An Annotational Approach to Compositional Semantics

Mats Dahllöf

Uppsala University
Department of Linguistics
E-mail: mats.dahllof@ling.uu.se

1 Introduction

In this paper[1] I will discuss a framework for semantics which allows us to record truth-conditional and compositional analyses as dependency-style corpus annotations in a direct and fine-grained fashion. This method eliminates the need for a semantic representation formalism by decomposing semantic information into simple statements about (word or morpheme) tokens. A collection of such data would form a new kind of linguistic treebank. The main purpose of this article is to show that the present approach makes it possible to combine formal semantics and corpus-oriented study of language use in new and interesting ways. The methodology of this framework, which I call Token Dependency Semantics (TDS, Dahllöf [4]), is in several respects different from the common one(s) in traditional formal semantics. TDS nevertheless delivers a fairly conventional (but ontologically restrained) analysis of truth-conditional meaning.

Currently, there is a detailed TDS account for a sample fragment of English. A system based on Head-driven Phrase Structure Grammar (HPSG) implements the description of this fragment. The HPSG system, which runs on a computer (Dahllöf [5]), shows that it is possible to apply the TDS semantics according to a conventional method in computational semantics, viz. integrated as a component of a grammar. The TDS idea is however independent from HPSG and in important respects different from all other HPSG-based proposals on semantics.

The plan of the paper is as follows: Section 2 explains the basic ideas behind TDS and their background. In section 3, I take a more general look at the metho-

[1] I am grateful to the three anonymous reviewers for advise on how to expand and elaborate the extended abstract and to Per Weijnitz for comments on a draft.

dology of TDS. The treatment of scope and scopal underspecification is the subject of section 4. Section 5 examines the TDS approach to truth conditions, contrasting it with that of more traditional varieties of formal semantics. In particular, I stress the annotational and surface-oriented nature of TDS. There is further illustration of this aspect in section 6, which focuses on the prospect of using TDS as a corpus annotation scheme and its relation to existing semantic annotation schemes. Section 7, finally, is a concluding discussion.

2 Token Dependency Semantics: Theoretical Background

A TDS analysis of a piece of discourse is a set of statements which primarily are about linguistic tokens. It decomposes composite tokens into semantically elementary subtokens and describes compositional structure in terms of a small number of dependency relations, holding of pairs of tokens (typically words). Lexical semantics is a matter of properties of these elementary tokens.

The TDS framework grew out of an attempt to implement and develop Davidson's [6, Essay 7] "paratactic" proposal on the semantics of intensional constructions. Davidson argues that an indirect speech report like *Galileo said that the earth moves* relates Galileo by the *said* relation to the subordinate clause token, not to something more abstract, such as a proposition or a representational structure. This idea motivates a kind of analysis that finds semantic structure directly in the linguistic tokens. However, I shall not discuss this aspect of TDS further here (but Dahllöf [4] elaborates on the issue). The TDS semantics also makes use of many ideas from the framework of Minimal Recursion Semantics (henceforth MRS, Copestake et al. [2], Copestake et al. [3]). Let me mention the most important ones:

- Semantic structures are sets of so-called elementary predications, which are instances of e.g. predicates and quantifiers. Coindexation (variable sharing) and scope relationships connect the predications.

- A relation of *equality modulo quantifiers* makes it possible to formulate underspecified descriptions of scopal relationships. It seems that this relation is particularly handy from the point of view of (English) compositional semantics.

- The compositional system of the implemented TDS sample grammar follows the hooks-and-holes model of Copestake et al. [3]. This model disentangles the semantic information structures and operations from the syntactic ones, and controls accessibility (which pieces of information the compositional operations can access; consult Dahllöf [4] for details).

MRS is an approach to semantic representation. The TDS framework applies and revises the principles of MRS in such a way that they form guidelines for direct description of linguistic tokens, thus avoiding the detour via representations.

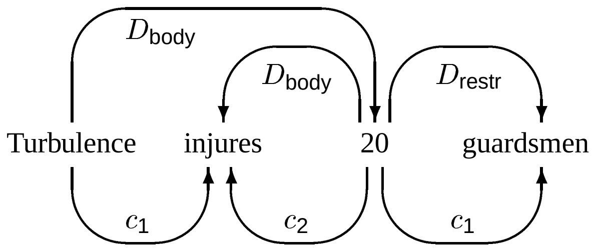
This leads us to the central methodological principle of TDS: the identification of the elementary predications with the corresponding linguistic tokens. Through this step, we eliminate the distinction between linguistic tokens and semantic structures. However, we should not require that all tokens be given predication status; some tokens may only play a syntactic role (see e.g. example (4), below).

