Semantic Network Manual Annotation and its Evaluation

Václav Novák

Institute of Formal and Applied Linguistics

Doctoral Thesis

Supervisor	Prof. RNDr. Jan Hajič, Dr. Institute of Formal and Applied Linguistics Malostranské náměstí 25 118 00 Praha 1 Czech Republic
Opponents	Prof. RNDr. Jaroslav Peregrin, CSc. Faculty of Philosophy and Arts, Charles University Prague, Czech Republic Mgr. Jan Štěpánek, Ph.D. Institute of Formal and Applied Linguistics, Charles University Prague, Czech Republic

Prague, May 23, 2008 Václav Novák

Abstract

The Prague Dependency Treebank (PDT) is a valuable resource of linguistic information annotated on several layers. These layers range from shallow to deep and they should contain all the linguistic information about the text. The natural extension is to add a semantic layer suitable as a knowledge base for tasks like question answering, information extraction etc. In this thesis I set up criteria for this representation, explore the possible formalisms for this task and discuss their properties. One of them, Multilayered Extended Semantic Networks (MultiNet), is chosen for further investigation. Its properties are described and an annotation process set up. I discuss some practical modifications of MultiNet for the purpose of manual annotation. MultiNet elements are compared to the elements of the deep linguistic layer of PDT. The tools and problems of the annotation process are presented and initial annotation data evaluated.

Acknowledgments

The work presented in this thesis was greatly influenced and supported by:

- Above all my supervisor Jan Hajič
- Institute of Formal and Applied Linguistics: Zdeněk Žabokrtský, Petr Němec, Eva Hajičová, and Petr Sgall
- Czech Academy of Science (grants 1ET201120505 and 1ET101120503)
- Czech Ministry of Education, Youth and Sports (projects LC536 and MSM0021620838)
- The most sturdy annotators: Chris Warkentin, Christi Brooks, and Simon Morton
- Intelligente Informations- und Kommunikationssysteme Department of Fernuniversität in Hagen: Sven Hartrumpf, Ingo Glöckner, Hermann Helbig, and Rainer Oswald
- Center for Language and Speech Processing of Johns Hopkins University: Keith Hall, Frederick Jelinek, and Robert Frank

Contents

Preface

> We want to say that there can't be any vagueness in logic. The idea now absorbs us, that the ideal '*must*' be found in reality. Meanwhile we do not as yet see *how* it occurs there, nor do we understand the nature of this "must". We think it must be in reality; for we think we already see it there.
>
> Wittgenstein (1953, I/§101)

At the very beginning, I would like to introduce the terminology I am using throughout this thesis by the preface in the form of a little dictionary of basic concepts. The best entry to start with is the **Goal**.

4$^{\text{th}}$ layer is the next natural layer of →**annotation**. The first layer is the morphology, the second is surface syntax, the third is the deep syntax a.k.a. linguistic meaning a.k.a. →**TR**. The fourth layer should have a form of a →**knowledge base**. The first three layers are described in →**FGD** and there is a large-scale annotation available in →**PDT**.

Annotation is a process in which humans produce a formalized representation (in our case they draw a →**semantic network**) of the newspaper texts. See some example annotations in Appendix A.

Argument of a →**function** can be any other →**concept** in the →**semantic network**.

Attributes Every →**concept** has at least one attribute: the sort of the concept. According to the sort, some other attributes may be relevant.

Concept is the basic unit of a →**semantic network**. Regular concepts (as opposed to relations) are visualized as →**node**s. The concepts should correspond to the mental images expressed by natural language expressions (de Saussure, 1916). An example of a concept is "my dog Fluffy", which is a specific →**subconcept** of a generic concept "dog".

Edge is an arrow →**represent**ing a →**relation**. The arrow goes from the →**source** to the →**target**. An alternative notation is (Source REL Target), where REL is the type of relation.

FGD Functional Generative Description is a linguistic theory describing the language system. This theory is not intended to offer a →**knowledge** representation format.

Function is a special kind of →**concept**. Apart from →**attributes** inherited from →**concept**, it also has the →**arguments** and the function type (e.g., ITMS for creating collection out of individuals, FLP for creating locations out of objects and local prepositions).

Goal of the →**annotation** project described in this thesis is to set up tools, instructions and evaluation measures to annotate the →**meaning** of newspaper sentences to produce the →**4th layer** of annotation.

Knowledge base is a collection of →**concepts** which represent the available →**knowledge**. Although the newspaper text as a string of words can be considered a knowledge base, it lacks some important properties required for a knowledge base to be practical. Therefore we →**annotate** the texts to the form of →**semantic network**.

