) σ: The standard deviation of the population.
2) *Detecting PPI:* Prior information on PPI is important for the successful reconstruction of large PPI network. Therefore, we first select pairs of proteins (i.e., seed proteins) that are known to interact together. We extract these pairs from databases such as the database of interacting proteins [30]. We then submit these pairs as search terms to PubMed [2] to retrieve the set of abstracts associated with them. Then, EDC_EDC is used for identifying the interacting proteins in the abstracts. This is done by identifying the constituents in each sentence that express proteins interactions, using the five sequential processing steps described in Section III-A. Then, an initial network of PPI is built from the extracted protein interactions. Edges are weighted by combining the following two scores using their harmonic mean.
   a) The first score is the number of interactions of each protein.
   b) The second score measures the importance of the protein in maintaining its neighborhood proteins connected. We do this because the importance of a protein in maintaining its neighbors connected reflects its biological relevance for particular molecular behaviors.

   Only proteins whose scores exceed a heuristically determined threshold are kept in the network. Maximum entropy classifier is used to make a judgment as to whether each protein pair has interaction relationship.
3) *Building disease-specific gene interaction network for detecting gene–disease associations:* In this case, T_i represents a target disease and R represents the set of genes that is semantically related to T_i. First, EDC_EDC builds a disease-specific gene interaction network as follows. We extract a set of seed genes known to be associated with the disease T_i from a database such as [31]. From this set, we select the subset S_g that is associated with at least one PubMed [2] abstract according to UniProtKB [32] entries. We then retrieve the abstracts of PubMed associated with the set S_g. From these abstracts, EDC_EDC identifies the genes that are semantically related to T_i and/or to each gene $\in S_g$ using the five processing steps described in Section III-A. It would use these genes to construct a disease-specific gene interaction network. It predicts the gene–disease associations by using centrality measures to locate the genes that are central in the network. This is based on the hypnosis that a disease is usually caused by functionally related genes and these genes are usually proximal to the center of the disease-specific gene interaction network.

IV. Determining Whether Two Nodes in POST Are Semantically Related

We observe that in order for the two nodes *A* and *B* in POST to be semantically related, the types of the nodes located in the

path from *A* to *B*, inclusive, in POST should be unique. Toward this, we introduce Proposition 1 next.

Proposition 1: In order for the two nodes t and t' in POST to be semantically related, the types of the nodes located in the path from t to t', inclusive, in POST should be unique (i.e., there should not be two or more nodes in the path from t to t' that have the same type, including t and t').

Let TYP_t denote the type of node *t*. We validate proposition 1 heuristically as follows:

Validation 1: We now validate: if the two nodes t and t' are semantically related, then $TYP_t \neq TYP_{t'}$. We are going to validate this rule by checking whether it conforms to the structural characteristics of existence dependency, as follows:

1) An object x is existence-dependent on an object y if the existence of x is dependent on the existence of y [33]. An arc in POST represents *part-of* relation between two nodes, and this *part-of* relation represents existence dependency between the two nodes, because: "*part-of has a specific meaning in a graph and a part of relation would only be added between nodes A and B if B is necessarily part of A: wherever B exists, it is as part of A, and the presence of B implies the presence of A*" [27]. "*part-of relation embodies some aspects of existence dependency. A part-of relation with existence dependent parts can simply be replaced by existence dependency: in case of existence dependent components, the existence dependency relation is identical to part-of relation*" [34], [35], [36], [37].
2) Snoeck and Dedene [34] argue that the existence dependency relation is a partial ordering of object types (*e.g., nodes' types*). The authors transform an OO schema into a graph consisting of the *object types* found in the schema and their relations. The object types in the graph are related only through associations that express existence dependency. The authors demonstrated through the graph that an object type is never existence- dependent on itself or on another object of the same type. A node in POST corresponds to an object in a graph, and the type of the POST's node corresponds to the type of the graph's object for the following reason. In both POST and graph, a type represents a class of characteristics: the type of a graph's object represents a class of characteristics shared by some of the graph's objects [34] and the type of a POST's node represents a class of lexical characteristic shared by some of the nodes. Objects in the graph that have the same class of characteristics are said to have the same type, and nodes in POST that have the same class of lexical characteristics are also said to have the same type.

Thus, a node in POST is never existence dependent on another node of the same type.

Validation 2: We validate: if the two nodes t and t' in POST are semantically related, then $TYP_{t''} \neq TYP_{t'''}$ where t'' and t''' are any two nodes in POST located in the path from t to t'. Intuitively, in order for t and t' to be semantically related, t'' and t''' should also be semantically related, because they relate (connect) between t and t' (*i.e., they pass the semantic contribution of* t to t'). In order for t'' and t''' to be semantically related, $TYP_{t''} \neq TYP_{t'''}$ (recall validation 1).

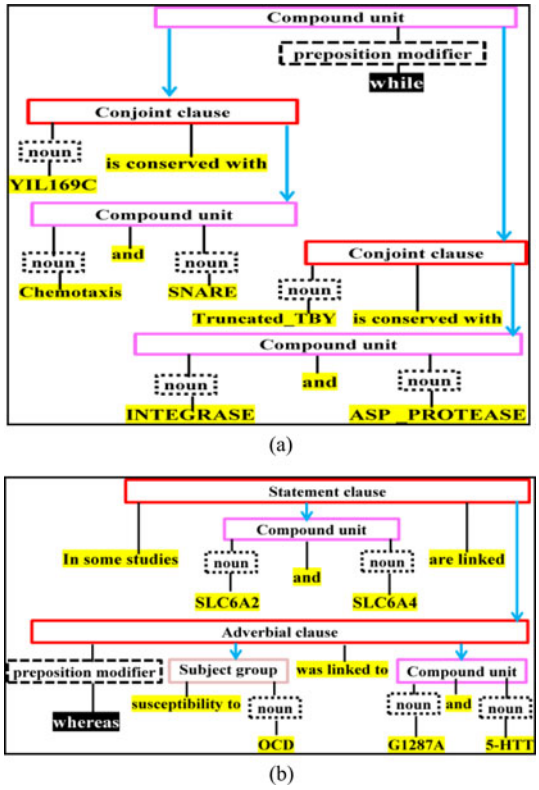

Fig. 2. POSTs for the sentences presented in the case studies of Section V-A.

