Categorial Proof Nets and Dependency Locality: A New Metric for Linguistic Complexity

Mehdi Mirzapour, Jean-Philippe Prost, Christian Retoré

Categorial Proof Nets and Dependency Locality: A New Metric for Linguistic Complexity

Mehdi Mirzapour, Jean-Philippe Prost, and Christian Retoré

LIRMM, Montpellier University CNRS, 161 Rue Ada, France,
email: {mehdi.mirzapour — jean-philippe.prost — christian.retore}@lirmm.fr

Abstract. This paper provides a quantitative computational account of why a sentence has harder parse than some other one, or that one analysis of a sentence is simpler than another one. We take for granted Gibson's results on human processing complexity, and we provide a new metric which uses (Lambek) Categorial Proof Nets. In particular, we correctly model Gibson's account in his Dependency Locality Theory. The proposed metric correctly predicts some performance phenomena such as structures with embedded pronouns, garden pathing, unacceptability of center embedding, preference for lower attachment and passive paraphrases acceptability. Our proposal extends existing distance-based proposals on Categorial Proof Nets for complexity measurement while it opens the door to include semantic complexity, because of the syntax-semantics interface in categorial grammars.

Keywords: Computational Linguistics · Psycholinguistics · Human Processing · Categorial Grammar · Linear Logic · Lambek Calculus

1 Introduction

Linguistics and especially generative grammar à la Chomsky makes a distinction between *competence* and *performance* in the human processing of natural language [5]. The competence is, roughly speaking, our ideal ability without time and resource constraints to parse a sentence, i.e. to decide that it is grammatical or not. Competence is formally described by a formal grammar. The performance is how we actually parse a sentence; whether we succeed in achieving that and how much the sentence resists to our attempt to analyze it. Computing the space and time algorithmic complexity is a fake solution because no one knows the algorithm being used by human if it depends on the individual and on the kind of conversation; even if it were so, nothing guarantees that space and time algorithmic complexity matches the degree of difficulty we experience when processing sentences. So this paper, as well as some earlier work by others[14, 21], try to provide a formal and computable account of the results of psycholinguistics experiences regarding linguistic complexity. We focus on syntactic complexity as studied in a number of linguistic processing phenomena such as garden paths, unacceptability of center embedding, preference for lower attachment, passive paraphrases acceptability, and structures with embedded pronouns.

Regarding the psycholinguistics aspects, we mainly follow the studies by Gibson of linguistic complexity of human parsing. Gibson first studied the notion of the linguistic difficulty [10] through the maximal number of incomplete syntactic dependencies that the processor has to keep track of during the course of processing a sentence. We refer to this theory as Incomplete Dependence Theory (IDT) as coined by Gibson. IDT had some limitations for referent-sensitive linguistic phenomena, which justified the later introduction of the Syntactic Prediction Locality Theory [8]. A variant of this theory, namely Dependency Locality Theory (DLT), was introduced later [9] to overcome the limitations with respect to the new linguistic performance phenomena. In the original works, both IDT and DLT use properties of linguistic representations provided in Government-Binding Theory [6].

On the formal side, in order to compute the complexity of a sentence — in a way that matches Gibson's results — we use Lambek Categorial Grammar [16] by means of proof nets construction [19, Chap 6]. Proof nets were originally introduced by Girard [12] as the mathematical structures of proof in linear logic. Categorial proof nets are to categorial grammar what parse trees are to phrase structure grammar. This kind of approach was initiated by Johnson [14], who defines a measure of the instantaneous complexity when moving from a word to the next one (in particular for center embedded relative clauses) in a way that matches Gibson's and Thomas' analysis [11]. To define the complexity of a sentence, Johnson considers the maximum complexity between the words in a given sentence. This approach was refined by Morrill [21], who re-interprets axiom links in categorial proof nets as incomplete (or unresolved) dependencies. We rename this technique as *IDT-based complexity profiling* since it clearly inherits many aspects of Gibson's IDT, plus the new notion of profiling that exists in some psycholinguistic theories. This technique is quite successful at predicting linguistic performance phenomena such as garden paths, unacceptability of center embedding, preference for lower attachment and heavy noun phrase shift. Nevertheless, there is some predictive limitation for referent-sensitive phenomena such as structures with embedded pronouns. Our strategy to overcome this issue is to apply DLT instead of IDT on proof nets constructions which would lead to introduction of *DLT-based complexity profiling*. We will show how this reformulation can improve the predictive power of the existing models in favor of the referent-sensitive linguistic phenomena.

The purpose of developing our computational psycholinguistic model is not solely limited to measuring linguistic complexity. It is potentially applicable to some specific tasks in the domain of the formal compositional semantics. For instance, ranking different possible readings of a given ambiguous utterance, or more generally translating natural language sentences into weighted logical formulas. The rest of the paper is organized as follows: Section 2 summarizes Gibson's ideas on modeling the linguistic complexity of human sentence comprehension, namely IDT and DLT. In section 3 we then define proof nets, and recall the success and limitation of IDT-based complexity profiling. In section 4 we define our DLT-inspired measure, we show how it fixes some problems in previous work and how it gives a correct account of those phenomena. In section 5, we would see a limitation of our approach and a possible future study for solving that limitation. In the last section we conclude our paper and discuss possible future works.

