Argument/Valency Structure in PropBank, LCS Database and Prague Dependency Treebank: A Comparative Pilot Study

Eva Hajičová and Ivona Kučerová

Center for Computational Linguistics
Faculty of Mathematics and Physics
Charles University, Prague
Malostranské náměstí 25
CZ-118 00 Prague
Czech Republic
{hajicova,kucerova}@ckl.mff.cuni.cz

Abstract

Three scenarios of corpora annotation on underlying syntactic level (PropBank, LCS Database and Prague Dependency Treebank) are compared as for the classification of values and structures assigned, and tentative steps of a mapping from PropBank to PDT are formulated.

1. Motivation and Objectives

The development of annotation schemes for large text corpora has entered a new stage of passing over from scenarios with tags concerning the phenomena ‚visible' or ‚transparent' on the surface shape of the sentence (be they of morphological or shallow syntactic character) to some kind of underlying structure of the sentence; also, not only intrasentential but also intersentential relations are paid attention to. The objective of this paper is to look (in Sect. 2) at three of the annotation schemes, namely the so-called PropBank (University of Pennsylvania) transforming the phrase-structure annotation of the Penn TreeBank into a predicate-argument structure, the UMD Lexical Database (LCS, University of Maryland), and the tectogrammatical tree structures of the Prague Dependency Treebank (PDT, Charles University), to compare the structures and values they work with (Sect. 3) and to attempt at a formulation of some basic steps of a possible mapping between these structures (Sect. 4). We believe that such a pilot study may contribute to the discussions on a possibility/ impossibility of formulating a ‚theory neutral' annotation scheme (Sect. 5).

2. A Brief Characterization of the Three Annotation Schemes

The predicate-argument structures of both the PropBank (Palmer et al., 2001) and the LCS Database (Dorr, 2001; Dorr & Olsen, 1997) are designed for English verbs (and their derivatives) and thus can find a solid motivation and support in the invaluable studies of Beth Levin (summarized most recently in Levin, 1993). The human annotators of the PropBank make use of the phrase-structure representations of sentences as present in the PennTreeBank and transform them into predicate-argument-structures (PAS). The lexical entry of a verb contains the basic form of the verb and a ‚frame' listing its possible complementations in terms of arguments differentiated by numbering (distinguishing five types of arguments) rather than by some semantically-based labels. In the lexicon, the roles assigned to individual arguments of the given verb are characterized by the so-called descriptions. The frames also contain information on relations between individual arguments. Thus, e.g. the entry for (one of the meanings of) the verb *keep* would be [arg0 [keep arg1 arg2]]: the verb can be modified by three arguments (as in *I kept the ice cream in the freezer*) and there are certain relationships between the arguments 1 and 2; in the frames, no distinction is made between obligatory and optional arguments.

The LCS database (LCS, 2001) works with a number of mnemotechnically labeled arguments such as *ag*, *th(eme)*, *goal*, *instr*, *loc*, *src(source)*, *ben(efactive)*, *info*, *purp*, *prop*, *perc*, *exp(eriencer)*; the frame for *keep* in the above quoted meaning would be *_ag_th,loc*, with a comma indicating that the given modification is optional; the other two are obligatory.

The Prague Dependency Treebank (PDT; see Hajič, 1998; Hajič et al., 2001; Straňáková-Lopatková & Žabokrtský, 2002) is designed as a complex annotation of Czech texts (taken from the Czech National Corpus); the underlying syntactic dependency relations (called functors) are captured in the so-called tectogrammatical tree structures (TGTS; see Hajičová et al., 2000). The set of functors comprises about 40 valency types (subclassified into (inner) participants and (free) modifications), each of which can be subcategorized into more subtle classes (expressing mainly the meanings of prepositions). Each verb entry in the lexicon is assigned a valency frame specifying which type of participant/ modification can be associated with the given verb; the valency frame also specifies which participant or modification (adjunct) is obligatory and which is optional with the given verb entry, which of them is deletable on the surface, which may or must function as a controller, and so on. Thus the valency frame for *keep* in the above quotes is ACT PAT LOC.

3. Comparison

3.1. Introduction

It is evident from the brief characterization of the three scenarios presented in the previous section that all of them work with (underlying) sentence structure in which the verb is the ‚head' and all its arguments or modifications are (direct) dependents of the verb. The differences lie in (i) the criteria distinguishing the types of arguments/modifications, (ii) the number of types distinguished, (iii) additional information in the lexical entries.