V. Case Studies

We present in this section six case studies to illustrate our approach and how two terms/nouns in a sentence are determined to be semantically related using our proposed semantic rules described in Section IV. In each case study, we present the following: 1) a sentence extracted from biomedical literature; 2) a POST representing the syntactic structure of the sentence; and 3) determining whether each two nouns in the sentence are semantically related by applying our proposed semantic rules described in Section IV.

Based on the syntactic structures of sentences, we classify sentences into three groups, as follows: 1) sentences containing preposition modifiers; 2) sentences containing pronouns defining antecedents; and 3) sentences containing grammatically coordinate nouns and clauses connected by conjunctions. Each of the three groups has well-known linguistics theories that govern the syntactic structures of the sentences that fall under the group. In each of Section V-A, -B, and -C, we validate the accuracy of the semantic rules adopted by EDC_EDC by matching them with the linguistics theories of one of the three groups. That is, in each section, we demonstrate how our semantic rules conform to the linguistics theories that govern the syntactic structures of the sentences that fall under one of the three sentence groups.

Figs. 2–4 show the POSTs for the sentences presented in the six case studies. The following are descriptions of the figures. A blue arc represents a "part-of" relation between two nodes. A black line connects a node with a noun, verb, pronoun, modifier, etc. A dotted rectangle represents a noun, verb, pronoun, preposition, or conjunction. Nodes that have the same type are

colored with the same color for easy reference, as follows: red color denotes Clause Type (CLT), asparagus color denotes Conjunct Type (CONT), charm pink color denotes Compound Type (CMPDT), blue color denotes Object Type (OBJT), baby pink color denotes Subject Type (SUBT), and black color denotes Adverbial Type (ADVT). Some constituents are merged for the sake of figure clarity.

A. Sentences Containing Preposition Modifiers

In linguistics, the two parts of a sentence connected by a preposition modifier (such as "while," "but," and "whereas") are usually unrelated. Let n_1 be a noun that belongs to one of the two parts of a sentence connected by a preposition modifier and let n_2 be a noun that belongs to the second part of the sentence. In case studies 1 and 2, we demonstrate how the semantic rules adopted by EDC_EDC conform to the linguistics theory stated earlier by determining that n_1 and n_2 are semantically unrelated.

Case study 1: Consider the POST in Fig. 2(a) and the following sentence: "YIL169C is conserved with Chemotaxis and SNARE while Truncated_TBY is conserved with INTEGRASE and ASP_PROTEASE."

1) Each of the nouns "YIL169C," "Chemotaxis," and "SNARE" is unrelated to each of the nouns "Truncated_TBY," "INTEGRASE," and "ASP_PROTEASE", because each path in POST between one of the nouns in the first set and another noun in the second set includes more than one nodes that have the same type (i.e., the nodes Conjoint Clauses and/or Compound Units). In linguistic, the two sets of nouns are also considered unrelated because they are connected by the preposition modifier "while."
2) The noun "YIL169C" is related to the nouns "Chemotaxis" and "SNARE."
3) The noun "Truncated_TBY" is related to the nouns "INTEGRASE" and "ASP_PROTEASE."

Case study 2: Consider Fig. 2(b) and the sentence: "In some studies, SLC6A2 and SLC6A4 are linked, whereas susceptibility to OCD was linked to G1287A and 5-HIT."

1) Each of the nouns "SLC6A2" and "SLC6A4" is unrelated to each of the nouns "OCD," "G1287A," and "5-HIT," because each path in POST between one of the nouns in the first set and another noun in the second set includes more than one nodes that have the same type (i.e., the nodes "Statement Clause," "Adverbial Clause," and/or "Compound Unit"). In linguistics, the two sets of nouns are also considered unrelated because they are connected by the preposition modifier "whereas."
2) The nouns "SLC6A2" and "SLC6A4" are related.
3) The nouns "OCD," "G1287A," and "5-HIT" are related.

B. Sentences Containing Pronouns Defining Antecedents

In linguistics, an antecedent noun connected with a subsequent noun by a pronoun (such as "which," "it," "who," "that," and "whom") are usually related. In case studies 1 and 2, we demonstrate how the semantic rules adopted by EDC_EDC conform to the linguistics theory stated earlier by determining that an antecedent noun connected with a subsequent noun by a pronoun is semantically related. EDC_EDC replaces each pronoun under a constituent node with the noun found under the closest predecessor constituent's node. For example, pronoun "which" in Fig. 3(A) is replaced with the noun "p16," which belongs to the closest predecessor constituent's node. This conforms to linguistics and grammar, which state that a pronoun is a word that substitutes for a noun or noun phrase. In Fig. 3, the texts inside the dotted ovals are antecedents, which replace pronouns.