As a consequence of this, TDS has to introduce relations applying directly to pairs of tokens to encode the semantic links that MRS captures by means of structure sharing in the representations (feature structures). Thus, the TDS scope relations are of this kind. TDS does not use so-called handle features for this purpose, as do MRS. Likewise, TDS expresses coindexation by means of relations applying to pairs of tokens, not by means of co-occurrence of variables. The distinctions among different argument positions correspond to distinctions among the TDS coindexation relations.

This leaves us with two groups of primitive TDS dependency relations, which we need in the analysis of truth conditions, viz. *scope relations* and *coindexation relations*. The relation of *equality modulo quantifiers*, that plays a prominent part in the account of scopal underspecification, is a derived one. I will turn to the issue of underspecification below, after having discussed truth-conditionally specific descriptions.

Let me exemplify all of this and add details by looking at the likely reading of a news headline, (1).

(1)



The TDS analysis of (1) takes the form of a graph whose nodes are the tokens which we treat as semantically elementary. As I hinted above, the current version of the TDS sample grammar (Dahllöf [4], [5]) produces scopally underspecified descriptions covering the possible readings, rather than truth-conditionally specific ones.

The verb and the noun in (1) function as predicates. I regard the two NPs as one-place quantifiers binding, respectively, the first and second argument positions on the verb. The two coindexation relations c_1 and c_2 reflect this difference in binding. There is also a c_1 coindexation between the determiner *20* and the noun *guardsmen* (a one-place predicate). This directly annotational mode of analysis stands in sharp contrast to the usual method of using representations involving variables to connect quantifiers and argument positions, e.g. as in the (generalized quantifier logic) formulae (2) and (3).

(2) $\textit{turbulence}(x_1, \textit{twenty}(x_2, \textit{guardsman}(x_2), \textit{injures}(x_1, x_2)))$

(3) $\textit{some}(x_1, \textit{turbulence}(x_1), \textit{twenty}(x_2, \textit{guardsman}(x_2), \textit{injures}(x_1, x_2)))$

The TDS coindexation approach presupposes that there is an upper limit to the number of argument positions on lexical predicate concepts. This seems to be a safe assumption.

A determiner, like *20*, is a two-place quantifier taking both a *restriction* and a *body* argument. We may distinguish the two arguments by using two immediate outscoping relations, D_{restr} and D_{body}. Ordinary logical representation languages capture this distinction by means of syntactic position, as the formulae (2) and (3) illustrate.

The two scope relations arrange the four word tokens into a tree structure, the *Turbulence* token being its root. It is possible to formulate a truth definition, e.g. a model-theoretic one, directly for tree structures of this kind (Dahllöf [4]).

3 Some General Principles behind TDS

The core principles of the TDS framework include a number of empirical assumptions and methodological constraints. These require that a TDS analysis take the form of a set of structurally and conceptually simple dependency-style annotations. They also eliminate redundancy. So, a truth-conditional TDS analysis of a sentence token, T, conforms to the following requirements:

- The semantic description of the sentence token, T, is a graph, G.
- Each node in G is identical to a word (or morpheme) token part of T. (This rules out a "determiner"-predicate decomposition in the manner of the analysis (3), for instance.)
- The relations defining the edges of G are primitive, in the sense of not allowing further analysis into conceptually more basic relations. (I will call the relations defining the edges *primitive TDS relations*).
- As we have seen, there are two kinds of primitive TDS relation. First, there are *scope* relations, e.g. D_{restr} (restriction) and D_{body}. Secondly, there are *coindexation* relations, such as c_1 and c_2.

- The number of primitive TDS relations is small (less than ten in Dahllöf [4]).
- The primitive TDS relations are mutually exclusive.

Furthermore, the more general notion of scope, the immediate outscoping relation, from which we get the general (not necessarily immediate) outscoping relation as its transitive closure, is related in a simple way to the primitive scope relations. The following are some basic TDS assumptions concerning scope:

- The union of the primitive scope relations forms the immediate outscoping relation.
- The (derived) immediate outscoping relation arranges the nodes of G into a tree.
- The graph (scopal tree) G is semantically interpretable in a simple and natural way by a truth definition (given in Dahllöf [4]).

A nominalist conception of language in the style of Davidson [6] has guided the development of TDS, both as regards its analysis of linguistic structure and its treatment of semantic interpretation. The aim has been to avoid positing abstract constructs wherever possible.