Knowledge is the collection of facts we have. It consists of the background knowledge, and the knowledge coming from the text. During →**annotation**, we want to encode the knowledge coming from the text in the most useful format possible.

Meaning of a natural language expression is the →**knowledge** we gain by interpreting the expression. Interpretation means a translation into a chosen structure. Since I will not speculate about the translation to the brain structure, →**meaning** will in practice stand for the corresponding changes in →**semantic network** caused by the text.

MultiNet is one type of →**semantic network** formalism. The differences among various semantic network formalisms lie mainly in the repertoire of →**relations**. MultiNet has a relatively small set of relations and a relatively complicated system of →**attributes** of concepts. An important feature of MultiNet is that it is not domain-specific.

Node is a →**concept** which is not a →**relation**. A node is also the visualization of a concept in the →**annotation** tool.

PDT The Prague Dependency Treebank (PDT) is a collection of annotations according to the →**FGD** theory. In this thesis I investigate the possibility of adding the →**4**$^{\text{th}}$ **layer** of annotation.

Relation is a special kind of →**concept**. Apart from →**attributes** inherited from →**concept**, it has also the →**source**, the →**target** and the relation type (e.g., AGT for agent, ANTE for temporal precedence).

Represent is used in two different meanings. A MultiNet →**concept** *represents* a sequence in text, when the text sequence corresponds to the same mental concept as the concept in the →**semantic network**, e.g., "Concept 123 represents *my dog*". A →**node** *represents* a MultiNet →**concept** in the →**annotation** tool iff they have the same ID (see also →**edge**).

Semantic network is a directed multigraph where the graph nodes →**represent** →**concepts** and →**edge**s represent →**relations** between concepts.

Source of a →**relation** is the →**concept** at which the →**edge** starts.

Subconcept is a more specific →**concept**, e.g., "dog" is the subconcept of "mammal", which itself is a subconcept of "animal".

Target of a →**relation** is the →**concept** at which the →**edge** points.

TR Tectogrammatical Representation (TR) is the deep syntactic representation. The main difference between TR and the →**4**$^{\text{th}}$ **layer** is that TR units are words of the natural language, while →**semantic network** units are the →**concepts**.

We are the annotators and I.

This thesis starts with Chapter 1, where I introduce the PDT and discuss the possible ways to extend this approach with a layer representing the meaning. Chapter 2 introduces the selected formalism: MultiNet. I give the overview of its features and existing tools, and present the modifications I have made to the formalism to make it easier to annotate and to fit to the PDT framework. Chapter 3 gives a survey of existing annotation projects and presents the annotation tool *cedit*, which I created to allow the manual annotation, and the problems we encountered during the annotation effort I led. Chapter 4 discusses the methodology of annotation evaluation and presents the inter-annotator agreement I measured. Chapter 5 gives a conclusion.

Chapter 1

Extending PDT with Semantic Annotation

> But how many kinds of sentence are there? Say assertion, question, and command?—There are *countless* kinds: countless different kinds of use of what we call "symbols", "words", "sentences". And this multiplicity is not something fixed, given once for all; but new types of language, new language-games, as we may say, come into existence and others become obsolete and get forgotten.
>
> Wittgenstein (1953, I/§23)

1.1 Introduction to PDT

The Prague Dependency Treebank 2.0 (PDT 2.0) described in Sgall *et al.* (2004) contains a large number of Czech texts with complex and interlinked morphological (2 million words), syntactic (1.5M words), and complex semantic (tectogrammatical) annotation (0.8M words); in addition, certain properties of sentence information structure and coreference relations are annotated at the semantic level.

The theoretical basis of the treebank lies in the Functional Generative Description (FGD) of the language system by Sgall *et al.* (1986).

PDT 2.0 is based on the long-standing Praguian linguistic tradition, adapted for the current computational-linguistics research needs. The corpus itself is embedded into the latest annotation technology. Software tools for corpus search, annotation, and language analysis are included. Extensive documentation (in English) is provided as well.

An example of a tectogrammatical tree from PDT 2.0 is given in figure 1.1. Function words are removed, their function is preserved in node attributes (*grammatemes* and *subfunctor*), information structure is annotated in terms of topic-focus articulation, and every node receives detailed semantic label corresponding to its function in the utterance (e.g., *addressee*, *from_where*, *how_often*, . . .). The square node indicates an obligatory but missing subject. The tree represents the following sentence:

Letos se snaží o návrat do politiky. (1.1)

This year he tries to return to politics.