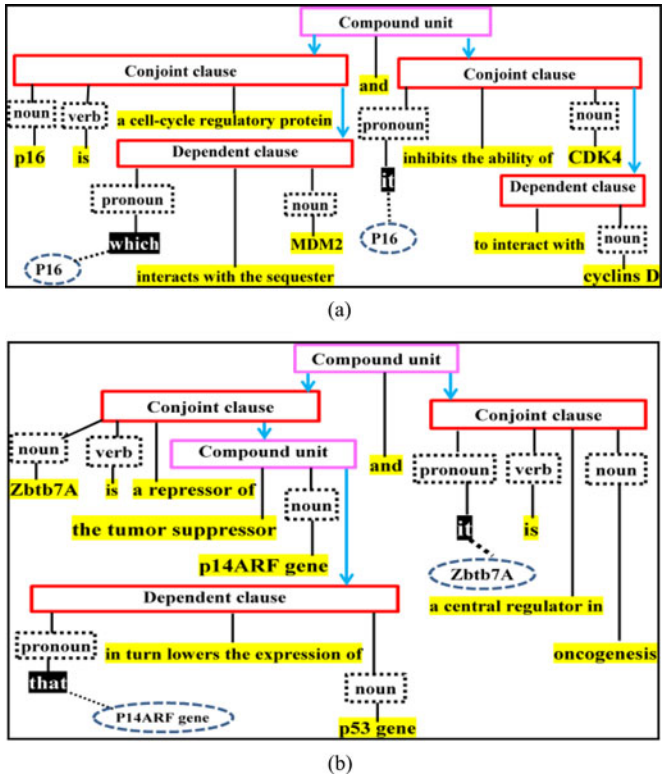

Fig. 3. POSTs for the sentences presented in the case studies of Section V-B. The text inside a dotted oval under a pronoun is the noun (i.e., the antecedent) found under the closest predecessor node, which replaces the pronoun.

Case study 1: Consider the POST in Fig. 3(a) and the sentence: "p16 is a cell-cycle regulatory protein, which interacts with the sequester MDM2, and it inhibits the ability of CDK4 to interact with cyclins D."

1) The noun "p16" is related to the noun "CDK4." This is because the pronoun "it" under the "Conjoint Clause" is replaced with the noun under the closest predecessor node, which is "p16."
2) The noun "p16" is related to the noun "MDM2." This is because the pronoun "which" under the "Dependent Clause" is replaced with the noun in the closest predecessor node, which is "p16."
3) The noun "MDM2" is unrelated to the nouns "CDK4" and "cyclins D," because the paths in POST between "MDM2" and the other nouns include more than one node that have the same type.

Case study 2: Consider the POST in Fig. 3(b) and the following sentence: "Zbtb7A is a repressor of the tumor suppressor p14ARF gene that in turn lowers the expression of p53 gene and it is a central regulator in oncogenesis."

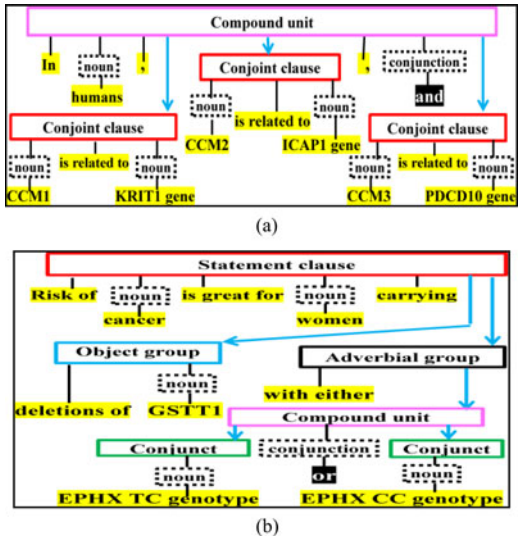

Fig. 4. POSTs for the sentences presented in the case studies of Section V-C.

1) The noun "Zbtb7A" is related to the noun "oncogenesis." This is because pronoun "it" under "Conjoint Clause" is replaced with the noun in the closest predecessor node, which is "Zbtb7A."
2) The noun "p14ARF gene" is related to the noun "p53 gene." This is because the pronoun "that" under the "Dependent Clause" is replaced with the noun in the closest predecessor node, which is "p14ARF gene."
3) The noun "Zbtb7A" is related to the noun "p14ARF gene."
4) The noun "Zbtb7A" is unrelated to the noun "p53 gene," because the path in POST between the two nouns includes more than one node that have the same type (i.e., the nodes "Conjoint Clause" and "Dependent Clause").
5) The noun "p53 gene" is unrelated to the noun "oncogenesis," because the path in POST between the two nouns includes more than one node that have the same type (i.e., the nodes "Conjoint Clause" and "Dependent Clause").

C. Sentences Containing Coordinate Nouns or Clauses Connected by Conjunctions

In linguistics, coordinate nouns/clauses connected by a conjunction (such as "and" and "or") are related to the *context* of the sentence, but these nouns/clauses themselves may not be related. For example, in the sentence "Jon likes travel and sports," the words "travel" and "sports" are related to the context of the sentence, which is "Jon," and, the two words themselves are unrelated. In case studies 1 and 2, we demonstrate how the semantic rules adopted by EDC_EDC conform to the linguistics theories stated earlier by determining that the nouns/clauses connected by conjunctions are semantically unrelated and that they are semantically related to the context of the sentence.

Case study 1: Consider Fig. 4(a) and the following sentence: "In humans, CCM1 is related to KRIT1 gene, CCM2 is related to ICAP1 gene, and CCM3 is related to PDCD10 gene."

1) Each of the pair of nouns ("CCM1," "KRIT1 gene"), ("CCM2," "ICAP1 gene"), and ("CCM3," "PDCD10 gene") is unrelated to the other pairs, because the paths in POST between them include more than one node that have the same type (i.e., the "Conjoint Clause" nodes). However, the three pairs are related to the noun "humans." Linguistics considers also the three pairs as unrelated, because they are connected by the conjunction "and," and also considers that the three pairs as related to the context of the sentence, which is the noun "humans."