2 Gibson's Theories on Linguistic Complexity

We provide a very quick review of Gibson's IDT and DLT in order to make the readers familiar with their underlying concepts. The question of how to automatically compute linguistic complexity based on both theories with categorial proof nets will be covered in the sections (3.2) and (4).

Incomplete dependency theory is based on the idea of counting missing incomplete dependencies during the incremental processing of a sentence when a new word attaches to the current linguistic structure. The main parameter in IDT is the number of incomplete dependencies when the new word integrates to the existing structure. This gives an explanation for the increasing complexity of the examples (1a)-(1c) which have nested relative clauses. In (1a), *the reporter* has one incomplete dependency; in (1b), *the senator* has three incomplete dependencies; in (1c) *John* has five incomplete dependencies at the point of processing. For the sake of space, we only explain the most complex case, i.e. (1c) in which the incomplete dependencies at the moment of processing *John* are: (i) the NP *the reporter* is dependent on a verb that should follow it; (ii) the NP *the senator* is dependent on a different verb to follow; and (iii) the pronoun *who* (before *the senator*) is dependent on a verb to follow; (iv) the NP *John* is dependent on another verb to follow; and (v) the pronoun *who* (before *John*) is dependent on a verb to follow. These are five unsaturated or incomplete or unresolved dependencies. IDT in its original form suggests to calculate the maximum number of incomplete dependencies of the words in a sentence. One can observe that the complexity is proportional to the number of incomplete dependencies.

(1a) The reporter disliked the editor.

(1b) The reporter [who the senator attacked] disliked the editor.

(1c) The reporter [who the senator [who John met] attacked] disliked the editor].

(1d) The reporter [who the senator [who I met] attacked] disliked the editor].

Dependency Locality Theory is a distance-based referent-sensitive linguistic complexity measurement put forward by Gibson to supersede the predictive limitations of the incomplete dependency theory. DLT posits two integration and storage costs. In this paper, we have only focused on the integration cost. The linguistic complexity is interpreted as the locality-based cost of the integration of a new word to the dependent word in the current linguistic structure which is relied on the number of the intervened new discourse-referents. By performing a measurement on these referents, we can predict the relative complexity, such as structures with embedded pronouns, illustrated in example (1d). The experiments [24] support the acceptability of (1d) over (1c). According to the discourse-based DLT structural integration cost hypothesis, referents for the first-person pronoun *I* is already present in the current discourse, so, integrating across them consumes fewer cognitive resources than integrating across the new discourse referents before *John*. By means of just two aspects of DLT, namely the structural integration and the discourse processing cost we would be capable to predict a number of linguistic phenomena as we will see in details with some examples.

3 Complexity Profiling in Categorial Grammars

3.1 Proof-nets as parse structures

Our exposition of the henceforth classical material on proof nets for categorial grammar follows [20] — the main original papers on this topic are [16, 12, 22, 23]. Categorial grammars are defined from a set $\mathcal{C}$ of grammatical categories, defined from base categories (for instance $\mathcal{B} = \{np, n, S\}$) including a special symbol S (for sentence) and operators, for instance :

$$\mathcal{C} ::= \mathcal{B} \mid \mathcal{C}\backslash\mathcal{C} \mid \mathcal{C}/\mathcal{C}$$

The symbols $\backslash$ and $/$ can be viewed as logical connectives, namely implication(s) of a logic, namely intuitionistic non-commutative multiplicative linear logic better known as the Lambek calculus. Such formulas can be viewed as formulas of linear logic, with conjunction $\otimes$ disjunction $\parr$ and negation $(\_)^\perp$ because implications can be defined from negation and disjunction:

$$\begin{array}{rll} \text{Definition of } \backslash \text{ and } /: & A \setminus B \equiv A^\perp \parr B & B \,/\, A \equiv B \parr A^\perp \\ \text{De Morgan equivalences } & (A^\perp)^\perp \equiv A & (A \parr B)^\perp \equiv B^\perp \otimes A^\perp \\ & (A \otimes B)^\perp \equiv B^\perp \parr A^\perp & \end{array}$$

Some formulas have a polarity. Formulas are said to be positive (output) $\circ$ or negative (input) $\bullet$ as follows:[1]

$$a : \circ, \qquad a^\perp : \bullet$$

$\otimes$	$\bullet$	$\circ$
$\bullet$	*undefined*	$\bullet$
$\circ$	$\bullet$	*undefined*

$\parr$	$\bullet$	$\circ$
$\bullet$	*undefined*	$\circ$
$\circ$	$\circ$	*undefined*