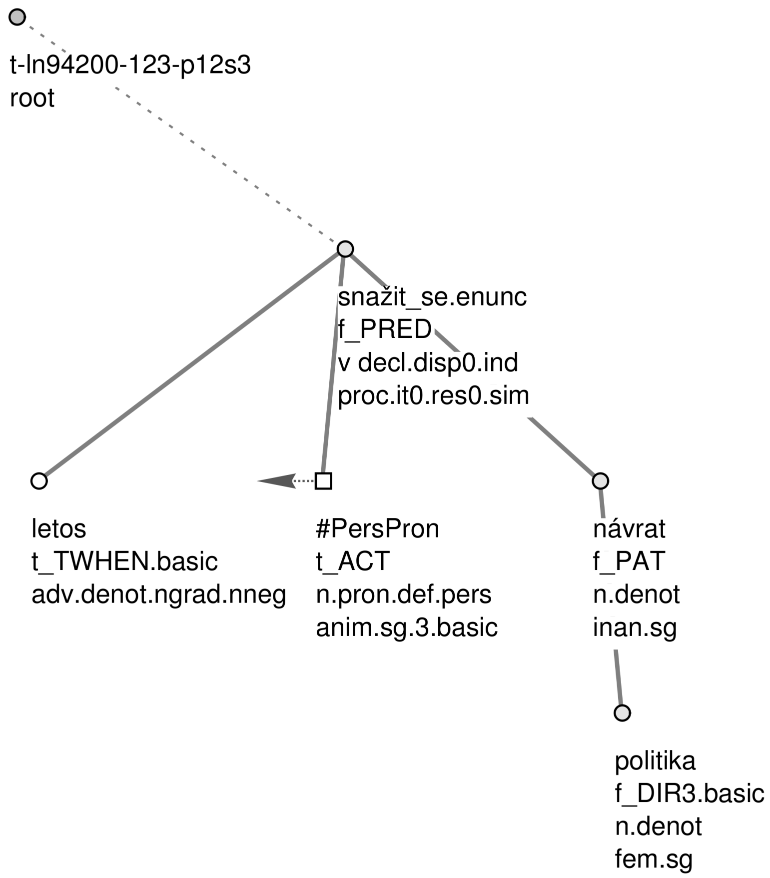


Figure 1.1: Tectogrammatical tree of sentence (1.1)

1.1.1 PDT layers

PDT 2.0 contains three layers of information about the text (as described in Hajič (1998)):

Morphosyntactic Tagging. This layer represents the text in the original linear word order with a tag assigned unambiguously to each word form occurrence, much like the Brown corpus does.

Syntactic Dependency Annotation. It contains the (unambiguous) dependency representation of every sentence, with features describing the morphosyntactic properties, the syntactic function, and the lexical unit itself. All words from the sentence appear in its representation.

Tectogrammatical Representation (TR). At this level of description, we annotate every (autosemantic non-auxiliary) lexical unit with its tectogrammatical function, position in the scale of the communicative dynamism and its grammatemes (similar to the morphosyntactic tag, but only for categories which cannot be derived from the word's function, like number for nouns, but not its case).

1.2 Motivation

The longterm goal of the research in the field of Artificial Intelligence has been to create a machine which would understand natural language input and be able to perform the reasoning necessary to perform the desired actions. It is obvious that such a machine must be capable of storing the acquired information in its memory in a form suitable for the necessary reasoning. We will call this form the *knowledge representation*. This chapter will discuss the criteria which should be imposed upon the form of the information representation, and the existing systems for knowledge representation and their properties with respect to the given criteria.

There are several reasons why TR may not be sufficient in a question answering system or machine translation:

1. There is no information about sorts of concepts represented by TR nodes. Sorts (the upper conceptual ontology) are an important source of constraints for semantic relations. Every relation has its signature which in turn reduces ambiguity in the process of text analysis and inferencing.

2. The syntactic functors Actor and Patient disallow creating inference rules for cognitive roles like *Affected object* or *State carrier*. For example, the axiom stating that an affected object is changed by the event ($(v$ AFF $o) \rightarrow (v$ SUBS `change.2.1`$)$) can not be used in the TR framework. However, if needed, this information can be stored in the lexicon for individual verb frames.

3. Lexemes of TR have no hierarchy which limits especially the search for an answer in a question answering system. In TR there is no counterpart of SUB, SUBR, and SUBS MultiNet relations, which connect subordinate concepts to superordinate ones and individual object representatives to corresponding generic concepts.

4. In TR, each sentence is isolated from the rest of the text, except for coreference arrows connected to preceding sentences. This, in effect, complicates inferences combining knowledge from multiple sentences in one inference rule.

5. Nodes in TR always correspond to a word or a group of words in the surface form of a sentence or to a structure which is deleted on the surface (e.g., obligatory verb argument, coordination member). There are no means for representing knowledge generated during the inference process, if the knowledge does not have a form of TR. For example, consider the axiom of temporal precedence transitivity (1.2):

$$(a \text{ ANTE } b) \wedge (b \text{ ANTE } c) \rightarrow (a \text{ ANTE } c) \tag{1.2}$$

   In TR, we can not add an edge denoting (a ANTE c). We would have to include a proposition like "a precedes c" as a whole new clause.