3.2. Criteria for the classification of arguments

As for the criteria distinguishing the different types of arguments, the PropBank avoids, to a large extent, semantic considerations: in principle, a one-argument verb has arg0 in its frame, a two-argument verb has arg0 and arg1, a three argument verb has arg0, arg1 and arg2, etc. Also an evidently semantically marked modification as *in the freezer* in the above quoted example *I kept the ice cream in the freezer* is labelled as arg2; it seems that those modifications that are (semantically, or better to say conceptually, from the perspective of the event structure identified by the given verb) obligatory (which is the case of the modification of location with the verb *keep* in the meaning exemplified by the quoted sentence) are always (?) classified as arguments (but not all arguments are obligatory). Another example of a similar sort is the modification *from his job* in *John earns 3K from his job*, which is labelled as arg3. On the other hand, some assignments seem to be guided by other than purely grammatical considerations: eg. *growl* has arg0 arg3=purpose (rendered by a nominal group with *for*) and all arguments are obligatory.

The LCS Database builds on semantically based considerations which is also reflected in the mnemotechnic shapes of the labels; the relation to the PropBank classes can be illustrated on a comparison of the above quoted examples: *in the freezer* in *I kept the ice cream in the freezer* is labelled directly as loc; *from his job* in *John earns 3K from his job* would be labelled as source (src). In the lexical entry of *earn* both the ag(entive) and th(eme) are obligatory, source being optional. The lexical entry of *growl* has ag(entive) and purp(ose), both obligatory.

In the Prague Dependency Treebank scenario, the shape of the structures as well as the repertory and classification of the types of modifications of the verbs is based on the theoretical framework of the Functional Generative Description formulated by Petr Sgall in the sixties as an alternative to Chomskyan transformational grammar, and developed further by the Prague research team of theoretical and computational linguistics (see e.g. Sgall et al., 1986). The first two arguments, though labeled by ‚semantically descriptive' tags ACT and PAT (Actor and Patient, respectively) correspond to the first and the second argument of a verb (cf. Tesnière's 1959 first and second *actant*), the other three arguments of the verb being then differentiated (in accordance with semantic considerations) as ADDR(essee), ORIG(in) or EFF(ect); these five functors belong to the set of participants (arguments) and are distinguished from (free) modifications (adjuncts) such as LOC(ative), several types of temporal and directional modifications etc. on the basis of two basic operational criteria: (i) can the given type of modification modify in principle every verb? (ii) can the given type of modification occur in the sentence more than once? (for a more detail, see Panevová, 1994).

3.3. Comparison of frames

A detailed comparison of the PropBank and LCS frames given in Dorr (2001) demonstrates that while PropBank's counterpart of LCS ag is always arg0, the LCS counterparts of arg0 are more subtly semantically differentiated: these may be exp(eriencer), goal, th(eme), loc(ative), possessive, prop(osition), src(source). Similar conclusions may be drawn from a comparison of frames including arg1 and arg2: prototypically, arg1 has a LCS counterpart of th(eme); however, this counterpart may be also labelled as goal, loc, exp, perc, src; arg2 has a prototypical counterpart in goal, but in specific cases, it may be an exp or theme. Dorr (2001) convincingly indicates that a mapping can be established between the LCS structures and those of PropBank taking into account the whole frames in which the given argument/ modification occurs, together with the prepositions accompanying some of the arguments/modifications.

A similar conclusion can be drawn from our tentative comparison between the LCS structures and PropBank representations on the one side and the TGTSs on the other. Prototypically, the functors ACT and PAT of TGTSs correspond to arg0 and arg1 of PropBank, respectively. This is not surprising because both scenarios do not rely merely on semantic considerations: the tags correspond to structural ‚slots' of the verbs rather than to the semantic ‚roles' of the arguments/participants. This is true about the first two slots (labelled ACT and PAT, respectively) of the frames in the TGTS framework; the criteria determining the functors of ADDR, ORIG and EFF (any of which can occupy the ‚third position') are of a more semantic nature, and the same holds, of course, about the (free) modifications. Thus, the functor assignment for the verb complementations in the above quoted examples would be as indicated in the brackets: *I* (ACT) *kept the ice cream* (PAT) *in the freezer* (LOC); *John* (ACT) *earns 3K* (PAT) *from his job* (ORIG). The frames of the verbs *keep* and *earn* in the lexicon would include ACT and PAT as obligatory participants for both verbs, and LOC as an obligatory modification with *keep* and ORIG as an optional participant with *earn*.