So $a \parr a$ has no polarity, $a^\perp \parr b$ is positive, it is $a \setminus b$ while $b^\perp \otimes a$ is negative, it is the negation of $a \setminus b$. Categories are, roughly speaking, analogous to non-terminal categories in phrase structure grammars. But observe that they are endowed with an internal structure, i.e. $(np \setminus S) \,/\, np$ is a compound category and the rules make use of this internal structure, connectors $\backslash$ and $/$ and subcategories n, np and S. The rules (of the logic) do not depend on the language generated (or analyzed) by the grammar. They are the same for every language, and the lexicon makes the difference. The lexicon maps every word to a finite set of possible categories. A parse structure in a categorial grammar defined by a lexicon $\mathcal{L}$ for a sequence of words $w_1, \ldots, w_n$ simply is a proof of $c_1, \ldots, c_n \vdash S$ with $c_i \in \mathcal{L}(w_i)$ in some variant of the Lambek calculus. The rules for the basic (associative) Lambek calculus are:

$$\overline{A \vdash A}$$

[1] Here we are stricter than in other articles, i.e. we neither allow $\otimes$ of positive formulas nor $\parr$ of negative formulas, because we only use the $\backslash$ and $/$ symbols in categories (and not $\otimes$): only combining heterogeneous polarities guarantees that a positive formula is a category, and that a negative formula is the negation of a category.

$$\frac{H_1, H_2 \ldots, H_{n-1}, H_n \vdash C}{H_1, H_2 \ldots, H_{n-1} \vdash C \,/\, H_n}$$

$$\frac{H_1, H_2 \ldots, H_{n-1}, H_n \vdash C}{H_2 \ldots, H_{n-1}, H_n \vdash H_1 \setminus C}$$

$$\frac{H_1, \ldots, H_i, \ldots, H_n \vdash C \qquad G_1, \ldots, G_n \vdash A}{H_1, \ldots, H_{i-1}, G_1, \ldots, G_n, A \setminus H_i, H_{i+1}, \ldots, H_n \vdash C}$$

$$\frac{H_1, \ldots, H_{i-1}, H_i, H_{i+1} \ldots, H_n \vdash C \qquad G_1, \ldots, G_n \vdash A}{H_1, \ldots, H_{i-1}, H_i \,/\, A, G_1, \ldots, G_n, H_{i+1}, \ldots, H_n \vdash C}$$

Since the Lambek sequent calculus enjoys the cut-elimination property whereby a sequent is provable if and only if it is provable without the cut rule, we do not mention the cut rule. Categorial grammars are known for providing a transparent and computable interface between syntax and semantics. The reason is that the categorial parse structure is a *proof* in some variant of the Lambek calculus, and that this *proof* gives a way to combine semantic lambda terms from the lexicon into a lambda term which encodes a formula expressing the meaning of the sentence. We cannot provide more details herein, the reader is referred e.g. to [20, Chapter 3]. For instance, the categorial analysis of *Every barber shaves himself.* with the proper semantic lambda terms for each word in the sentence yields the logical form $\forall x.barber(x) \Rightarrow shave(x, x)$.

It has been known for many years that categorial parse structures, i.e. proof in some substructural logic, are better described as proof nets [23, 22, 18, 20]. Indeed, categorial grammars following the parsing-as-deduction paradigm, an analysis of a c phrase $w_1, , \ldots, w_n$ is a proof of c under the hypotheses $c_1, ..., c_n$ where c_i is a possible category for the word w_i; and proofs in those systems are better denoted by graphs called proof nets. The reason is that different proofs in the Lambek calculus may represent the same syntactic structure (constituents and dependencies), but these essentially similar sequent calculus proofs correspond to a unique proof net. A proof net is a graph, whose nodes are formulas, and it consists of two parts:

subformula trees of the conclusions, in the right order, whose leaves are the base categories, and branching are two connectives $⅋$ and $\otimes$ —– as we have seen formulas with $\setminus$ and $/$ can be expressed from base categories and their negations with $⅋$ and $\otimes$ — for nodes that are not leaves the label can be limited to the main connective of the subformula instead of the whole formula, without loss of information;

axioms that are a set of pairwise disjoint edges connecting a leaf z to a leaf $z^{\perp}$, in such a way that every leaf is incident to some axiom link.

However not all such graphs are proof nets, only the one satisfying:[2]

Acyclicity Every cycle contains the two edges of the same $⅋$ branching.

Connectedness There is a path not involving the two edges of the same $⅋$ branching between any two vertices.

[2] This list is redundant: for instance intuitionism plus acyclicity implies connectedness.

Intuitionism Every conclusion can be assigned some polarity.
Non commutativity The axioms do not cross (are well bracketed).

The advantage of proof-nets over sequent calculus is that they avoid the phenomenon known as spurious ambiguities— that is when different parse structures correspond to the same syntactic structure (same constituent and dependencies). Indeed proofs (parse structures) with unessential differences are mapped to the same proof net. A (normal) deduction of $c_1, ..., c_n \vdash c$ (i.e. a syntactic analysis of a sequence of words as a constituent of category c) maps to a (normal) proof net with conclusions $(c_n)^\perp, ..., (c_1)^\perp, c$ [23, 20]. Conversely, every normal proof net corresponds to at least one normal sequent calculus proof [22, 20].

3.2 Incomplete Dependency-Based Complexity Profiling and its Limitation

In this subsection we recall the IDT-based measure of the linguistic complexity by Morrill [21] which itself improves over a first attempt by Johnson [14]. Both measures are based on the categorial proof nets. The general idea is simple: to re-interpret the axiom links as dependencies and to calculate the incomplete dependencies during the incremental processing by counting the incomplete axiom links for each word in a given sentence. This is almost the same as Gibson's idea in his IDT, except the fact that he uses some principles of Chomsky Government-Binding theory [6] instead of the categorial proof nets. The notion of counting incomplete dependencies for each node, called complexity profiling, is more effective in terms of prediction than approaches that only measures maximum number of the incomplete dependencies or the maximum cuts [14].