Case study 2: Consider the POST in Fig. 4(b) and the following sentence: "Risk of cancer is great for women carrying deletions of *GSTT1* with either *EPHX TC* genotype or *EPHX CC* genotype."

1) The two nouns "EPHX TC genotype" and "EPHX CC genotype" are unrelated, because the path in POST between them includes more than one node that have the same type. However, each of the two nouns is related to the nouns "cancer" and "women." Linguistics considers also the two nouns "EPHX TC genotype" and "EPHX CC" as unrelated, because they are connected by the conjunction "or," and also considers the two nouns as related to the contexts of the sentence, which are the nouns "cancer" and "women."

VI. Experimental Results

We implemented EDC_EDC in Java, run on Intel(R) Core(TM) i5-4200U processor, with a CPU of 2.30 GHz and 4 GB of RAM, under Windows 8. A demo of EDC_EDC that identifies the Biological Process annotations of the complete Yeast protein dataset downloaded from [38] is available at: http://ecesrvr.kustar.ac.ae:8080/ (click on "Gene Ontology 1").

A. Description of the Systems to be Compared With EDC_EDC

We experimentally evaluated the quality of EDC_EDC for determining the functions of proteins, detecting PPI, and inferring gene–disease associations by comparing it experimentally with Text-KNN [9], Prodisen [12], LEAP-FS [13], PRED [11], ACT [16], and CGDA [20]. The following are brief descriptions of the six systems.

1) *Text-KNN* [9]*:* It determines the functions of unannotated proteins by extracting information from the abstracts of biomedical literature. It represents each protein with characteristic terms. A term is considered a characteristic if it has high occurrence probability in abstracts. It then assigns an unannotated protein the functions of annotated proteins that have similar characteristic terms using a *k*-nearest neighbor classifier.
2) *LEAP-FS* [13]*:* It predicts protein function from the protein sites mentioned in biomedical abstracts. It categorizes protein sites based on their protein structures determined by the amino acid residues found in biomedical abstracts. To identify functionally important residues, it checks if they match physical residues in PDB entries. It then assigns a protein *p* the functions of proteins that fall under the same functional sites as *p*.
3) *Prodisen* [12]*:* It predicts protein function from the functional descriptions of proteins found in biomedical literature. It categorizes the functional description of genes

found in the texts. It assigns unannotated protein the functions of annotated proteins that fall under the same textual classification.

4) *PRED* [11]*:* It predicts protein function from the sentential structures of biomedical literature's sentences. It represents the dependency structure of a sentence as a graph. It determines the relationships between nodes representing proteins and functions using the shortest path algorithm.
5) *ACT* [16]*:* It predicts protein function from the syntactic features in biomedical texts. It uses a dependency parser to identify the relationships between words and to extract protein names from the grammar relations found in the texts. It uses machine learning to classify PPI and protein function articles. It uses the multiword features known as *n*-grams for determining *n*-consecutive words that reflect PPI and protein functions.
6) *CGDA* [20]*:* It predicts gene–disease associations from biomedical literature. It constructs a disease-specific gene interaction network to predict gene–disease associations. It constructs the network from genes known to be related to the disease from biomedical literature. It uses centrality measures to predict gene–disease associations.

B. Evaluating the Performance of the Systems on Predicting Protein Function

1) Evaluating the Prediction Performance Using GO Dataset: We selected fragments of GO graph from the biological process and molecular function subontologies as sources for the evaluation dataset [27]. The fragment from the biological process subontology contains 70 GO terms annotating 583 846 proteins. The fragment from the molecular function subontology contains 30 GO terms annotating 603 438 proteins. From the proteins annotated with the functions of the 100 GO terms, we selected only the ones that: 1) are associated with at least one PubMed abstract according to their entries in UniProtKB [32]; and 2) have experimental evidence code: IC, IDA, EXP, IEP, TAS, IPI, IGI, IMP, or IC. The number of proteins that satisfy these two conditions is 78 908 (62 353 proteins annotated with the biological process subontology and 16 555 proteins annotated with the molecular function subontology). We downloaded the 100 GO terms and the 78 908 proteins annotated with their functions from [27]. We retrieved 576 028 PubMed abstracts associated with the selected 78 908 proteins according to their entries in the UniProtKB/Swiss-Prot database [32].

We performed tests resemble fivefold cross validation using the 78 908 proteins and the 576 028 PubMed abstracts associated with them. The protein dataset and the set of abstracts associated with them are partitioned (at random) into five disjoint subsets. That is, each disjoint subset contains 15 781 proteins and the subset of the 576 028 PubMed abstracts associated with them according to their entries in the UniProtKB/Swiss-Prot database [32]. EDC_EDC and the first five systems described in Section VI-A are evaluated five times, where at each time the proteins in one of the subsets are considered as unannotated and used for the testing while the *abstracts* associated with the remaining four subsets are used for training the systems. These training

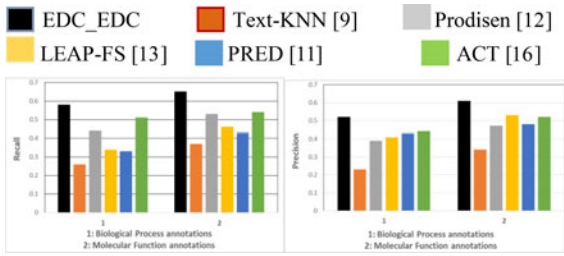

Fig. 5. Overall average Recall and Precision of the systems for predicting protein functions using the GO dataset described in Section VI-B1.

