HPSG-Based Annotation Scheme for Corpora Development and Parsing Evaluation[*]

Kiril Iv. Simov
BulTreeBank Project
http://www.BulTreeBank.org
Linguistic Modelling Laboratory Bulgarian Academy of Sciences
Acad. G. Bonchev St. 25A, 1113 Sofia, Bulgaria
`kivs@bultreebank.org`

Abstract

This paper proposes a formal framework for development and exploitation of a corpus, based on the HPSG linguistic theory. The formal representation of the annotation scheme facilitates the annotation process and ensures the quality of the corpus and its usage in different application scenarios. Also, evaluation over HPSG annotation scheme is discussed. The advantages of the approach are presented in comparison with other related works.

Keywords: Treebank, HPSG, Parsing Evaluation, Dynamic Resources

1 Introduction

This paper proposes a strategy for the construction of a corpus, based on the HPSG (Head-driven Phrase Structure Grammar) linguistic theory (see (Pollard and Sag, 1994)). The annotation scheme for the corpus is formally defined in a formalism for HPSG and reflects the developments in the theory. Thus our approach tries to incorporate both - the language theory and the underlying formal assumptions. It has the following important advantages:

- By imposing constraints over the HPSG-derived annotation scheme, the annotation process becomes more efficient;
- It supports the definition of validation theories, which encode more consistently the otherwise informal annotation guidelines.
- When the annotation scheme is changed at some later stage of the corpus development, the previously annotated sentences can be reclassified with respect to the new scheme on the basis of: (1) the information that has already been encoded, and (2) the minimal human intervention.
- The formalism allows for different levels of abstraction over the data in the corpus. This can be very useful for the application possibilities of the corpus. For example, different learning mechanisms might rely upon different types of information.
- Also the inference mechanisms can be used for evaluating parsers with respect to such a corpus. The data can be tuned to a particular task and thus used as a standard for the task.

As a result, the intended corpus becomes an electronic linguistic resource of a high quality and, consequently - a good candidate for a test suite.

The work reported here has been developed within the BulTreeBank project (Simov, Popova and Osenova, 2002), which started in February 2001. The main goal of the project being the construction of an HPSG-based treebank for Bulgarian. This goal presupposes the choice of the linguistic theory and its adequate formalization. In our case it is the HPSG theory and the SRL grammar formalism that we rely upon. The choice is motivated by the fact that HPSG and SRL meet the requirements for a consistent representation of the linguistic knowledge within the treebank. Of course, there exist other formalisms for the HPSG theory, but the comparison with them is beyond the scope of this paper. Note that the ideas presented here can be worked out for other grammar formalisms as well as for other grammar theories.

We would like to discuss briefly some questions often raised with respect to the development of a treebank[1], namely, the role of the linguistic theory, and the relation between a certain grammar and the treebank. Concerning the role of the linguistic theory, we believe that the notion of the ‘theory independency’ is impossible in case of detailed linguistic description, if at all. In our view, the annotation scheme of each treebank always involves some linguistic theory, especially when taking specific decisions on the representation of the linguistic facts and their interrelation.

[*] The work reported here is done within the BulTreeBank project. The project is funded by the Volkswagen Stiftung, Federal Republic of Germany under the Programme *“Cooperation with Natural and Engineering Scientists in Central and Eastern Europe”* contract I/76 887. I would like to thank Petya Osenova for the constant help during the work on the paper and to the three anonymous reviewers for their comments. All the remaining errors are mine.

[1] They were posed, also, by the reviewers of the paper.

The connection between the grammar and treebank development is bidirectional. On one hand, a recent survey on treebanks (Abeillé, 2003) shows that most of the treebanks are grammar-based in the following sense: they use a pre-defined grammar for the production of all the possible sentence analyses, which later are manually corrected. The main advantage of such a treebank is the additional knowledge, entered by the annotator in the post parsing phase. On the other hand, the constructed treebanks can be used for grammar extraction and specialization. How a treebank of our kind can be exploited for such purposes, is described elsewhere: (Simov et al., 2002; Simov, 2002). The work reported here is in close connection with our previous investigations.

The structure of the paper is as follows: In Section 2 the HPSG Language model is presented. Section 3 describes in detail the formalism that is employed for the representation of the HPSG grammar, the corpus and the annotation scheme. Section 4 demonstrates how this formalism can be used for facilitating the annotation process. Section 5 focuses on the reclassification algorithm. In Section 6 the evaluation process is described. Section 7 discusses related works. The last section concludes the paper.