3.4. Remark on universality of the schemes

None of these systems claims to be independent of individual languages, although the attempt to model a kind of semantic level necessarily needs as a background some universal concepts. In the case of LCS the

proclaimed background is the concept of event structure, being tested syntactically. The PropBank system works basically with syntactic alternations worked out in Levin's classes non-trivially reflecting semantic content of a verb. In PDT the tectogrammatical structures should reflect the grammatical (underlying) structuring of the sentence.

There are other systems working with some kind of ‚deep syntactic' annotation, e.g. Berkeley's Framenet (based on a long-term research of underlying syntactic relations carried out by C. Fillmore), the Taiwan project MARVS, or the broadly conceived Italian project carried out in Pisa (N. Calzolari, A. Zampolli). These three frameworks are very closely related to the study of lexical semantics. Relevant data for a comparison of annotation schemes can be also found in WordNet. Another related framework is presented by the German project NEGRA, basically surface oriented, with which the newly produced subcorpus TIGER contains more information on lexical semantics. This project is also interesting for its claim to be ‚theory neutral'.

In a sense all systems work with a combination of some kind of ‚deep syntactic' structures and surface realization of the arguments of verbs. This may be the main reason why one possible danger can be hidden in semantic representation: the treatment of thematic roles seems to be similar for all concepts, but in the actual application the assignments may be radically different.

4. Basic Steps of a Possible Mapping between PropBank, LCS and PDT

4.1. From PropBank directly to PDT?

At the start, in the first step of the formulation of a mapping algorithm, we assumed that an argument structure of a verb might contain the same types of arguments in all three systems. There seemed to be only two real problems: how to find the right matching between the arguments and how to delimit the set of possible exceptions.

The work on this step clearly showed that the situation is much more complicated. Both in the PropBank and PDT there is a big difference between the first two arguments on the one hand, and the rest of the argument structure on the other. The first two arguments reflect the syntactic and semantic neutrality of structural cases. The mapping between PropBank and PDT can be easily based on the order of arguments, and all possible exceptions are deduced from the theoretical differences between the systems. The most visible difference is the treatment of the so-called ergative (and unergative) verbs, which is not a traditional category in Slavic linguistics and there is no linguistic consensus about its status for languages like Czech. But this controversy has a minimal impact on mapping. The correction of mapping can be based on an open list of ergative verbs, or on a closed list of PropBank ‚descriptions' (such as ‚logical subject' or ‚non-intentional theme').

Closed lists of ‚descriptions' seemed to be a good help when we considered a direct mapping between PropBank and PDT, although they would have to be combined with the order of arguments, and in some special cases also with information on surface realization. A closed list of descriptions is sufficient for mapping between arg0 and arg1 in PropBank and the corresponding functors in PDT. However, the attempt to base mapping of other arguments on descriptions has revealed deep differences between the two annotating systems. For example, the description ‚benefactive' in PropBank has several different counterparts in PDT. The strong uncertainty whether the found items are elements of the argument structure would not be a serious problem if we could use the information on surface alternations. For example, the difference between functors ADDR and BEN(efactive) can be tested by a possible surface realization with the preposition *to* vs. *for*. The verb *allocate* has only one surface realization, i.e. with *to*; the arg2 can then be mapped as an obligatory item with the functor ADDR. On the other side, the second argument of the verb *call* shows the surface alternation with *for*: *John called Mary a doctor* vs. *John called a doctor for Mary*; this argument should then be mapped as BEN. In this case we can find examples of alternations directly in the PropBank lexicon, but it is not trivial to do it automatically, and even more, the hand-made lexicon entries are not exhaustive.

The first attempts have thus shown that a direct mapping between PropBank and PDT is impossible nor can we generate different structures from it. This observation has led us to change our strategy for mapping and to try to combine pieces of information from all the three systems, i.e. from PropBank, LCS and PDT. Thus the envisaged procedure should in principle map the PropBank tags to the LCS ones, and from there it should proceed to the PDT tags (functors). Let us start with the discussion of the second step first.

4.2. From LCS to PDT: theta roles → functors

The system of assigning theta roles in LCS is quite similar to the system of assigning functors in PDT. This similarity is not so clearcut for the first two arguments, but mapping of them can be directly gained from PropBank. In the mapping rules below, on the left-hand side of the rule there is the LCS tag, on the right-hand side the PDT tags (functors). The examples are taken over from the PropBank or the LCS and the elements assigned the given tags are italicized.