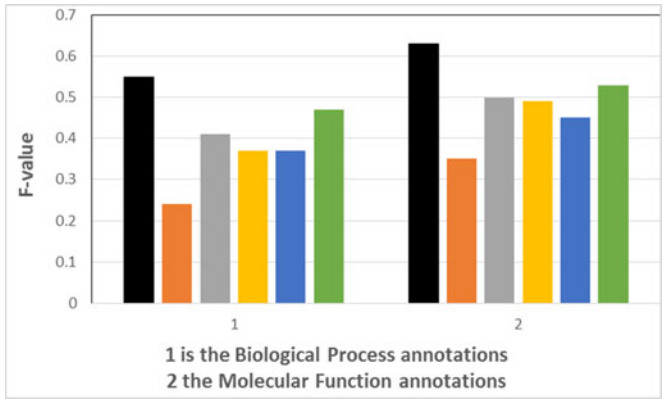

Fig. 6. Overall average *F*-value of each system for predicting protein functions using the GO dataset described in Section VI-B1.

abstracts will be used by: 1) EDC-EDC for determining functional categories; 2) Text-KNN for determining characteristic terms; 3) LEAP-FS for categorizing protein sites based on the amino acid residues mentioned in the abstracts; 4) Prodisen for categorizing the functional description of the genes found in the abstracts; 5) ACT for training its machine learning; and 6) PRED for constructing graphs based on the sentential structures of the sentences in the abstracts.

We measured the Recall, Precision, and *F*-value of the results returned by the six systems. We measured the prediction performance for an unannotated protein *P* using the standard metrics of *Recall*, *Precision*, and *F-value* as follows:

$$\text{Recall} = \frac{c_p}{n_p}, \text{Precision} = \frac{c_p}{m_p}$$

$$\text{and } F - \text{value} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

where c_p is the number of *correctly* predicted functions, n_p is the number of actual functions, and m_p is the number of predicted function. The results are shown in Figs. 5 and 6.

2) Evaluating the Prediction Performance Using the Complete Set of Yeast Dataset: We evaluated the systems using the complete 6086 Yeast protein dataset downloaded from [38]. We retrieved 47 652 PubMed abstracts associated with the 6086 proteins according to their entries in the UniProtKB/Swiss-Prot database [32], using the same procedure described in Section VI-A. We performed fivefold cross validation using the same

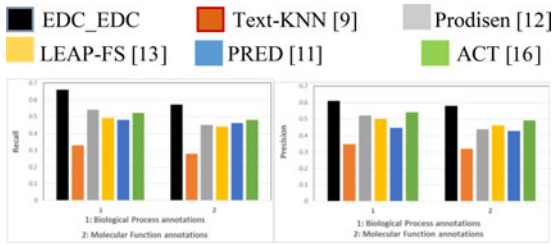

Fig. 7. Overall average Recall and Precision of the systems for predicting protein functions using the Yeast protein dataset described in Section VI-B2.

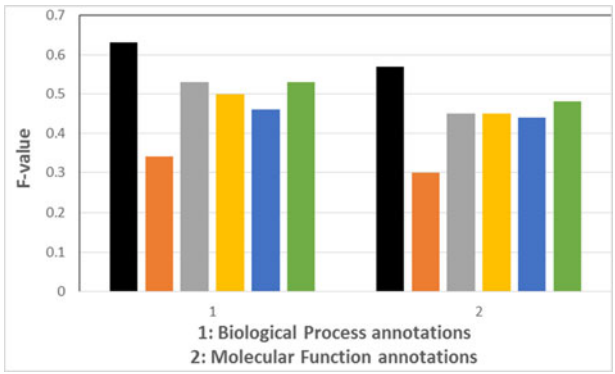

Fig. 8. Overall average *F*-value of the systems for predicting protein functions using the Yeast dataset described in Section VI-B2.

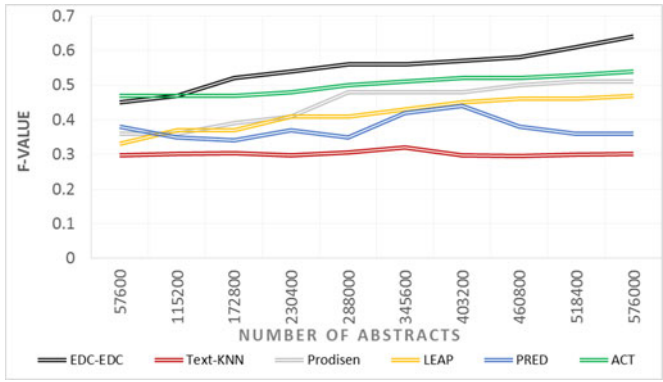

Fig. 9. Overall average *F*-value of the systems for predicting protein functions using the cumulative validation dataset described in Section VI-B3.

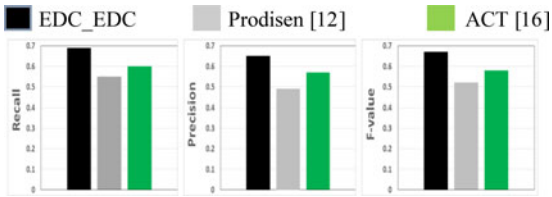

Fig. 10. Overall average Recall, Precision, and *F*-value for detecting PPI using IEPA corpus described in Section VI-C.

procedure described in Section VI-B. Figs. 7 and 8 show the results.

3) Evaluating the Prediction Performance Using Cumulative-Validation Dataset: We performed ten evaluation runs over a set of training abstracts that accumulates in each run successively. We randomly partitioned the 78 908 proteins and the 576 028 abstracts into ten subsets. In the first run, the proteins in one of the subsets are considered unannotated and used for testing while the abstracts in this subset are used for training the systems. Thereafter, and after each run, the proteins in a different subset are considered unannotated and used for testing. Also, the previous set of abstracts is added to the current set, and the accumulating set is used for training. Fig. 9 shows the *F*-value of the results.