2 The HPSG Language Model

In this section we present the general language model, accepted within HPSG. HPSG is a lexicalist linguistic theory, in which the linguistic objects are represented via feature structures. It includes: a linguistic ontology (sort hierarchy) and grammar principles (constraints over the sort hierarchy). The sort hierarchy represents the main types of linguistic objects and their basic characteristics. The principles impose restrictions on the objects and thus predict the well-formed phrases. A basic mechanism for ensuring the right sharing of information among the various parts of the linguistic objects is the *co-reference*. The main linguistic object in HPSG is of sort *sign* (whose subsorts are *word* and *phrase*). It is a complex entity that is assigned two features: PHON (string of phonemes) and SYNSEM (syntactic and semantic characteristics). Further within the attribute SYNSEM there are three important attributes: CATEGORY (which encodes the syntactic information), CONTENT (which encodes the semantic information) and CONTEXT (which encodes the pragmatic information). The constituent structure is encoded for each phrase via the attribute DTRS. Assigning different values to this feature, HPSG theory distinguishes between (at least) the following types of phrases – *headed-phrase* and *non-headed-phrase*. The first kind is additionally divided into *head-complement*, *head-subject*, *head-adjunct*, and *head-filler*. The *head-filler* phrases account for the cases of unbounded dependency. The *non-headed-phrase* is used for dealing with coordination phrases. The current hierarchy of phrases is presented in the following sort hierarchy:

```
sign
    PHON : phonlist
    SYNSEM : synsem
  word
  phrase
      DTRS : dtrs
    headed-phrase
        head-complement
        head-subject
        head-adjunct
        head-filler
    non-headed-phrase
```

The linearization of the constituents in HPSG is separated from the constituent structure and in this way the theory allows for different orders of the same constituent structure and discontinuous realization of the constituents. This separation ensures the representation of the grammatical relations within the constituent structure. The actual realization of the head dependents is governed by a set of immediate dominance schemata. The realization of the dependents follows the sequence: **complements** → **subject** → **adjuncts**. The actual number and kind of dependents is determined by the lexical head within each phrase. The structure of the linguistic objects in HPSG makes its language model very appropriate for encoding the information in a treebank. In fact, we could consider it as a hybrid approach to representation of syntactic information because it represents the constituent structures and grammatical relations at the same time.

An example of a tree is presented in Fig. 1. The tree consists of four types of nodes and two types of arcs. The leaves in the tree correspond to the words and punctuation. The circles correspond to the sign objects in HPSG, the labels inside them determine the subsort of the sign and its constituent structure - lexical (N,V,Prep,Pron,A), head-complement (VPC,PP), head-subject (VPS), head-adjunct (NPA,VPA) and the category of the sign. The rectangles correspond to some additional properties of the signs below them. Here three kinds of such properties are shown: the root node of the sentence [S], the da-clause [CLDA] and a representation of the unexpressed subject of the

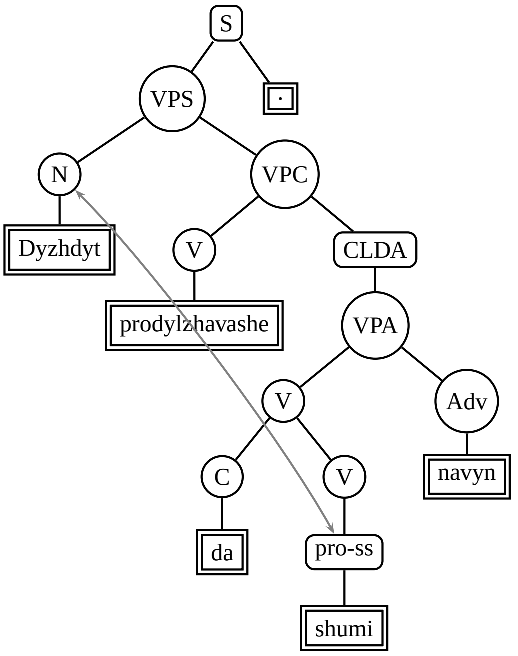

Figure 1: A schematic tree for the sentence: **Dyzhdyt prodylzhavshe da shumi navyn.** (The rain continued babbling outside.)

da-clause [pro-ss]. The immediate dominance relation between the signs is given by the structure of the tree itself. We allow for crossing branches (not presented here). The co-references among the indices of the signs are given by additional arcs between the nodes of the tree. Here we have one such link which connects the unexpressed subject of the CLDA clause with the expressed one of the main verb. For more explanations on the current version of the annotation scheme see (Simov and Osenova, 2003).