(i) **Straightforward mappings** (one-to-one)

ag → ACT (*The hammer* broke the vase.)

exp → ACT (The internal *investigation* also criticized MiniScribe's auditors, Coopers&Lybrand, for allegedly ignoring numerous red flags.)

th → PAT (… that the government would abolish its golden *share* in Jaguar…)

pred → EFF (John turned *into a monkey*. We considered him *a fool*.)

perc → DIR3 (He looked *into the room*.)

loc → LOC (He lived *in France*.)

poss → PAT (This box carries *five eggs*.)

ben → BEN (John baked the cake *for Mary*. John baked *Mary* a cake.)

instr → MEANS (John broke the glass *with a stick*.)

purp → AIM (He studied *for exam*.)

manner → MANN (Insurers *typically* pass on a small percentage of the risks …)

mod-pred → CPR (She imagined him *as a prince*.)

(ii) **Complex cases** (one-to-many mappings)

scr → ACT or PAT (…*both men*.ACT exuded confidence and seemed to work well together.- John pitted the *cherries*.PAT)
The decision can be based on the PropBank descriptions.

info → PAT or EFF (We are advising a lot of our clients *to make*.PAT moves that make sense to them. - "*What matters is what advertisers will pay*".EFF said Newsweek's chairman …)
The decision can be based on the PropBank descriptions.

src → ORIG or DIR1 (The mark climbed to 77.70 yen *from 77.65 yen*.ORIG late Tuesday in New York. – Loki draws his name *from Norse mythology*.DIR1)
This distinction is crucial because of the uncertainty involved in the PDT frames; our tentative approximation is based on whether the arguments are obligatory (→ ORIG) or not (→DIR1); to make such a decision it is necessary to compare English frames with their Czech counterparts; mapping to Czech can be based on pointers to WordNet that are involved in LCS Data and in the Czech frame lexicon Vallex (Straňáková-Lopatková & Žabokrtský, 2002).

goal → ADDR or DIR3 or INTT (What this tells *us*.ADDR is that US trade law is working. – Arbitrage simply transfers his selling pressure from Chicago *to New York*.DIR3. John was trained by Mary *to correct*.INTT the treebank.)
The choice between ADDR and DIR3 can be based on specifications listed in LCS Data; the theta role specified as ‚human+' is preferred to be mapped as ADDR. Another indicator, though with a lower bias, is the obligatoriness of theta roles in LCS: the obligatory value is preferably connected with ADDR. Other hints can be gained from PropBank, e.g. the description ‚hearer' is always connected with ADDR. However, PropBank should be used only for smoothing rather than for direct mapping. The choice of INTT can be based only on the morphological information, because the only indicator for INTT is the infinitival particle *to*.

time → a tag of one of the temporal modifications (e.g. TWHEN; John ate *at nine*.) or PAT
The functor PAT is connected with the predicate *be*; the decision on a specific temporal tag can be made only according to the surface realization, i.e. to the preposition present in the LCS scheme.

mod-poss → SUBST or MEANS (He bought it *for $5*.SUBST – He loaded the cart *with hay*.MEANS)
The decision is triggered by the surface realization: e.g. the preposition *for* indicates SUBST(itution), preposition *with* indicates MEANS.

mod-loc → LOC or DIR or other spatial functor (She held the child *in her arms*.)
The decision on a concrete spatial tag can be made on the basis of the preposition, but this is not a very safe indication because many prepositions are used in several meanings.

4.3. PropBank → LCS: verb senses

The main problem for mapping, however, still remains, namely how to map word senses from PropBank to LCS. This mapping can be done with a set of rules with different weights. The highest weight is allocated on the basis of the membership to one of Levin's classes. The appurtenance to Levin's classes is not encoded in PropBank straightforwardly, but it can be deduced by means of the so-called ‚roleset names' assigned to individual verbs in the PropBank. A ‚roleset' is a set of verb senses with similar lexical meanings. For example, the roleset named ‚give' contains predicates (strictly speaking, one of their senses) *contribute*, *leave*, *pass*, *pass on*, *send*, *slip*, *submit* and *turn in*.

Though the rolesets need not be identical with Levin's classes, they reflect similar syntactic and semantic behavior of verbs. One roleset often covers several arguments and their specifications, i.e. their descriptions, on a more abstract level: e.g. the arg0 may be a ‚giver' or a ‚sender', depending on the particular verb belonging to the same roleset. If verbs of one roleset belong to a single Levin's class, this class can be used as a pointer to LCS. For example, the roleset named ‚announce' (with predicates *declare* and *profess*) belongs to the Levin's class 37.7 and this class used as the pointer to LCS returns the LCS frame (_ag_info,goal(to)); this frame can be directly used for the derivation of the PDT frame [ACT, PAT, ADDR].