C. Evaluating the Performance of the Systems on Detecting PPI

In this section, we evaluate the performance of EDC_EDC for detecting PPI by comparing it with Prodisen [12] and ACT [16], which are the only ones in the list described in Section VI-A that can detect PPI besides predicting protein function. The corpus we used for this evaluation is the interaction extraction performance assessment (IEPA) [39]. The corpus consists of 303 abstracts downloaded from PubMed. It includes 487 sentences; each contains at least one pair of proteins. The corpus contains 644 PPI pairs, including 336 positive pairs. We divided *each* of the positive and the negative PPI pairs into two equal parts. We considered one part as a test set and the other as the training set. We submitted the test set to the three systems and measured the Recall, Precision, and *F*-value. Fig. 10 shows the results.

D. Evaluating the Performance of the Systems on Detecting Gene–Disease Associations

We evaluated the performance of EDC_EDC for detecting gene–disease associations by comparing it with CGDA [20]. We used the same dataset and experimental setting used for evaluating CGDA in [20]. The evaluation was done for prostate cancer. The initial seed genes consist of 15 prostate cancer genes, which are extracted from the Morbid Map component of the OMIM database. The two systems used the list of seed genes to construct prostate cancer-specific gene-interaction network mined automatically from 48 245 articles from PubMed Central (PMC) Open Access. We used the Prostate Gene DataBase (PGDB) [40] for the evaluation of the results returned by the two systems. PGDB is a curated database of genes related to prostate cancer. We ranked result genes using the same ranking metrics used for evaluating CGDA in [20], which are degree centrality, betweenness, and closeness centrality. Table II shows the precisions of the two systems for the top-ranked *n* genes. That is, the percentage of the top-ranked "*n*" genes that are marked by PGDB as being related to prostate cancer.

E. Discussion of the Results

As Figs. 5–10 and Table II show, EDC_EDC outperformed the other systems. The results reveal the robustness of the EDC_EDC's method and its ability to reflect the *semantic relationships* between different terms within the sentences of

TABLE II
PERCENTAGE OF THE TOP-RANKED n GENES THAT ARE MARKED BY PGDB AS BEING RELATED TO PROSTATE CANCER

Top n	Degree		Betweenness		Closeness	
	EDC_EDC	CGDA	EDC_EDC	CGDA	EDC_EDC	CGDA
10	82.8	80	86.4	90	77.3	70
20	78.1	75	81.9	70	65.7	55
30	69	60	73.5	63.3	58.4	56.7
40	63.6	55	62	52.5	54.8	47.5
50	50.2	46	52.3	48	47	42
75	36	33.3	41.7	34.7	35.5	33.3
100	31.9	26.	33	26	33.2	27.
125	29.4	23	27.1	23.2	27	23.3
150	25	20.7	25.3	20	24.8	20
175	23.6	18.3	24	18.3	20.9	18.3
200	20.5	17.5	19	18.5	17.9	17
226	20.3	17.7	17.8	17.7	17.4	17.7

Ranking is done using the Degree, Betweenness, and Closeness measures.

TABLE III
CLASSIFICATION OF POST UNDER THREE CRITERIA BASED ON THE SYNTACTIC STRUCTURE OF TARGET AND REFERENCE TERMS WITHIN A SENTENCE

Criterion #	POST Criterion
1	Nodes in POST containing the terms describing the functions of target and reference proteins in a sentence, respectively, are in shallow hierarchical levels in POST.
2	Nodes in POST containing the terms describing the functions of target and reference proteins in a sentence, respectively, are in deep hierarchical levels in POST.
3	Some of the nodes in POST containing the terms describing the functions of target and reference proteins in a sentence are in shallow levels while the others are in deep levels in POST.

biomedical literature. In the following sections, we discuss the results of the experimental evaluations.

1) Predicting the Functions of Proteins: As can be seen from Figs. 5–9, EDC_EDC outperformed the other five systems in terms of predicting protein function. And, as can be seen from the results of the cumulative-validation dataset shown in Fig. 9, as the size of training abstracts gets larger, EDC_EDC, LEAP-FS, Prodisen, and ACT tend to predict the functions of unannotated proteins more accurately. For EDC_EDC, the reason of the improvement is that every time the size of abstracts increases, EDC_EDC updates and optimizes its current set of functional categories. Also, as the size of abstracts gets larger, LEAP-FS, Prodisen, and ACT tend to categorize the functional descriptions of proteins more accurately. This is advantageous to the four systems, because in real world the size of abstracts associated with annotated proteins keeps increasing. As for Text-KNN and PRED, the accumulation of training abstracts has no noticeable impact on their prediction accuracy.

We also studied the prediction performance of the systems under *each of the three criteria* shown in Table III. We analyzed the behavior of the systems in terms of their *recall* and *precision* under each of the three criteria. The results are shown in Table IV. EDC_EDC outperformed the three systems under criteria 1 and 2, which is due to EDC_EDC's consideration to the *structural dependences* and *semantic relationships* between each two constituents in a sentence. Prodisen outperforms EDC_EDC under criterion 3. This is because, in complex sentences that include many pronouns, EDC_EDC may not be able to identify *all* the semantic relationships between terms separated by these pronouns.

TABLE IV
AVERAGE RECALL AND PRECISION UNDER THE THREE CRITERIA OF POST SHOWN IN TABLE III

Criterion number	EDC_EDC		Text-KNN		Prodisen		LEAP-FS	
	R	P	R	P	R	P	R	P
1	0.68	0.66	0.34	0.27	0.46	0.40	0.40	0.41
2	0.65	0.61	0.28	0.38	0.44	0.47	0.49	0.47
3	0.52	0.47	0.32	0.31	0.56	0.49	0.45	0.45

"*R*" denotes Recall and "*P*" denotes Precision.