We can rewrite IDT-based complexity profiling [21] by the following definitions:

Definition 1: Let π be a a syntactic analysis of $w_1, \cdots, w_n$ with categories $C_1, \ldots,$ C_n — that is a categorial proof net with conclusions $(C_n)^\perp, ..., (C_1)^\perp, S$. Let C_{i_0} be one of the C_i ($i \in [1, n]$). The incomplete dependency number of C_{i_0} in π, written as $ID_\pi(C_{i_0})$, is the count of axioms $c - c'$ in π such that $c \in (C_{i_0-m} \cup S)$ ($m \geq 0$) and $c' \in C_{i_0+n+1}$ ($n \geq 0$).

Definition 2: Let π be a a syntactic analysis of $w_1, \cdots, w_n$ with categories $C_1, \ldots,$ C_n — that is a categorial proof net with conclusions $(C_n)^\perp, ..., (C_1)^\perp, S$. We define the IDT-based linguistic complexity of π, written $f_{idt}(\pi)$ by $(1 + \sum_{i=1}^{n} ID_\pi(C_i))^{-1}$.

Definition 3: Given two syntactic analyses π_i and π_j, not necessarily of the same words and categories, we say that π_i is IDT-preferred to π_j whenever $f_{idt}(\pi_i) >$ $f_{idt}(\pi_j)$.

Example: Figure (1) shows the two relevant proof nets for examples (2a) with subject-extracted relative clause and (2b) with object-extracted relative clause (examples from [9]). The relevant complexity profiles for (2a) and (2b) are illustrated in the figure (2). As it can be seen, the total sum of the complexity for (2b) is greater than (2a), thus, it can predict correctly the preference of (2a) over (2b) which is supported

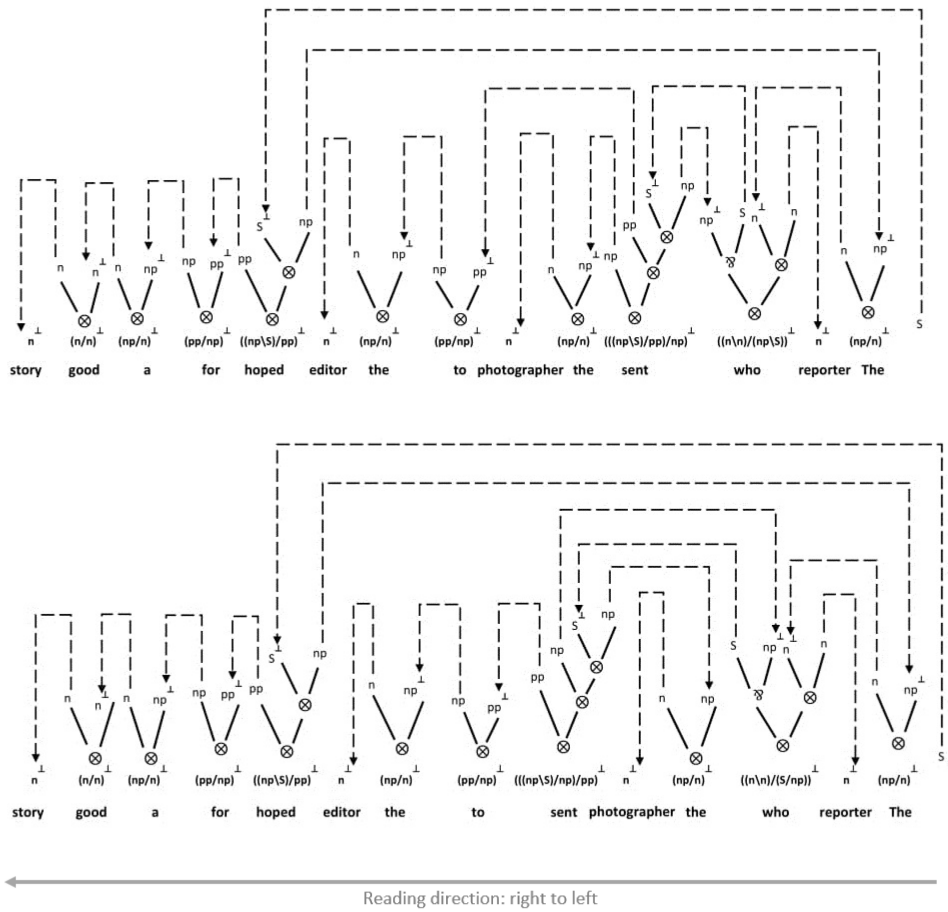

Fig. 1. Proof net analyses for (2a) located in top (subject-extracted relative clause) and (2b) in bottom (object-extracted relative clause).