3 Formalism for HPSG

In this section we present a logical formalism for HPSG. Then a normal form for a finite theory is defined as a set of feature graphs. In (Simov, 2001; Simov et al., 2002; Simov, 2002) shows that this normal form is suitable for the representation of an HPSG corpus and an HPSG grammar (see also (King and Simov, 1998)). In the paper we extend the idea further viewing these graphs as a representation of an HPSG annotation scheme as well. Here we shortly present the syntax of the logic (SRL). For full description see (King, 1989).

$\Sigma = \langle \mathcal{S}, \mathcal{F}, \mathcal{A} \rangle$ is a finite **SRL signature** iff $\mathcal{S}$ is a finite set of *species*, $\mathcal{F}$ is a set of *features*, and $\mathcal{A} : \mathcal{S} \times \mathcal{F} \to Pow(\mathcal{S})$ is an *appropriateness function*.

τ is a **term** iff τ is a member of the smallest set $\mathcal{T}$ such that (1) $: \in \mathcal{T}$, and (2) for each $\phi \in \mathcal{F}$ and each $\tau \in \mathcal{T}$, $\tau\phi \in \mathcal{T}$. δ is a **description** iff δ is a member of the smallest set $\mathcal{D}$ such that (1) for each $\sigma \in \mathcal{S}$ and for each $\tau \in \mathcal{T}$, $\tau \sim \sigma \in \mathcal{D}$, (2) for each $\tau_1 \in \mathcal{T}$ and $\tau_2 \in \mathcal{T}$, $\tau_1 \approx \tau_2 \in \mathcal{D}$ and $\tau_1 \not\approx \tau_2 \in \mathcal{D}$, (3) for each $\delta \in \mathcal{D}$, $\neg\delta \in \mathcal{D}$, (4) for each $\delta_1 \in \mathcal{D}$ and $\delta_2 \in \mathcal{D}$, $[\delta_1 \wedge \delta_2] \in \mathcal{D}$, $[\delta_1 \vee \delta_2] \in \mathcal{D}$, and $[\delta_1 \to \delta_2] \in \mathcal{D}$. Each subset $\theta \subseteq \mathcal{D}$ is an **SRL theory**.

An HPSG grammar $\Gamma = \langle \Sigma, \theta \rangle$ in SRL consists of: (1) a signature Σ, which gives the ontology of entities that exist in the universe and the appropriateness conditions on them, and (2) a theory θ, which gives the restrictions upon these entities. We represent grammars and corresponding sentence analyses in a normal form based on feature graphs.

Let $\Sigma = \langle \mathcal{S}, \mathcal{F}, \mathcal{A} \rangle$ be a finite signature. A **feature graph** with respect to Σ is a directed, connected and rooted graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{V}, \rho, \mathcal{S} \rangle$ such that: (1) $\mathcal{N}$ is a set of **nodes**, (2) $\mathcal{V} : \mathcal{N} \times \mathcal{F} \to \mathcal{N}$ is a partial **arc function**, (3) ρ is a **root node**, (4) $\mathcal{S} : \mathcal{N} \to \mathcal{S}$ is a total **species assignment function**, and (5) for each ν_1,$\nu_2 \in \mathcal{N}$ and each $\phi \in \mathcal{F}$ such that $\mathcal{V}\langle \nu_1, \phi \rangle \downarrow$ and $\mathcal{V}\langle \nu_1, \phi \rangle = \nu_2$, then $\mathcal{S}\langle \nu_2 \rangle \in \mathcal{A}\langle \mathcal{S}\langle \nu_1 \rangle, \phi \rangle$. We say that the feature graph $\mathcal{G}$ is **finite** if and only if the set of nodes is finite. A feature graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{V}, \rho, \mathcal{S} \rangle$ such that for each node $\nu \in \mathcal{N}$ and each feature $\phi \in \mathcal{F}$ if $\mathcal{A}\langle \mathcal{S}\langle \nu \rangle, \phi \rangle \downarrow$ then $\mathcal{V}\langle \nu, \phi \rangle \downarrow$ is called a **complete feature graph**. For each two graphs $\mathcal{G}_1 = \langle \mathcal{N}_1, \mathcal{V}_1, \rho_1, \mathcal{S}_1 \rangle$ and $\mathcal{G}_2 = \langle \mathcal{N}_2, \mathcal{V}_2, \rho_2, \mathcal{S}_2 \rangle$ we say that graph $\mathcal{G}_1$ **subsumes** graph $\mathcal{G}_2$ ($\mathcal{G}_2 \sqsubseteq \mathcal{G}_1$) iff there is an *isomorphism* $\gamma : \mathcal{N}_1 \to \mathcal{N}_2'$, $\mathcal{N}_2' \subseteq \mathcal{N}_2$, such that (1) $\gamma(\rho_1) = \rho_2$, (2) for each $\nu, \nu' \in \mathcal{N}_1$ and each feature ϕ, $\mathcal{V}_1\langle \nu, \phi \rangle = \nu'$ iff $\mathcal{V}_2\langle \gamma(\nu), \phi \rangle = \gamma(\nu')$, and (3) for each $\nu \in \mathcal{N}_1$, $\mathcal{S}_1\langle \nu \rangle = \mathcal{S}_2\langle \gamma(\nu) \rangle$. For each two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ if $\mathcal{G}_2 \sqsubseteq \mathcal{G}_1$ and $\mathcal{G}_1 \sqsubseteq \mathcal{G}_2$ we say that $\mathcal{G}_1$ and $\mathcal{G}_2$ are **equivalent.**