2) Detecting PPI: As Fig. 10 shows, EDC_EDC outperforms Prodisen and ACT on detecting PPI, which is attributed to the fact that EDC_EDC considers the *structural dependences* and *semantic relationships* between constituents that express proteins interactions in sentences, while Prodisen and ACT do not. Consider the structural dependences help in identifying more accurate neighbors for proteins in networks and for calculating more accurate harmonic means. Also, Prodisen and ACT are not well suited for identifying complex phrases that express proteins interactions in sentences.

3) Detecting Gene–Disease Associations: As can be seen from Table II, ECD_EDC outperformed CGDA. This is attributed, mainly, to the fact that EDC_EDC considers the structural dependences not only between constituents that express gene–disease associations in sentences, but also among all genes. By not considering the structural dependences among constituents, CGDA may falsely return positive association between a gene and a disease. Also, when using the centrality measures to determine the genes that are central for a disease-specific genetic network, EDC_EDC considers the interactions among all genes. As for CGDA, it considers only the interactions between seed genes and between seed genes and their neighbors. It overlooks the interactions between the neighbors. As a result, CGDA favors more seed genes.

VII. CONCLUSION

We proposed in this paper a hybrid constituency–dependency parser for biological NLP information extraction called EDC_EDC. It aims at enhancing the state of the art of biological text mining by applying novel linguistic computational techniques that overcome the limitations of current constituency and dependency parsers, as follows: 1) it determines the *semantic relationship* between each pair of constituents in a sentence using novel semantic rules (*as opposed to most current constituency parsers, which determine the structural relationships between individual words*); and 2) it applies *semantic relationship extraction models* that extract information from different structural forms of constituents in sentences. The proposed approach disambiguates by comparing the likelihoods of plausible relationship between each pair of constituents in a sentence based on similarity to formal sematic rules. We experimentally evaluated the quality of EDC_EDC and compared it with six

systems. The results showed that EDC_EDC outperforms the other systems in predicting protein function, detecting PPI, and inferring gene–disease association.

REFERENCES

[1] K. B. Cohen and L. Hunter, "Natural language processing and systems biology, "in *Artificial Intelligence Methods and Tools for Systems Biology*, W. Dubitzky and F. Azuaje, Eds. Dordrecht, The Netherlands: Kluwer, 2004.

[2] PubMed. (2014). [Online]. Available at: http://www.ncbi.nlm.nih.gov/pubmed

[3] E. M. Marcotte, I. Xenarios, and D. Eisenberg, "Mining literature for protein-protein interactions," *Bioinformatics*, vol. 17, pp. 359—363, 2001.

[4] F. Ginter, T. Pahikkala, S. Pyysalo, J. Boberg, J. Jarvinen, and T. Salakoski, "Extracting protein—protein interaction sentences by applying rough set data analysis," presented at the 4th International Conference on Rough Sets and Current Trends in Computing, 2004, pp. 780–785.

[5] S. Tsumoto, R. Slowinski, J. Komorowski, and J. Grzymala-Busse, "Evaluation of two dependency parsers on biomedical corpus targeted at protein–protein interactions," in *Lecture Notes in Artificial Intelligence*. New York, NY, USA: Springer, 2004.

[6] A. Yakushiji, Y. Tateisi, Y. Miyao, and J. Tsujii, "Event extraction from biomedical papers using a full parser," in *Proc. Pacific Symp. Biocomput.*, 2001, vol. 6, pp. 408–419.

[7] D. Sleator and D. Temperley, "Parsing english with a link grammar," Dept. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-91-196, 1991.

[8] M. Craven and J. Kumlien, "Constructing biological knowledge bases by extracting information from text sources," in *Proc. 7th Int. Conf. Intell. Syst. Mol. Biol.*, 1999, pp. 77–86.

[9] A. Wong and H. Shatkay, "Protein function prediction using text-based features extracted from the biomedical literature," *BMC Bioinformat.*, vol. 14, no. Suppl 3, p. S14, 2013.

[10] P. Groth, B. Weiss, H. D. Pohlenz, and U. Leser, "Mining phenotypes for gene function prediction," *BMC Bioinformatics*, vol. 9, p. 136, 2008.

[11] S. Kim, J. Yoon, and J. Yang, "Kernel approaches for genic interaction extraction," *Bioinformatics*, vol. 24, pp. 118–126, 2008.

[12] M. Krallinger, R. Malik, and A. Valencia, "Text mining and protein annotations: The construction and use of protein description sentences," *Genome Informat.*, vol. 17, no. 2, pp. 121–130, 2006.

[13] M. Verspoor, D. Cohn, E. Ravikumar, and E. Wall, "Text mining improves prediction of protein functional sites," *PLoS One*, vol. 7, no. 2, p. e32171, 2012.

[14] C. Xiaoxiao and H. Jingyu, "An iterative approach of protein function prediction," *BMC Bioinformat.*, vol. 12, no. 1, p. 437, 2011.

[15] M. Andrade and A. Valencia, "Automatic extraction of keywords from scientific text: Application to the knowledge domain of protein families," *Bioinformatics*, vol. 14, no. 7, pp. 600–607, 1998.

[16] S. Kim and W. John Wilbur, "Classifying protein-protein interaction articles using word and syntactic features," *BMC Bioinformat.*, vol. 12, no. 8, p. 59, 2010.

[17] J. Xiao, J. Su, G. Zhou, and C. Tan, "Protein-protein interaction: A supervised learning approach," presented at the International Symposium on Semantic Mining in Biomedicine, Hinxton, U.K., 2005.