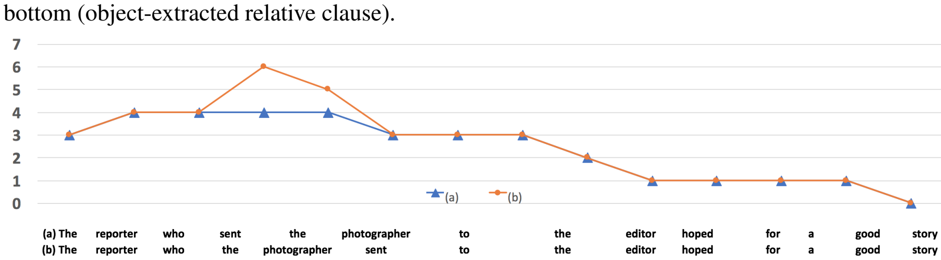

Fig. 2. IDT-based Complexity Profiles for (2a) and (2b).

by measuring reading time experiments [7].[3]

(2a) The reporter who sent the photographer to the editor hoped for a good story.

(2b) The reporter who the photographer sent to the editor hoped for a good story.

Obviously, IDT-based account does not use DLT as its underlying theory. Not surprisingly, the linguistic phenomena that can only be supported by DLT would not be supported by IDT-based complexity profiling. Figure (3) shows this failure. We can verify this by applying the definitions on the relevant proof nets as it is illustrated in the in the figure (4). As one may notice, the corresponding proof nets for the examples (1c) and (1d) are almost the same. Consequently, IDT-based complexity profiling cannot discriminate both examples, i.e. it generates the same number for both sentences in contrast to the experiments [24] as it is shown in the figure (3). This shows the importance of introducing DLT-based complexity profiling for proof nets in order to make more predictive coverage—as we will do so.

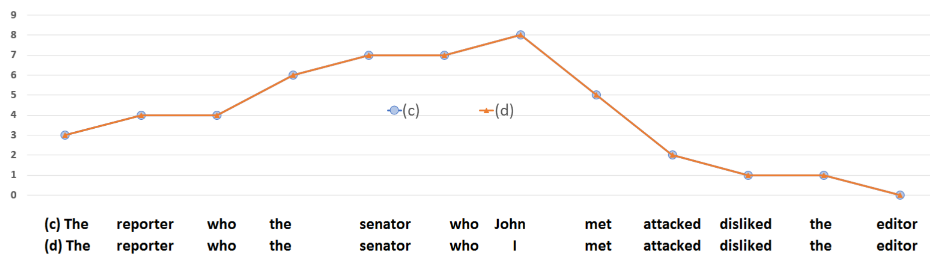

Fig. 3. IDT-based Complexity Profiles for (1c) and (1d).

4 A New Proposal: Distance Locality-Based Complexity Profiling

As we discussed, IDT-based complexity profiling is a distance-based measurement. However, it is not a referent-sensitive criterion and due to this fact, it cannot support some of the linguistic phenomena such as structures with embedded pronouns. One plausible strategy to overcome this lack is introducing DLT-based complexity profiling. This will allow us to have a referent-sensitive measurement. In this section, we provide the precise definitions of our DLT-based proposal on the basis of the categorial proof nets. Here they are:

Definition 4: A word w is said to be a discourse referent whenever it is a *proper noun*, *common noun* or *verb*.

[3] The same procedure, would show the the increasing complexity of the examples (1a)-(1c) by drawing the relevant proof-nets. This practice is avoided in this paper due the space limitation and its simplicity comparing to the running examples here.