For finite feature graphs, we could define a translation into SRL descriptions using the correspondences between paths in the graph and terms. Thus we can interpret each finite feature graph as a description in SRL. Using the set of all finite feature graphs that subsume a given infinite feature graph, we can also define the interpretation of each infinite feature graph. Thus we can speak about satisfiable graphs. For them there exists an interpretation in which they denote non-empty set of objects. Moreover, we can define a correspondence between the finite SRL theories and the feature graphs. This representation of the theory as a set of graphs has the following very important properties:

1. each graph $\mathcal{G}$ in the set of graphs is satisfiable (for some interpretation the graph $\mathcal{G}$ denotes

some objects in the interpretation), and

2. each two graphs $\mathcal{G}_1$, $\mathcal{G}_2$ in the set have disjoint denotations (for each interpretation there is no object in the interpretation that is denoted by the two graphs).

These properties of the set of graphs theory representation allows for classification of linguistic objects with respect to the graphs. We are going to use such an algorithm for the tasks connected to the creation and usage of the corpus. Also, an inference procedure over feature graphs is developed as composition of graphs. The procedure reflects the semantics of the corresponding SRL theory.

We aim at proving out that feature graphs are adequate for the following important scenarios: (1) **Representation of an HPSG grammar.** The construction of a graph representation of a finite theory demonstrates that using feature graphs as grammar representation does not impose any restrictions over the class of possible finite grammars in SRL. (2) **Representation of an HPSG corpus.** Each sentence in the corpus is represented as a complete feature graph. One can easily establish a correspondence between the elements of the strong generative capacity of an HPSG grammar and the complete feature graphs. Thus complete feature graphs naturally become a good representation for an HPSG corpus. (3) **Representation of the annotation scheme.** We assume that an annotation scheme over the HPSG sort hierarchy can be considered a grammar. The feature graphs of such an annotation scheme will be constrained by the lexicon, which is available to the annotators, by the principles, which are stated as a theory, and by the input sentences. As a result, all the constraints that follow logically from the above sources of information can be effectively exploited during the annotation process.

4 Corpus Annotation

The corpus annotation within this framework is based on the idea of parse selection from a number of automatically constructed sentence parses. The parses are constructed by the inference mechanism using the graph representation of the annotation scheme and graph encoding of the sentence. This approach to corpora development is well known as *grammar-based corpus annotation*. See (Dipper, 2000) for an example among others. Our approach differs in the formal mechanisms that are incorporated within the implementation. The assumptions, which the annotation process is based on, are listed below:

- The annotation scheme is defined as a set of graphs. Thus each sentence annotation has to be consistent with respect to the logical properties of the annotation scheme. Nevertheless, the annotator is not constrained too much, because the annotation scheme is still general and therefore, it will overgenerate massively.

- The annotator cannot simultaneously observe all the parses, generated by the annotation scheme. Thus he/she has to make a choice relying only on the partial information as a prompt for the sentence true analysis. Thus annotators' work is incremental.

- The information, added by the user during the annotation process, is propagated further. The propagation reduces the number of the possible choices in other places of the sentence analysis.

Thus, the overall annotation process is organized as follows:

1. First, the selected sentence is processed partially. This processing is compatible with the HPSG sort hierarchy and comprises: morphological analysis, disambiguation and non-recursive partial parsing. As a result, the complexity of the following steps is reduced. Note that at this point the sentences receive a unique partial analysis.

2. The result from the previous step is encoded as feature graph and it is further processed by an HPSG processor with the help of the described annotation scheme. The result is a set of complete feature graphs.