4 Scopal Underspecification

It seems that the compositional principles often leave sentences scopally underspecified. A grammar treating sentences should consequently produce semantic analyses which make claims about scope at the appropriate level of specificity. We may then look upon the process of determining truth conditions, which includes specifying the scopal relationships, as a next step of contextual interpretation. The issue of scopal underspecification has drawn considerable attention in computational semantics during the last fifteen years or so. The MRS treatment, which is one of the best worked out proposals in this area, is easy to import into the TDS framework.

Both coindexation and constraints which are directly scope-related impose conditions on possible scopings. Coindexation plays a role through the requirement that a quantifier token binding an argument position on another word must outscope that word. There are also specificly scope-related constraints of lexical origin. As we saw above, TDS expresses them in terms of the relation of *equality modulo quantifiers* (henceforth $=_q$, definition below). A sentence like (4) provides two examples. The $=_q$ relation holds of the determiner token and the noun it modifies in

NPs like these. A TDS grammar should analyze (4) as shown in (5), where token variables (T_1, T_2) correspond to restrictions subtrees.

(4) Several speakers of two languages laughed.

(5) $c_1(\text{Several}, \text{speakers}) \wedge c_1(\text{Several}, \text{laughed}) \wedge$

$D_{\mathsf{restr}}(\text{Several}, T_1) \wedge T_1 {=_q} \text{speakers} \wedge$

$c_1(\text{two}, \text{languages}) \wedge c_2(\text{two}, \text{speakers}) \wedge$

$D_{\mathsf{restr}}(\text{two}, T_2) \wedge T_2 {=_q} \text{languages}$

Here, I use the words as names of the matching tokens, and assume that the *speakers* token stands for a relational speaker-of concept, whose c_1 and c_2 arguments correspond to the speaker and language roles, respectively. I also assume that the preposition lacks predication status. The annotations in (5) capture coindexation and, at the grammatical level of specificity, the scopal situation. The determiner entries impose conditions on the determiner restrictions, T_1 and T_2 (Dahllöf [4, p. 353]). Quantifier body scopes tend to be less constrained.

We may define the TDS version of the $=_q$ relation as follows (Dahllöf [4, p. 342]): $T_A {=_q} T_B$ if and only if $T_A = T_B$ or there is (in the TDS graph at hand) a token T_x, such that $D_{\mathsf{body}}(T_A, T_x)$ (T_A being a quantifier) and $T_x {=_q} T_B$.

This allows two readings (scopal resolutions) of (4), which the TDS statements in (6) and (8), respectively, characterize. (7) is a logic formula structurally equivalent to (6), as is (9) to (8).

(6) $D_{\mathsf{restr}}(\text{Several}, \text{two}) \wedge D_{\mathsf{body}}(\text{Several}, \text{laughed}) \wedge$

$D_{\mathsf{restr}}(\text{two}, \text{languages}) \wedge D_{\mathsf{body}}(\text{two}, \text{speakers})$

[i.e. $T_1 = \text{two}$ and $T_2 = \text{languages}$]

(7) $\textit{several}(x_1, \textit{two}(x_2, \textit{languages}, \textit{speaker-of}(x_1, x_2)), \textit{laughed}(x_1)))$

(8) $D_{\mathsf{restr}}(\text{Several}, \text{speakers}) \wedge D_{\mathsf{body}}(\text{Several}, \text{laughed}) \wedge$

$D_{\mathsf{restr}}(\text{two}, \text{languages}) \wedge D_{\mathsf{body}}(\text{two}, \text{Several})$

[i.e. $T_1 = \text{speaker}$ and $T_2 = \text{languages}$]

(9) $\textit{two}(x_2, \textit{languages}(x_2), \textit{several}(x_1, \textit{speaker-of}(x_1, x_2)), \textit{laughed}(x_1))$

Now, the presence of token variables (T_1, T_2) in the underspecified description (5) introduces, in a manner of speaking, scopal nodes which we have not *yet* identified with tokens. One way to get rid of them is to use a derived relation, which we may call *restriction modulo quantifiers* ($D_{\mathsf{restr\text{-}q}}$), such that $D_{\mathsf{restr\text{-}q}}(T_A, T_B)$ holds if and only if there is a token T_x, such that $D_{\mathsf{restr}}(T_A, T_x)$ and $T_x {=_q} T_B$. This allows us to describe both specific and underspecified scopal situations by means of relations holding of two identified tokens. In both cases, scopal annotations take the form of ordinary dependency edges.