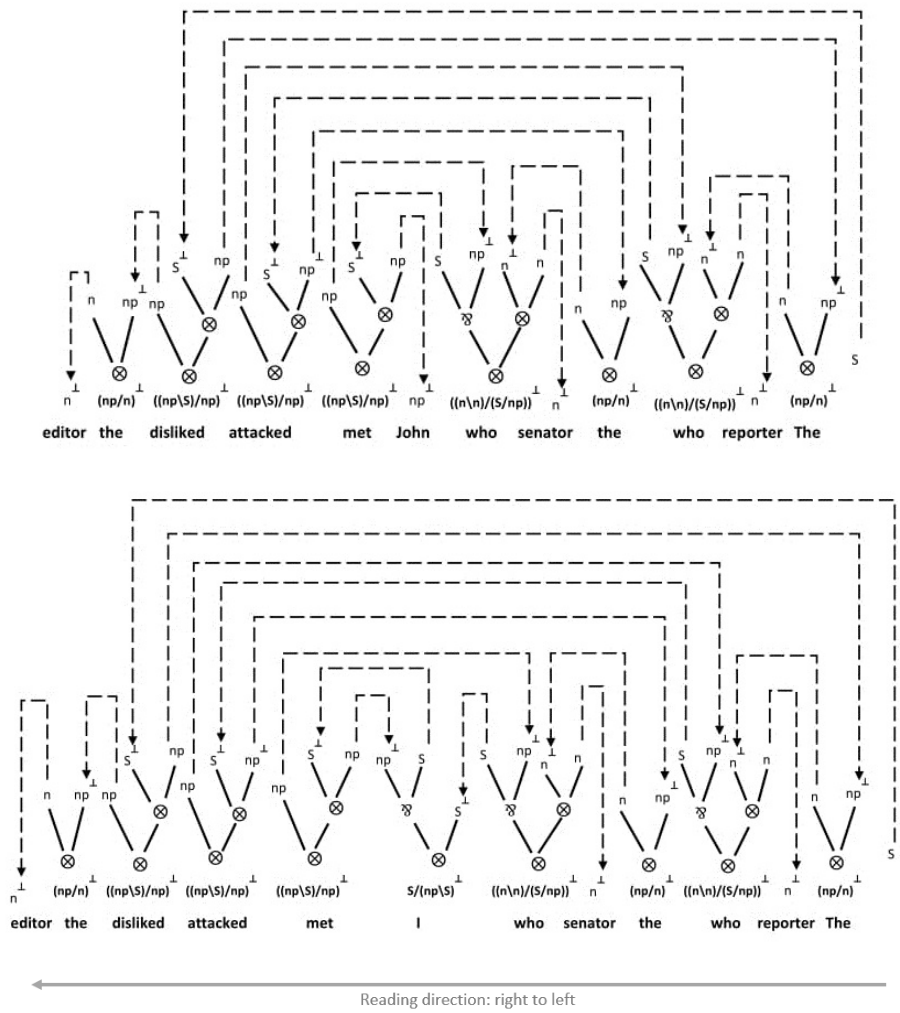

Fig. 4. Proof net analyses for both examples (1c) and (1d).

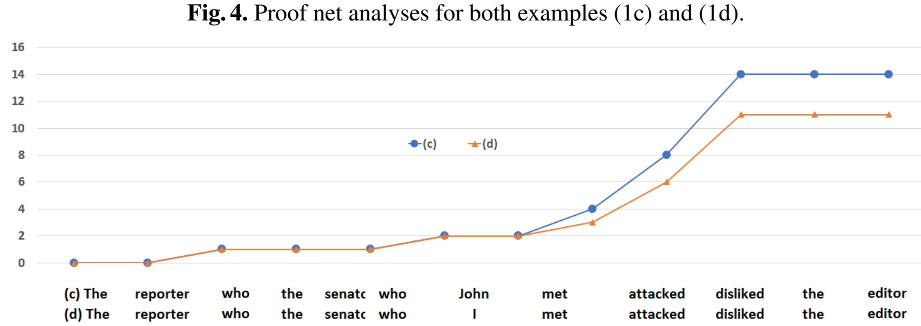

Fig. 5. Accumulative DLT-based Complexity Profiles for (1c) and (1d).

Definition 5: Let π be a a syntactic analysis of $w_1, \cdots, w_n$ with categories $C_1, \ldots,$ C_n — that is a categorial proof net with conclusions $(C_n)^{\perp}, ..., (C_1)^{\perp}, S$. Let $c - c'$ be an axiom in π such that $c \in C_i$ and $c' \in C_j$ ($i, j \in [1, n]$). We define the **length** of axiom $c - c'$ as the integer $i + 1 - j$.

Definition 6: Let π be a a syntactic analysis of $w_1, \cdots, w_n$ with categories $C_1, \ldots,$ C_n — that is a categorial proof net with conclusions $(C_n)^{\perp}, ..., (C_1)^{\perp}, S$. Let C_{i_0} be one of the C_i, and let consider axioms $c - c'$ with c in C_{i_0} and c' in some C_{i_0-k}. Let us consider the largest k for which such an axiom exists — this is the longest axiom starting from C_{i_0} with the previous definition. The dependency locality number of C_{i_0} in π, written $DL_{\pi}(C_{i_0})$ is the number of discourse referent words between $w_{i_0} : C_{i_0}$ and $w_{i_0-k} : C_{i_0-k}$. The boundary words, i.e. $w_{i_0} : C_{i_0}$ and $w_{i_0-k} : C_{i_0-k}$ should also be counted. Alternatively, it may be viewed as $k+1$ minus the number of non-discourse references among those $k + 1$ words.

Definition 7: Let π be a a syntactic analysis of $w_1, \cdots, w_n$ with categories $C_1, \ldots,$ C_n — that is a categorial proof net with conclusions $(C_n)^{\perp}, ..., (C_1)^{\perp}, S$. We define the DLT-based linguistic complexity of π, written $f_{dlt}(\pi)$ by $(1 + \sum_{i=1}^{n} DL_{\pi}(C_i))^{-1}$.

Definition 8: Given two syntactic analyses π_i and π_j, not necessarily of the same words and categories, we say that π_i is DLT-preferred to π_j whenever $f_{dlt}(\pi_i) > f_{dlt}(\pi_j)$.

Examples: We apply our new metric on examples (1c) and (1d). Figure (4) shows the relevant proof net for (1c) and (1d). The proof nets for both examples are the same except a difference in one of the lexicons in each example, i.e. *John* and *I*.[4] Figure (5) shows the accumulative chart-based representation of our measurement for each example. The axis Y shows the accumulative sum of dependency locality function applied to each category in axis X. The quick analysis of the profiles shows the total complexity numbers 14 and 11 for (1c) and (1d), respectively. This correctly predicts the preference of example (1d) over (1c) which was not possible in the IDT-based approaches.

The measurement for dependency locality number is quite straightforward. As an example, we calculate the dependency locality number for the word *attacked* in figure (4) for (1d). We can find the longest axiom link starting from *attacked* and ended to its right most category, namely, *who*. Then, we count the number of discourse referents intervened in the axiom link, which is actually three; namely, *attacked*, *met* and *senator*.