3. The selection of the correct analysis is considered *a classification* of the partial description of the true sentence analysis with respect to the set of complete feature graphs, produced in the previous step. The classification starts with the common for all complete graphs information. This information contains all the partial analyses from the first step, because the HPSG processor operates monotonically and thus, it cannot delete information. On the basis of the differences between the complete graphs *an index* over them is created. This index supports the propagation of the information, added by the annotator. When the user adds enough information, the partial analysis can be extended to exactly one of the complete graphs. If the sentence allows more than one analysis, the annotator has to classify it more than once.

In the rest of the section we present the index and describe its contribution to the process of the classification of the partial sentence analysis with respect to all sentence analyses. The idea behind the classification is that the annotator states the new information about the analysis as elementary descriptions of the relevant graph. The elementary descriptions are of the following kinds: $\pi \sim \sigma$ (the path π is defined in the graph and the species of the end node is σ), $\pi \not\sim \sigma$ (the path π is defined in the graph and the species of the end node is not σ), $\pi_1 \approx \pi_2$ (the paths π_1 and π_2 are defined in the graph and they share their end nodes), and $\pi_1 \not\approx \pi_2$ (the paths π_1 and π_2 are defined in the graph and they have different end nodes)[2].

Here are some examples of elementary descriptions that the user can supply: "the phrase is of type *head-complement*", "the verbal adjunct is not a secondary predication", "the unexpressed subjects of two relative clauses are the same". If, for example, the sentence contains also reflexive pronouns, bound to the unexpressed subject in one of the relative clauses, the last claim will automatically add a binding link from this pronoun to the unexpressed subject of the other relative clause.

In (King and Simov, 1998) we have shown that for a set of graphs, representing a theory, there is a set of elementary descriptions, such that each description in the set discriminates over the set of graphs. Thus, it is true for at least one graph and it is false for at least one graph. Using the last properties one can construct an index over the set of graphs. The index is a tree, such that the nodes of the tree are marked with elementary descriptions and the edges of the tree are marked with the truth values: *true* or *false*. And the descriptions are chosen in such a way that each path from the root of the tree to some of the leaves of the tree determines exactly one graph in the initial set of graphs. The descriptions, presented in the index, can be chosen on the basis of the graphs in the set.

In order to use such indices for facilitating the annotation process, we encode all possible indices over the complete graphs, returned by the HPSG processor. This work is being done incrementally over the differences of the graphs and thus the indices share some of their parts. The index is not a tree in this case, but rather a forest. This step is necessary for annotators' convenience, because it is not clear at the beginning, which information will be easy to be provided manually.

It can be proved that for finite set of graphs there exists a finite index. Stating one of the elementary descriptions in the index, the annotator always reduces the number of the graphs that are presented by this description. Providing several descriptions, the annotator arrives at exactly one graph from the set. Thus, the classification is performed in the following way:

1. At the beginning all the nodes in the index are available to be chosen and the annotator has the possibility to state any of the elementary descriptions in the index.

2. The annotator decides on an elementary description about the sentence from the set of the allowed descriptions.

3. The elementary description is found in the index and this operation reduces the number of the possible graphs. It also means that some of the elementary descriptions in the index are not eligible any more, because they will contradict the selected description.

4. If the set of the possible graphs is a singleton (has only one member), then this graph is a result from the classification. If the set contains more than one graphs, then the algorithm goes to point 2 and offers the annotator to make a new choice of an allowed elementary description.

The chosen graph is in fact the analysis of the sentence. It is important to say that this algorithm of classification works not only over a set of complete graphs, but also over graph representations of finite SRL theories.

An additional facility for the annotator is the possibility for him/her to provide larger descriptions in one step. Such descriptions represent the linguistically motivated characteristics of the sentence. Larger descriptions can be considered macros. For example, macroses are the constituent labels like VPS for verbal head-subject phrase, NPA for noun head-adjunct phrase etc.

As a speed measure of the annotation we consider all the necessary selections made by the annotator in his/her steps to the complete analysis. The number of the selections are in the worst case equal to the number of all analyses, produced by the HPSG grammar. This can happen when the annotator rules out exactly one analysis per choice. The average number of selections is a logarithm from the number of the analyses. An important advantage of this selection-analysis-approach is that the annotator works locally.

[2] The description $\pi \approx \pi$ states that the path π is defined in the graph. The description $\pi \not\approx \pi$ states that the path π is not defined in the graph.

Thus the number of parameters necessary to be considered simultaneously is minimized.

5 Reclassification

The need for a reclassification of already classified linguistic objects arises in connection with the following problems and tasks:

- Changes in the target linguistic description of the elements in the corpus;
- New tasks, for which the corpus might be adjusted;
- New developments in the linguistic theory;
- Misleading decisions, taken during the design phase of the corpus development.