5 TDS as a Truth-Conditional Semantics

Compared to most work in compositional formal semantics, the TDS approach is a methodologically and ontologically restrained one, imposing strict empirical conditions on the relation between the audible or visible surface of language use and the form of its compositional semantic analysis. Most varieties of formal semantics primarily relate linguistic *expressions* to semantic representations and to whatever these represent. Linguistic tokens only enter the picture as instances of expressions. These theories do not give any simpler account of how (phrase, word, or morpheme) subtokens relate to each other and to a superordinate sentential token than one involving the full compositional derivation. As different approaches as Montague grammar (Montague [12]) and MRS exemplify this. We end up with a formula in intensional logic (representing a proposition in Montague's sense) or with a certain kind of feature structure (standing for a bag of elementary predications), neither of whose elementary components have any explicit connection to linguistic tokens. Consequently, we cannot generally in a direct way decompose semantic analyses according to these frameworks into simple statements about individual tokens on a phrase, word, or morpheme granularity level. As far as I know, TDS is the first variety of truth-conditional semantics that primarily analyzes linguistic tokens, and in doing so avoids the traditional representation-based method.

Computational semantics often employs mechanisms that tangle up the links between semantic representation components and linguistic tokens. For instance, Montague grammar assigns a determiner such as *every* a representation like the one in (10) (Dowty et al. [7, pp. 196]).

(10) $\lambda P[\lambda Q \forall x[P\{x\} \rightarrow Q\{x\}]]$

Although the λ-term (10) represents one function, it may disintegrate in the computation of a sentence representation, as we combine it with the representations of other words and subject it to λ-conversion. The important thing from the point of view of Montague grammar is that semantics associates each sentence with the right proposition (in Montague's sense). A token of *every* will not stay in a one-to-one relationship to any one component of the semantic representation. In fact, it is arbitrary whether the (10) λ-term remains as a constituent of the sentence representation at all. λ-conversions may decompose it without modifying the representation in any way that is semantically (model-theoretically) essential.

Some approaches to syntactic analysis, e.g. transformational ones, posit underlying *syntactic* structures whose elementary components do not directly correspond to audible (or visible) parts of linguistic tokens. The subscripted pronoun mechanism of Montague (cf. Dowty et al. [7, pp. 203–206]) is an example of a transformational operation (substitution) serving the purposes of computational semantics. A semantic theory making use of such mechanisms only allows us to understand the relation between a piece of audible language use and its semantic representation in terms of, inter alia, a series of mediating derivational structures. Again we see how the methodology of Montague grammar treats sentences in a holistic way that precludes rendering a semantic analysis as a set of simple statements about smaller tokens.

The methodological constraints of TDS eliminate several sources of analytical latitude which characterize many traditional approaches to formal semantics. For instance, TDS does not give us the resources of a logical calculus that allows us to assume that a single morpheme introduces a structure of meaning components as complex as formula (10). A TDS-style semantic analysis produces conceptually primitive and surface-oriented claims, without positing a rich inventory of abstract structure. Statements concerning compositional semantics are annotations applying to concrete linguistic tokens. These annotations are conceptually primitive in the sense that we cannot define them in more basic terms. Furthermore, a TDS-based semantics is completely monotonic, both as regards semantic composition and scopal resolution, i.e. those operations only add statements that leave old pieces of information unchanged. Several frameworks of *syntax* have explicitly rejected transformational operations for methodological reasons, favouring a more surface-oriented mode of analysis (see e.g. Sag and Wasow [13]). The TDS way of looking at language involves a conceptually similar rejection of inaudibilia in *semantic* analysis.

6 TDS as an Annotation Scheme

The TDS annotation scheme is comparable to a number of previous approaches to semantic annotation. Bod et al. [1] use a corpus of 716 constituent trees, annotating each node with an extensional type theory formula, as training and evaluation data for heuristic semantic interpretation. This scheme is similar to TDS in aiming for a truth-conditional analysis. Bod et al. follow the tradition of Montague, which I discussed above, in describing compositional semantics in terms of complex syntactico-semantic representations. The TDS method of decomposing semantic information in terms of conceptually primitive and surface-oriented concepts is, from a structural point of view, an antithesis of the Montague-style scheme of Bod et al.

There are also semantic annotation schemes which are more structurally similar to TDS, but not geared towards a truth-conditional analysis. Of the two kinds of compositional connections according to the TDS model, the only one that empirical linguists have studied extensively is coindexation, i.e. operator-argument structure.