We can evaluate our proposal for measuring the linguistic complexity against other linguistic phenomena. Our experiment shows that the new metric supports both referent-sensitive and some of the non-referent-sensitive phenomena such as garden pathing, unacceptability of center embedding, preference for lower attachment and passive paraphrases acceptability. For saving space, we just illustrate Passive Paraphrases Acceptability [21] in this paper. This linguistic phenomenon is illustrated by examples (3a)

[4] Following Lambek [16], we have assigned the category $S/(np\backslash S)$ to relative pronoun *I*. Note that even assigning np, which is not a type-shifted category, would not change our numeric analysis at all.

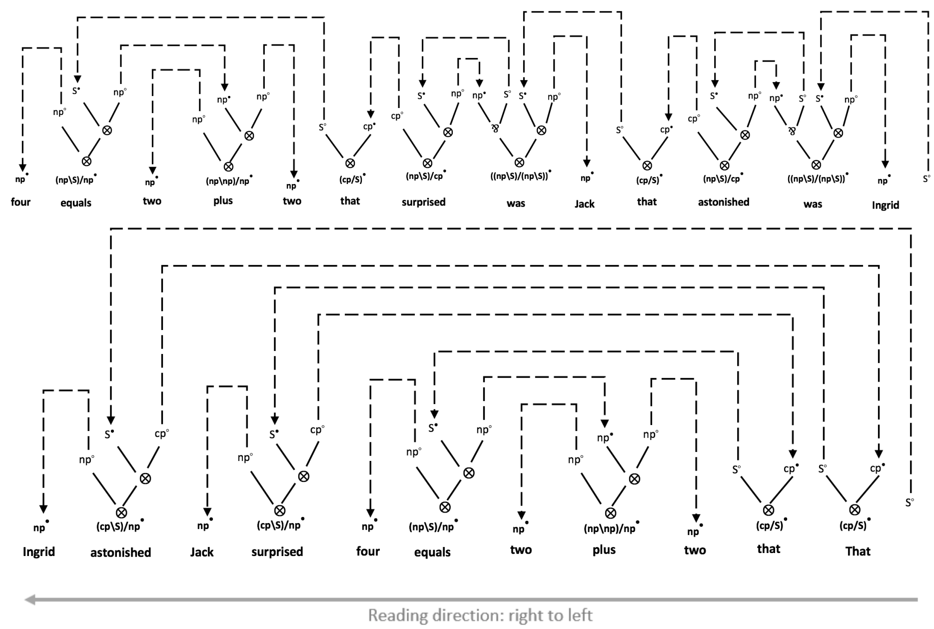

Fig. 6. Proof net analyses for (3a) in the top and (3b) in the bottom.

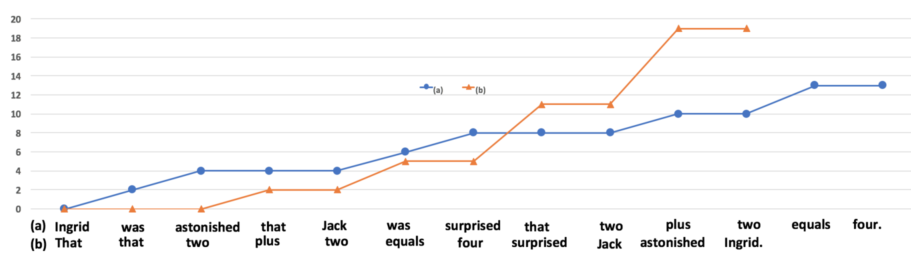

Fig. 7. Accumulative DLT-based complexity profiles for (3a) and (3b)

and (3b). Notice that the DLT-based complexity profile of the (3a) is lower even though the number of the sentences and the axiom links are more comparing to (3b). The real preference is on the syntactic forms in which (3a) is preferred to (3b). The relevant proof nets and the accumulated complexity profiles are illustrated in the figures (6) and (7), respectively.

Example 3a: Ingrid was astonished that Jack was surprised that two plus two equals four.

Example 3b: ?That that two plus two equals four surprised Jack astonished Ingrid.

5 Limitation

There is a limitation in our approach and it is the problem of ranking valid semantic meanings of a given multiple-quantifier sentence which cannot be supported by our proposal. A study [3] has shown the same problem in the IDT-based approach when dealing with some type of the expressions such as sentence-modifier adverbials and nested sentences. Thus, both IDT-based and DLT-based complexity profiling cannot correctly predict ranking the quantifier scoping problem. Hopefully, this can be treated with the hybrid models [3] in which Hilbert's epsilon and tau [13, 4] are exploited.