In each of these cases, the development of a new annotation scheme is necessary. The problems concerning such a step are well known: What about the corpus built up to now? How to use it in the new circumstances and at minimal costs? Here we offer an algorithm for reclassification within our formalism for HPSG.

There are two possible scenarios for the application of the reclassification to an already created corpus:

- The first holds when the changes in the annotation scheme are relatively small. For instance, addition of new features, new sorts or new principles to the initial HPSG grammar.
- In the second case there is a substantial change in the annotation scheme. For example, complete substitution of the sort hierarchy parts with new ones.

Of course, there are not clear boundaries between the two kinds of changes.

Let Σ_{old} and Σ_{new} be two signatures and let A_{old} be the annotation scheme constructed on the basis of Σ_{old} and A_{new} be the annotation scheme constructed on the basis of Σ_{new}. The idea of reclassification is based on the notion of the *correspondence rules* between descriptions with respect to the old and to the new annotation schemes. The general format of these rules is:

$$\delta_{old} \Rightarrow \delta_{new}$$

where the δ_{old} is a description with respect to Σ_{old} and δ_{new} is a description with respect to Σ_{new}. The meaning of such rules is: for each model $\mathcal{I}_{old}$ of A_{old} such that the description δ_{old} is satisfiable in it, there exists a model $\mathcal{I}_{new}$ of A_{new} such that the description δ_{new} is satisfiable in it. Thus we consider the correspondence rules as rules for transferring knowledge between the two annotation schemes.

Then the algorithm for reclassification works in the following way:

1. Let Σ_{old} and Σ_{new} be two signatures and let A_{old} be the annotation scheme constructed on the basis of Σ_{old} and A_{new} be the annotation scheme constructed on the basis of Σ_{new}. Let CR is a set of correspondence rules.

2. Let $\mathcal{G}_{old}$ be a graph with respect to A_{old} for the sentence S. Let $\{\mathcal{G}^1_{new}, \ldots, \mathcal{G}^n_{new}\}$ are the candidate analyses for the sentence S with respect to the new annotation scheme A_{new}.

3. The algorithm constructs the set ED_{old} of all descriptions δ_{old} such that there exists a correspondence rule $\delta_{old} \Rightarrow \delta_{new} \in CR$ and $\mathcal{G}_{old}$ is in the denotation of δ_{old} for each interpretation of A_{old} that satisfies $\mathcal{G}_{old}$. Thus ED_{old} contains all the descriptions on the left side of the correspondence rules that are true for the graph.

4. Then the algorithm constructs the set ED_{new} of descriptions δ_{new} such that there exists a correspondence rule $\delta_{old} \Rightarrow \delta_{new} \in CR$ and $\delta_{old} \in ED_{old}$. We consider the set ED_{new} to be the transferred knowledge from the old annotation of the sentence S to the new annotation.

5. Then the algorithm uses the set ED_{new} and the index for the new potential analyses for the sentence S in order to find the minimal number of graphs from the set $\{\mathcal{G}^1_{new}, \ldots, \mathcal{G}^n_{new}\}$, which satisfies all the descriptions in ED_{new}.

The result of this algorithm is a set of graphs. If the set is empty, it means that the transferred knowledge is in contradiction with the new annotation scheme and cannot be really used. In this case the developers of the corpus have to reconsider the correspondence rules. If the set is a singleton, then it equals the analysis of the sentence with respect to the new annotation scheme. If the set contains more than one element, then the old analysis does not contain enough information for a unique classification of the sentence with respect to the new annotation scheme and some human intervention will be necessary. In fact we expect the last point to be the majority of the cases. Nevertheless the reclassification process will be helpful in this case also because it will reduce the number of the candidate analyses.

The two scenarios mentioned above differ from each other mainly on the basis of the complexity of the correspondence rules. In the first case one can state that each old description that is eligible with respect to the new scheme is mapped on itself. The second case will require more complicated rules.

6 Evaluation over an HPSG Annotation Scheme

Every corpus, which is constructed with respect to this formalism, offers various opportunities for evaluation of parsing systems. It is evoked by several factors. First, HPSG as a theory describes the linguistic objects by using both mechanisms for the representation of the syntactic information: constituent structure and head-dependent structure. Hence, one could rely on the most of the current evaluation metrics like: PARSEVAL precision and recall over bracketing and the mean number of overlapping brackets (Harrison et al., 1991), on evaluations focusing on grammatical relations as in (Carroll et al., 2003), or on dependency relations in (Kübler and Hinrichs, 2001) and (Lin, 2003). Additionally, such a corpus provides mechanisms for mixed evaluation schemes where both of these inventories can be used.