It is obviously one of the most central aspects of both syntax and semantics. We can thus compare the TDS treatment of coindexation with many frameworks of syntax and semantics. Operator scope, on the other hand, has almost exclusively been a concern for formal semanticists.

In the field of treebank linguistics, semantically oriented research has focused on semantic roles of verb arguments. Projects such as the Proposition Bank (Kingsbury et al. [10]) and FrameNet (Lowe et al. [11]) have developed and used annotation schemes exemplifying this.

The Proposition Bank project annotates a subset of the Penn English Treebank with information about verb arguments. Kingsbury and Palmer [9] use the sentence in (11) to illustrate their annotation scheme, which yields the description in (12).

(11) ... Sotheby's ... offered the Dorrance heirs a money-back guarantee

(12)

Arg0:	Sotheby's
Rel:	offered
Arg2:	the Dorrance heirs
Arg1:	a money-back guarantee

It is easy to understand the Proposition Bank verb argument labels in terms of the TDS coindexation relations. TDS also annotates other kinds of argument, e.g. those of determiners and nouns, which are not yet treated in the Proposition Bank. So, using the words of the example to form names of the corresponding tokens, we arrive at the TDS coindexation analysis in (13), keeping my relation c_1 for the noun binding.[2]

(13) $\text{Arg0}(\text{Sotheby's}, \text{offered}) \wedge c_1(\text{the}, \text{Dorrance\_heirs}) \wedge$

$\text{Arg2}(\text{the}, \text{offered}) \wedge c_1(\text{a}, \text{money-back\_guarantee}) \wedge$

$\text{Arg1}(\text{a}, \text{offered})$

The so-called *frame* for a verb lists the semantic roles corresponding to the labels Arg0, Arg1, Arg2, etc., in relation that verb. Sentence (11) exemplifies a certain frame for the verb *offer*. As annotation work progresses, the number of frames grows. The frame collection provides a key to the corpus annotations and guidelines for the annotators. Although semantic roles have not been a primary concern in the development of TDS, the Proposition Bank approach to semantic roles is fully applicable also if we employ a TDS-style coindexation annotation scheme.

It seems that it generally is quite an expensive affair to add semantic annotations to a corpus. Human annotators have to make judgements which typically are more fine-grained and sophisticated than those required for syntactic analysis.

[2] The TDS framework should also deliver an analysis of the compounds *Dorrance heirs* and *money-back guarantee*, but I leave them unanalyzed here.

As regards the prospect of using automated methods for annotation, it is plausible that the TDS coindexation annotation scheme, as far as verbs are concerned, is comparable to other argument annotation schemes, in view of the similarities discussed above. The TDS scheme also gives us a treatment of other argument-taking parts of speech. We may assume that the coindexation annotations pertaining to these are easier to produce than the verb-related ones, at least for a language like English, as the syntactic patterns corresponding to the binding of e.g. noun, adjective, and preposition arguments are comparatively few and simple. Underspecified annotations based on the $D_{\textsf{restr-q}}$ relation (which I introduced in section 4) closely correlate with syntactic connections (at least as far as the construction types treated in Dahllöf [4] go). It is thus a reasonable conjecture that $D_{\textsf{restr-q}}$ annotation would be about as tough for an automatic annotation system as coindexation annotation. Truth-conditionally specific scope annotation is likely to be more difficult, as it involves resolving scopal ambiguities, which is a matter of contextual interpretation (cf. Villalta [14]).[3]

The TDS annotation scheme delivers truth-conditional compositional analyses in the form of conceptually simple statements about linguistic tokens. This orientation sets it apart from other semantic annotation schemes. Even if the TDS framework ultimately turns out to be untenable as an account of linguistic meaning, the TDS relations may still give us a useful and relatively theory-neutral semantic annotation scheme.

7 Concluding Discussion

The main purpose of this article is to argue that a TDS-style conception of compositional and truth-conditional semantics yields a kind of annotation scheme that makes it possible to explore corpus data in new and interesting ways, while the Dahllöf [4] article tries to show that TDS applies successfully to a number of linguistic construction types.

The TDS view of semantic analysis focuses on concrete language use. It decomposes utterances into semantically elementary tokens and gives an account of their interrelations. TDS avoids the detour via semantic representations, by capturing coindexation and scope in terms of relations applying directly to linguistic tokens. I have not said much about the treatment of lexical aspects. Let me just note that a TDS-style semantics somehow must associate predication tokens with appropriate semantic properties, e.g. extensions in the case of predicate word tokens. There are different ways in which we may spell out the details.