	PropBank	**LCS**	**PDT**
‚**pass**'	arg0 arg1 arg2 roleset name = ‚give' [Levin'class 13.1]	_ag _th _goal (to)	ACT PAT ADDR
Remark: The mapping is based on the numbers of arguments (arg0 → ACT, arg1 → PAT), and on the intersection of Levin's classes (13.1: arg2 → goal (to) → ADDR).			
‚**note**'	arg0 arg1 [descr = ‚utterance'] arg2 roleset name = ‚say' [Levin'class 37.7]	_ag _info ‚goal (to)	ACT EFF ADDR
Remark: This mapping is based on the number of the first argument (arg0 → ACT), on descriptions (‚utterance': arg1 → EFF), and on the intersection of Levin's classes (37.7: arg2 → goal (to) → ADDR).			
‚**profess**'	arg0 arg1 [descr = ‚utterance'] arg2 arg3 roleset name = ‚announce' [Levin'class 37.7]	no sense based on the Levin's class 37.7	ACT EFF ADDR PAT
Remark: This mapping is based on the number of the first argument (arg0 → ACT) and on descriptions (‚utterance': arg1 → EFF). The mapping of the other arguments cannot be based on LCS Data, because there is no sense connected on intersection of Levin's classes (i.e. on the class 37.7), but the frame can be copied from the other verbs in the roleset named ‚announce' (i.e. *announce* and *declare*).			

Table 1: Examples of mapping between scenarios

A similar rule can be used also for more complicated cases. For example, the roleset named ‚cause to stop' contains two predicates, *end* and *finish*, belonging to a single Levin's class (55.1). This class can be used as the pointer to LCS in the following way: The relevant LCS frame is (_th,prop) and the derived PropBank frame is (arg1, arg2). However, the original PropBank frame is (arg0, arg1, arg2). When mapping to the PDT frame, its first two arguments can be derived by a direct mapping from the original PropBank frame (based on descriptions) so that arg0 → ACT and arg1 → PAT. The third argument can be derived from the last LCS argument, i.e. optional ‚prop', and can be rewritten as the functor MANN (e.g. *John celebrated his graduation prematurely*).

If a roleset contains only one verb sense, the mapping can be carried out straightforwardly. In case a roleset covers several senses of the same Levin's class, the automatically generated frames for PropBank, which are involved in LCS Data, can be used. These frames are not necessarily identical with the original PropBank frames, they often have a different number of arguments and they sometimes differ also in the value of the arguments. However, if they are identical (or strongly similar, i.e. they differ only in the number of arguments), they can be used as a control (checking) mechanism. The different number of arguments can be caused by different data seen in the corpora. For example, for the roleset ‚attach' a conjoined class only for the verbs *tie*, *link*, *attach* and *lock* can be found; the verb *join* is supposed to belong to a different class (22.4). However, for the sense assigned to this verb we can copy the frame of the verbs under the ‚attach' roleset.

In case the verb used as the name of a PropBank roleset is included among the verbs of a Levin's class, this class can be used as a pointer with a high preference and the mapping may be straightforward, under the condition that the original and the derived PropBank frames of the given verb are similar. If the verb used as the name of a PropBank roles is not among the verbs of the given Levin's class, then other verb classes are to be checked as for the identity or at least similarity of the two PropBank frames. This strategy is necessary for rolesets that include only a single predicate and the name of the roleset is other than the verb itself (i.e. the name of the roleset is descriptive).

If none of the strategies described above can be applied, the only way how to achieve an automatic mapping between PropBank and PDT is to base the mapping on a combination of the surface realization and the order of arguments. To give just an example: The counterparts of arg0 and arg1 are derived according to the PropBank descriptions; arg2 with the preposition *to* is rewritten as ADDR; if arg2 is followed only by a single argument, this argument is EFF; if the counterpart of arg2 with the preposition *from* is followed by arg3 with preposition *to*, then arg2 is rewritten as ORIG.

The system of mapping is illustrated by several examples in Table 1.