Another advantage of such a corpus is the high granularity of the information presented in it. This is a pre-requisite for the definition of different levels of degree where the evaluation process can take place. Note that it supports multi-level evaluation processes rather than mono-level ones. Recall one important fact is that HPSG-based analyses subsume the constituent structure as well as the head-dependent structure of the elements in the sentence. For example, one can work on the level of bracketing, but also she/he can view the constituents types as defined by the grammatical features of the head. Thus the two popular evaluation approaches can be easily implemented over the same corpus, i.e. either the bracketing precision and recall and bracketing overlap measures, or the grammar relations measures. One can even combine them in one measure parameter specific for certain evaluation requirements.

In order to achieve this one has to define a new annotation scheme, which reflects the evaluation task. Then it is necessary an appropriate set of correspondence rules to be defined. Afterwards the corpus is reclassified with respect to the new annotation scheme. In most cases this process will be a simplification of the information that is already in the corpus, and human intervention would not be necessary. For instance, one can keep only the information about head-dependents and delete all the information about the constituent structure. In this case a dependency-like evaluation can be implemented. One important point here is that the deletion of information is not exactly transformation of the graphs that already exist in the corpus. In fact, this is a construction of a new corpus using the information that is stored in the old one. There is no need the graphs in the new corpus to be isomorphic to subgraphs in the old one. As a very simple example we can consider the transformation of a deep adjunct attachment (one at a time) into a flat adjunct attachment (all at once).

Another possibility is the context dependent evaluation. Generally, this means to mix several evaluation approaches depending on the linguistic information in the sentence analyses. For example, consider the case when the evaluation aims at determining the right argument recognition for the verbs, but not for the prepositions. Then one can require all NPs with attached to them PPs to be transformed into some flat structured NPs, and keep the PPs only when they are arguments of the verbs. This can be implemented again via reclassification, because it is context sensitive.

7 Related Works and Discussion

There are several existing works related to ours. Using graphs for representation of corpora data is presented in a number of papers of Steven Bird and co-workers (see (Cotton and Bird, 2002) for applying their format to treebanks). The main difference between their approach and our work is that their graphs are defined purely in a graph-theoretical manner with some additions related to corpus practice (mainly speech corpora). This way of definition of annotation graphs requires some additional work on facilities for their manipulation like operations for transformation, querying. Also some logical formalism for annotation graphs is necessary in order to ensure the consistency of the represented linguistic information and the result from the operations over them. In comparison, our feature graphs are directly related to well established logical formalism which ensures the necessary functionality for their manipulation. Also the expressive power of feature graphs seems to be greater than the one of annotation graphs. Generally this question is outside the scope of the current paper, but the claim is based on the observation by Cotton and Bird: “...

An example of this kind of corpus is the HPSG Treebank for Polish (Marciniak et al., 2000). Representing such treebank using annotation graphs would require a more expressive model of arc labels than is currently permitted (namely attribute-value matrices)." Thus our feature graphs can be regarded as a variant of annotation graphs based on rigorous formal basis.

Another related work is: Redwood HPSG treebank of English (see (Oepen et. al., 2002)). The creation of this treebank uses decision trees approach to support the annotators in selection of the right HPSG ananlyses for the sentences. This approach is very close to our classification based on feature graphs. Another important characteristics of their treebank is its dynamic nature. This generally is concerned with changes and new developments in the underling linguistic theory. The problem is the following: when the theory changes the treebank becomes out of date. In order to support easily the updates of the already represented information one needs a mechanism for reuse of old analyses with small amount of work. The people working on Redwood treebank achieve this again by the means of decision trees. In our case we can use the reclassification for the same purpose.

The reclassification is also related to the approach described recently in (Kinyon and Rambow, 2003) which is used for transformation of treebanks from one linguistic theory to another (see the citation there for previous works on the problem). Again the difference with our approach is that we offer these operation to be done on the basis of logical formalism and the consistency of the result is guaranteed if the original information is consistent.

8 Conclusion

In this paper we presented a formal framework for development of a corpus based on HPSG linguistic theory. There are several advantages of such a formal framework:

- Uniformity of the annotation with respect to an HPSG grammar;
- Classification algorithm for facilitating the annotation process.
- Potential for reclassification which can be helpful during the development of the corpus and during its exploitation.

One very interesting side of such usage is for parser evaluation. First, HPSG as theory offers simultaneously representation of the constituent structure and the dependency relations. Second, the reclassification of the corpus can be context sensitive and this allows for different kinds of evaluation for different constructions. At the moment, we are developing a corpus based on the above formalism (see (Simov, Popova and Osenova, 2002; Simov et al., 2002; Simov and Osenova, 2003)). We have annotated about 5000 sentences. It is too early for real evaluation of the speed of annotation but the results are promising.