[3] Gambäck and Bos [8] do however describe a successful heuristic method for resolution of scopal ambiguities.

The TDS method avoids mechanisms and structures that apply to sentences in a holistic fashion and makes it possible to decompose semantic analyses into simple statements, each one describing how two word (or morpheme) tokens are related coindexically or scopally. All of this works in a simple monotonic fashion and in combination with a state-of-the-art treatment of scopal underspecification.

The TDS framework is at an early stage of development and several open issues confront it. First of all, of course, is the question of where it will lead us in the analysis of English and other languages. Intensionality is probably still one of the most difficult aspects, although I have specificly addressed this issue from the point of view of TDS in another paper (Dahllöf [4]). The present implementation uses an HPSG-style grammar, which is an expensive thing from both a computational and an engineering perspective.

However, the surface-oriented, conceptually simple, and fine-grained nature of the TDS scheme invites experimentation with statistical annotation methods. For this reason, the TDS scheme is of potential relevance for applications such as information extraction, document classification, and machine translation. A heuristic system for TDS annotation would be more or less a necessity if we want to produce a sizeable TDS-based treebank. Such an implementation would also exemplify a new approach to semantic analysis and be of theoretical interest in its own right.

Another important issue concerning corpus annotation is the relation between semantics and other aspects of linguistic structure. We may be interested in, for instance, prosody, typography, grammar, coreference, rhetorical relations, and information structure. The semantic analysis should be in harmony with the annotation of these aspects and make is easy to explore whatever links there are between them and semantic facts. It seems to me that the best way to achieve this is to ensure that the analysis directly and firmly anchors all descriptive statements in audible and visible language use, and keeps everything is as simple as possible.

References

[1] Bod, R., Bonnema, R. and Scha, R. (1996) A Data-Oriented Approach to Semantic Interpretation. *Proceedings of the Workshop on Corpus-Oriented Semantic Analysis, ECAI-96*, Budapest.

[2] Copestake, A., Flickinger, D., Sag, I.A. and Pollard C. (1999) Minimal Recursion Semantics: An Introduction. Unpublished MS, CSLI, Stanford University. (URL: `http://www.cl.cam.ac.uk/~aac10/`).

[3] Copestake, A., Lascarides, A. and Flickinger, D. (2001) An Algebra for Semantic Construction in Constraint-based Grammars. In *Proceedings of the*

39th Annual Meeting of the Association for Computational Linguistics (ACL 2001) (Toulouse), Morgan Kaufmann Publishers, pp. 132–139.

[4] Dahllöf, M. (2002) Token Dependency Semantics and the Paratactic Analysis of Intensional Constructions. *Journal of Semantics* **19**, pp. 333–368.

[5] Dahllöf, M. (2003) An Implementation of Token Dependency Semantics for a Fragment of English'. *RUUL* (Reports from Uppsala University Department of Linguistics) **36**, pp. 101–227. (URL: `http://stp.ling.uu.se/~matsd/pub/ruul36.pdf`).

[6] Davidson, D. (1984) *Inquiries into Truth and Interpretation*. Oxford: Clarendon Press.

[7] Dowty, D. R., Wall, R. E. and Peters, S. (1981) *Introduction to Montague Semantics*. Dordrecht: Reidel.

[8] Gambäck, B. and Bos, J. (1998) Semantic-Head Based Resolution of Scopal Ambiguities. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics,* volume 1, pp. 433–437, Montreal.

[9] Kingsbury, P. and Palmer, M. (2002) From Treebank to PropBank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)* (Las Palmas, Spain), pp. 1989–1993.

[10] Kingsbury, P., Palmer, M. and Marcus, M. (2002) Adding Semantic Annotation to the Penn TreeBank. In *Proceedings of the Human Language Technology Conference,* San Diego, California.

[11] Lowe, J.B., Baker, C.F. and Fillmore, C.J. (1997) A frame-semantic approach to semantic annotation. In *Proceedings of the SIGLEX workshop "Tagging Text with Lexical Semantics: Why, What, and How?"*, Washington, D.C.

[12] Montague, R. (1974) *Formal Philosophy: Selected Papers of Richard Montague.* New Haven: Yale University Press.

[13] Sag, I.A. and Wasow, T. (1999) *Syntactic Theory: A Formal Introduction*. Stanford: CSLI Publications.

[14] Villalta, E. (2003) The Role of Context in the Resolution of Quantifier Scope Ambiguities. *Journal of Semantics* **20**, pp. 115–162.