[18] E. H. Davidson, J. P. Rast, P. Oliveri, A. Ransick, C. Calestani, D. R. McClay, L. Hood, and H. Bolouri, "A genomic regulatory network for development," *Science*, vol. 295, no. 5560, pp. 1669–1678, 2002.

[19] N. Domedel-Puig and L. Wernisch, "Applying GIFT, a gene interactions finder in text, to fly literature," *Bioinformatics*, vol. 21, no. 17, pp. 3582–3583, 2005.

[20] A. Ozgür, T. Vu, G. Erkan, and D. R. Radev, "Identifying gene-disease associations using centrality on a literature mined gene-interaction network," *Bioinformatics*, vol. 24, no. 13, pp. i277–i285, 2008.

[21] P. Tapanainen and T. Jarvinen, "A non-projective dependency parser," presented at the 5th Conference on Applied Natural Language Processing, Association for Computational Linguistics, Somerset, NJ, USA, 1997.

[22] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2009.

[23] Y. Tsuruoka and J. Tsujii, "Boosting precision and recall of dictionary-based protein name recognition," in *Proc. Workshop Natural Language Process. Biomed.*, 2003, pp. 41–48.

[24] N. Ge, J. Hale, and E. Charniak, "A statistical approach to anaphora resolution," in *Proc. Workshop Very Large Corpora*, Hong Kong, 1998.

[25] D. Gildea, and D. Jurafsky, "Automatic labeling of semantic roles," *Comput. Linguistics*, vol. 28, no. 3, pp. 245–288, 2002.

[26] D. Blaheta and E. Charniak, "Assigning function tags to parsed text, "in *Proc. 1st Annu. Meeting North Amer. Chapter Assoc. Comput. Linguistics*, 2000, pp. 234–240.

[27] Gene Ontology (GO). (2014). [Online]. Available: http://www.geneontology.org/

[28] Stanford Tokenizer, Part-of-Speech Tagger, and Named Entity Recognizer. (2014). [Online]. Available: http://nlp.stanford.edu/software

[29] K. Taha, P. Yoo, and M. Al Zaabi, "iPFPi: A system for improving protein function prediction through cumulative iterations," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, 2014, DOI: 10.1109/TCBB.2014.2344681

[30] I. Xenarios, L. Salwinski, X. J. Duan, P. Higney, S-M. Kim, and D. Eisenberg, "DIP, the database of interacting proteins: A research tool for studying cellular networks of protein interactions," *Nucl. Acids Res.*, vol. 30, pp. 303–305, 2002.

[31] CBioC: Collaborative Bio Curation. (2014). [Online]. Available at: http://cbioc.eas.asu.edu/

[32] A. Bairoch, R. Apweiler, C. H. Wu, W. C. Barker, N. Redaschi, and L. Yeh, "The universal protein resource (UniProt)," *Nucleic Acids Res.*, vol. 33, no. 1, pp. 154–159, 2005.

[33] N. Widjaya, D. Taniar, and W. Rahayu, "Aggregation transformation of XML schemas to object-relational databases," in *Proc. Int. Conf. Innovative Internet Community Syst.*, 2003, pp. 251–262.

[34] M. Snoeck and G. Dedene, "Existence dependency: The key to semantic integrity between structural and behavioral aspects of object types," *IEEE Trans. Softw. Eng.*, vol. 24, no. 24, pp. 233–251, Apr. 1998.

[35] K. Taha, "Determining semantically related significant genes," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 11, no. 6, pp. 1119–1130, Nov./Dec. 2014.

[36] K. Taha, "GRtoGR: A system for mapping GO relations to gene relations," *IEEE Trans. NanoBiosci.*, vol. 12, no. 4, pp. 289–297, Aug. 2013.

[37] K. Taha, "Determining the semantic similarities among gene ontology terms," *IEEE J. Biomed. Health Informat.*, vol. 17, no. 3, pp. 512–525, Feb. 2013.

[38] SGD (Saccharomyces Genome Database). (2014). [Online]. Available at: http://www.yeastgenome.org/download-data/curation

[39] J. Ding, D. Berleant, D. Nettleton, and E. Wurtele, "Mining MEDLINE: Abstracts, sentences, or phrases," in *Proc. Pacific Symp. Biocomput.*, Hawaii, USA, Jan. 2002, pp. 326–337.

[40] L. Li, H. Zhao, H. Shiina, C. J. Kane, and R. Dahiya, "PGDB: A curated and integrated database of genes related to the prostate," *Nucleic Acids Res.*, vol. 31, pp. 291–293, 2003.

Kamal Taha (SM'14) received the Ph.D. degree in computer science from the University of Texas at Arlington, Arlington, TX, USA, in March 2010.

He has been an Assistant Professor in the Department of Electrical and Computer Engineering, Khalifa University, Abu Dhabi, United Arab Emirates, since 2010. He has more than 50 refereed publications that have appeared in prestigious top-ranked journals, conference proceedings, and book chapters. Ten of his publications have appeared in IEEE Transactions journals. He was an Instructor of Computer Science at the University of Texas at Arlington, USA, from August 2008 to August 2010. He was an Engineering Specialist for Seagate Technology, USA, from 1996 to 2005 (*Seagate is a leading computer disc drive manufacturer in the US*). His current research interests include bioinformatics databases, information retrieval in semistructured data, keyword search in XML documents, recommendation systems and social networks, and data mining.

Dr. Taha serves as a member of the Program Committee, editorial board, and review panel for a number of international conference and journal publications such as IEEE and ACM. He was included in the 2012 Edition of Who's Who in Science and Engineering.