5. Conclusion

Our original aim was twofold: first, we were curious how different frameworks annotating some kind of deep (underlying) syntactic level compare with each other, and, second, we had in mind a practical application, namely a machine translation project the modules of which would be ‚machine-learned' based on syntactically annotated parallel corpora. For that purpose, we want to use the PDT scenario; to make our task simpler, we intend to make use also of the texts gathered in the PennTreeBank and their annotation. Thus, the task was to gain a direct mapping between PropBank and PDT as two annotating systems, and to use the following mapping on LCS Data as a semantic and checking source. The investigation of three different systems, i.e. LCS Data, PropBank and PDT has led us to a preliminary strong hypothesis of a possible direct mapping between these systems. This idea was postulated on the basis of a virtual similarity of these scenarios and it presupposed their closeness in the classification of arguments (if not even their ‚theory neutral' properties). However, as we have tried to demonstrate in our contribution, the task had to be reformulated as a two-stage mapping with the use of additional information present in the three treebanks.

We believe that such a pilot study may also contribute to the discussions on a possibility/impossibility of formulating a ‚theory neutral' syntactic annotation scheme. The idea of a theory neutral annotation scenario seems to us an unrealistic (though understandable and wishful) goal: it is hardly possible to imagine a classification of such a complex subsystem of language as the syntactic relations are, without a theoretical background; moreover, the languages of the annotated texts are of different types, and the theoretical frameworks the authors of the schemes are used to work with differ in the ‚depth' or abstractness of the classification of the syntactic relations. However, as we have tried to indicate in our contribution, the different annotation schemes seem to be translatable if the distinctions made in them are stated as explicitly as possible, with the use of operational criteria, and supported by larger sentential contexts. The third condition is made realistic by very large text corpora being available electronically; making the first two conditions a realistic goal is fully in the hands of the designers of the schemes.

6. Acknowledgement

The research reported on in this paper has been carried out under the project MŠMT LN00A063.

7. References

Dorr, B.J. (2001). LCS Verb Database, Online Software Database of Lexical Conceptual Structures and Documentation, UMCP. http://www.umiacs.umd.edu/~bonnie/LCS_Database_ Documentation.html"

Dorr, B.J. & Olsen B.M. (1997). Aspectual Modifications to a LCS Database for NLP Applications. Technical Report: LAMP-TR-007, UMIACS-TR-97-21, CS-TR-3763, University of Maryland, College Park. http://lamp.cfar.umd.edu/Media/Publications/Papers/LAMP_007/LAMP_007.pdf

Hajič, J. (1998). Building a Syntactically Annotated Corpus: The Prague Dependency Treebank. In E. Hajičová (Ed.): *Issues of Valency and Meaning. Studies in Honor of Jarmila Panevová* (pp.106--133). Prague: Karolinum, Charles University Press.

Hajič, J. et al. (2001). The Prague Dependency Treebank. CDROM. Linguistic Data Consortium, Univ. of Pennsylvannia, PA. Cat. #LDC2001T10. ISBN 1-58563-212-0.

Hajičová, E. et al. (2000). A Manual for Tectogrammatic Tagging of the Prague Dependency Treebank. UFAL/CKL Technical Report TR-2000-09, Czech Republic: Charles University.

LCS (2001). http://www.umiacs.umd.edu/ ~bonnie/LCS_Database_Documentation.html

Levin, B. (1993). English Verb Classes and Alternations: A Preliminary Investigation. Chicago and London: The University of Chicago Press.

NEGRA (2002). http://www.coli.uni-sb.de/sfb378/negra-corpus/

Palmer, M., Rosenzweig, J. & Cotton, S. (2001): Automatic Predicate Argument Analysis of the Penn TreeBank. In J. Allan (Ed.): *Proceedings of HLT 2001, First International Conference on Human Language Technology Research*. San Francisco: Morgan Kaufamm. http://hlt2001.org/papers/hlt2001-10.pdf

Panevová, J. (1994). Valency Frames and the Meaning of the Sentence. In Ph.L. Luelsdorff (Ed.): *The Prague School of Structural and Functional Linguistics* (pp. 223--243). Linguistic and Literary Studies in Eastern Europe 41, Amsterdam-Philadelphia: John Benjamins.

Sgall P., Hajičová E. & Panevová J. (1986). The Meaning of the Sentence in Its Semantic and Pragmatic Aspects. Dordrecht: Reidel, Prague: Academia.

Straňáková-Lopatková, M. & Žabokrtský, Z. (2002). Valency Dictionary of Czech Verbs: Complex Tectogrammatical Annotation. *Proceedings of Third International Conference on Language Resources and Evaluation.* Las Palmas, Spain.

Tesnière, L. (1959). Élements de syntaxe structurale. Paris.