6 Conclusion and Possible Extensions

In this paper we explored how our DLT-based complexity profiling on proof nets can give a proper account of the complexity of a wide range of linguistic phenomena. We have also shown that IDT-based method could not support referent-sensitive linguistic performance phenomena. This was one of the main reasons for introducing the DLT-based complexity profiling technique within the framework of Lambek calculus. There are some extensions for our study and research:

- As we mentioned it is possible to bridge our model with other study [3] to overcome the problem of ranking quantifier scoping, which our proposal already has. As we discussed, we can exploit Hilbert's epsilon and tau operators [13, 4] for neutralizing the quantifier effect and making possible the complexity measurement by the penalty cost of the quantifiers re-ordering.
- Another important direction is to take into account not only the axioms of the proof-nets but also the logical structure, i.e., par-links, tensor-links and the correctness criterion. This is important indeed, because this structure is needed to compute the logical form (semantics) from the syntactic structure given by proof nets. For instance, nesting Lambek slashes (that are linear implications, and therefore par-links in the proof net) corresponds to higher order semantic constructions (e.g. predicates of predicates) and consequently this nesting of par-links increases the complexity of the syntactic and semantic human processing.
- It is possible to combine our method with studies in other directions: One potential candidate is the task of sentence correction/completion in Lambek Calculus [17]. The other task is measuring semantic gradience in natural language. Some line of research suggests this feature within lexical/compositional frameworks by creating and enrichment of the wide-coverage weighted lexical resources from crowd-sourced data [15].

Acknowledgement We would like to show our gratitude to Philippe Blache for his insightful discussion at our lab and also for inspiration that we got from his papers [1, 2]. We would like to thank our colleague Richard Moot as well for his numerous valuable comments on this work.

References

1. Blache, P.: A computational model for linguistic complexity. In: Proceedings of the first International Conference on Linguistics, Biology and Computer Science (2011)
2. Blache, P.: Evaluating language complexity in context: New parameters for a constraint-based model. In: CSLP-11, Workshop on Constraint Solving and Language Processing (2011)
3. Catta, D., Mirzapour, M.: Quantifier scoping and semantic preferences. In: Proceedings of the Computing Natural Language Inference Workshop (2017)
4. Chatzikyriakidis, S., Pasquali, F., Retoré, C.: From logical and linguistic generics to hilbert's tau and epsilon quantifiers. IfCoLog Journal of Logics and their Applications **4**(2), 231–255 (2017)
5. Chomsky, N.: Aspects of the Theory of Syntax. MIT Press, Cambridge (1965)
6. Chomsky, N.: Some concepts and consequences of the theory of government and binding, vol. 6. MIT press (1982)
7. Gibson, E., Ko, K.: An integration-based theory of computational resources in sentence comprehension. In: Fourth Architectures and Mechanisms in Language Processing Conference, University of Freiburg, Germany (1998)
8. Gibson, E.: Linguistic complexity: Locality of syntactic dependencies. Cognition **68**(1), 1–76 (1998)
9. Gibson, E.: The dependency locality theory: A distance-based theory of linguistic complexity. Image, language, brain pp. 95–126 (2000)
10. Gibson, E.A.F.: A computational theory of human linguistic processing: Memory limitations and processing breakdown. Ph.D. thesis, Carnegie Mellon University Pittsburgh, PA (1991)
11. Gibson, E., Thomas, J.: The processing complexity of english center-embedded and self-embedded structures. In: of Massachusetts, U. (ed.) The proceedings of the North-Eastern Linguistic Society 1996 (1996)
12. Girard, J.Y.: Linear logic. Theoretical Ccomputer Science **50**, 1–102 (1987)
13. Hilbert, D.: Die logischen grundlagen der mathematik. Mathematische Annalen **88**(1), 151–165 (1922)
14. Johnson, M.E.: Proof nets and the complexity of processing center-embedded constructions. In: Retoré, C. (ed.) Special Issue on Recent Advances in Logical and Algebraic Approaches to Grammar. Journal of Logic Language and Information, vol. 7(4), pp. 433–447. Kluwer (1998)
15. Lafourcade, M., Mery, B., Mirzapour, M., Moot, R., Retoré, C.: Collecting weighted coercions from crowd-sourced lexical data for compositional semantic analysis. In: New Frontiers in Artificial Intelligence, JSAI-isAI 2017. Lecture Notes in Computer Science. vol. 10838, pp. 214–230. Springer (2018)
16. Lambek, J.: The mathematics of sentence structure. The American Mathematical Monthly **65**(3), 154–170 (1958)
17. Mirzapour, M.: Finding missing categories in incomplete utterances. In: 24e Conférence sur le Traitement Automatique des Langues Naturelles (TALN). p. 149
18. Moot, R.: Proof nets for linguistic analysis. Ph.D. thesis, UIL-OTS, Universiteit Utrecht (2002)
19. Moot, R., Retoré, C.: The logic of categorial grammars: a deductive account of natural language syntax and semantics, vol. 6850. Springer (2012)
20. Moot, R., Retoré, C.: The logic of categorial grammars: a deductive account of natural language syntax and semantics, LNCS, vol. 6850. Springer (2012), http://www.springer.com/computer/theoretical+computer+science/book/978-3-642-31554-1

21. Morrill, G.: Incremental processing and acceptability. Computational linguistics **26**(3), 319–338 (2000)
22. Retoré, C.: Calcul de Lambek et logique linéaire. Traitement Automatique des Langues **37**(2), 39–70 (1996)
23. Roorda, D.: Proof nets for Lambek calculus. Logic and Computation **2**(2), 211–233 (1992)
24. Warren, T., Gibson, E.: The effects of discourse status on intuitive complexity: Implications for quantifying distance in a locality-based theory of linguistic complexity. In: Poster presented at the Twelfth CUNY Sentence Processing Conference, New York (1999)