References

Anne Abejllé. (editor) 2003. *Treebanks. Building and Using Parsed Corpora.* Kluwer Academic Publishers.

J. Carroll, G. Minnen, T. Briscoe. 2003. *Parser Evaluation Using a Grammatical Relation Annotation Scheme.* In A. Abeillé (ed.), *Treebanks. Building and Using Parsed Corpora.*. Dordrecht: Kluwer

Scott Cotton and Steven Bird. 2002. *An Integrated Framework for Treebanks and Multilayer Annotations.* In: *Proceedings from the LREC conference,* Canary Islands, Spain. pp 1670-1677.

Stefanie Dipper. 2000. *Grammar-based Corpus Annotation.* In *Proceedings of the Workshop on Linguistically Interpreted Corpora.* Luxembourg.

P. Harrison, St. Abney, E. Black, D. Flickinger, C. Gdaniec, R. Grishman, D. Hindle, B. Ingria, M. Marcus, B. Santorini, and T. Stralkowski. 1991. *Evaluating syntax performance of parser/grammars of English.* In *Proc. of the Workshop on Evaluating Natual Language Processing Systems*. pages 71-77. Berkeley, Ca, USA.

Paul J. King. 1989. *A Logical Formalism for Head-Driven Phrase Structure Grammar.* Doctoral thesis, Manchester University. Manchester, England.

Paul J. King and Kiril Simov. 1998. *The automatic deduction of classificatory systems from linguistic theories.* In *Grammars*, volume 1, number 2, pages 103-153. Kluwer Academic Publishers, The Netherlands.

Alexandra Kinyon, Owen Rambow. 2003. *The MetaGrammar: a cross-framework and cross-language test-suite generation tool.* In: Proc. of The 4th International Workshop on Linguistically Interpreted Corpora Budapest, Hungary.

S. Kübler and E. Hinrichs. 2001. *From Chunks to Function-Argument Structure: A Similarity-Based Approach.* In *Proceedings of ACL-EACL 2001.* Toulouse, France.

Dekang Lin. 2003. *Dependency-based Evaluation of Minipar.* In *Proc. of the ESSLLI Workshop on Machine Learning Approaches in Computational Linguistics.* In A. Abeillé (ed.), *Treebanks. Building and Using Parsed Corpora.*. Dordrecht: Kluwer

Stephan Oepen, Ezra Callahan, Dan Flickinger and Christopher D. Manning. 2002. *LinGO Redwoods. A Rich and Dynamic Treebank for HPSG,* In: *Proc. of The Workshop Beyond PARSEVAL. The Third LREC Conference.* Las Palmas, Spain.

Carl J. Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar.* University of Chicago Press, Chicago, Illinois, USA.

Kiril Simov. 2001. *Grammar Extraction from an HPSG Corpus.* In *Proc. of the RANLP 2001 Conference.* Tzigov Chark, Bulgaria. pages 285–287.

Kiril Simov. 2002. *Grammar Extraction and Refinement from an HPSG Corpus.* In *Proc. of the ESSLLI Workshop on Machine Learning Approaches in Computational Linguistics.* Trento, Italy. pages 38–55.

Kiril Simov, Gergana Popova and Petya Osenova. 2002. *HPSG-based syntactic treebank of Bulgarian (BulTreeBank).* In: *"A Rainbow of Corpora: Corpus Linguistics and the Languages of the World",* edited by Andrew Wilson, Paul Rayson, and Tony McEnery; Lincom-Europa, Munich, pp. 135-142.

Kiril Simov, Petya Osenova, Milena Slavcheva, Sia Kolkovska, Elisaveta Balabanova, Dimitar Doikoff, Krasimira Ivanova, Alexander Simov, Milen Kouylekov. 2002. *Building a Linguistically Interpreted Corpus of Bulgarian: the BulTreeBank.* In: *Proceedings from the LREC conference,* Canary Islands, Spain.

Kiril Simov, Milen Kouylekov, Alexander Simov. 2002. *Incremental Specialization of an HPSG-Based Annotation Scheme.* In: *Proceedings of the Workshop on "Linguistic Knowledge Acquisition and Representation: Bootstrapping Annotated Language Data",* the LREC conference, Canary Islands, Spain.

Kiril Simov and Petya Osenova. 2003. *Practical Annotation Scheme for an HPSG Treebank of Bulgarian.* In: *Proc. of the 4th Workshop on Linguistically Interpreted Corpora.* Budapest, Hungary.