For all these reasons we need to extend our text annotation to a form suitable to more advanced tasks. It is shown in Helbig (2006) that MultiNet (as described in Chapter 2) is capable of solving all the above mentioned issues.

In the process of deeper understanding, it seems obvious that we need another layer in order to include the whole discourse in a structure allowing inferences provided by comprehensible axioms that may be both handmade and automatically acquired. It is shown in Lin and Pantel (2001) that even the layer of shallow syntactic dependency annotation can already successfully allow useful inferences by induction to be statistically derived. It is our hope that the structure provided by the MultiNet formalism can rival the simple syntactic structure as the input for the inference inductor.

With a sufficient initial size of the knowledge base and the dictionary we can then expand the knowledge automatically by means of bootstrapping (Eisner and Karakos, 2005) or other automatic knowledge mining methods like GUHA (Hájek and Havránek, 1978). These and other methods can operate independently and concurrently on the knowledge base.

The MultiNet format is much more suitable for the above mentioned algorithms than surface syntax or even plain text. Even if we used it directly at the level of tectogrammatical representation, we could not insert the results of the modules back to the representation and we would not be able to use a module on data already enriched by another one. The resulting knowledge base is an invaluable resource in its own right, creating opportunities for further research.

Furthermore, the axiomatic system connected to MultiNet can not be applied to other layers of annotation, because they lack the necessary regularity with respect to inference rules (e.g., the syntactic roles can not give the kind of information that the AFF relation in MultiNet gives us – the affected object is changed by the event).

1.3 Criteria

In order to efficiently retrieve and process the knowledge acquired in the form of natural language input, these criteria should be fulfilled by the internal knowledge representation format:[1]

I. **Associativity**: The knowledge concerning a concept should be available without the necessity to iterate over the whole knowledge base. A representation lacking this property would not be scalable to real problems.

II. **Local interpretability**: The knowledge necessary for interpretation of an object should be limited to an easily identifiable local neighborhood of the concept (the knowledge may include a contextual embedding which is crucial for the concept interpretation).

III. **Inference friendliness**: The knowledge data format should allow for further inclusion of new facts, acquired both by new texts and by automatic inferencing. It is important to be robust with respect to contradictions.

[1] Criteria II., A., B. and C. are modifications of some of the criteria imposed by Helbig (2006). I formulated criteria I. and III.

Apart from the overall necessary requirements, there are also further criteria necessary for a representation if it is to be annotated manually:

A. **Consistency**: Analogous facts should be treated analogously.

B. **Cognitive Adequacy**: The representation must be understandable to the annotators and easy to visualize and review.

C. **Communicability**: The instructions should contain applicable operational criteria (Hajičová and Sgall, 1980), definitions, and standards.

The next requirement for the representational formalism is to integrate smoothly into the layered nature of the PDT (Karcevskij, 1929; Callmeier *et al.*, 2004).

Why are these requirements crucial?

Without associativity (I.), the query for information would always require a search through the whole knowledge base. Furthermore, for queries which can not be answered using only one sentence, one would have to create a kind of associative structure on the fly to make use of disambiguation, coreferences etc.

Local interpretability (II.) is needed for concepts embedded in a way that changes their mode of existence. Consider the clause "*If I were you*". We do not want to extract the information that *I* refers to the same person as *you*. However, this is what we would infer if we ignored the contextual embedding associated with the word *if*. Therefore the knowledge representation must ensure this information is readily available for every piece of information without the need to iterate through the whole knowledge base.

Inference friendliness (III.) allows us to enrich the acquired knowledge by applying inference rules. If we know that "*Mrs. Hill is the current vice president finance*", we can infer for instance that "*The current vice president finance is Mrs. Hill*". An inference friendly representation will allow a compact representation of such an inference. Without this compactness (e.g., in the case where the inference must be included as a whole new sentence) the scale of practical inferences would be very limited.

Without consistency (A.) the annotation process is unimaginable, because annotators are able to use only a limited set of instructions and they always treat the new sentences by analogy. If this were not the correct way to annotate, they could not produce meaningful results.

Cognitive adequacy (B.) is practical when the annotators must deal with complicated sentences. There are few people who understand modal operators and first order logic axioms, but there are many people who understand the sentences in *The Wall Street Journal*. Ideally, the complexity of annotating a sentence should be 100% correlated with the complexity of understanding its meaning. Without cognitive adequacy of the representation, the annotation can not leave the realm of toy sentences.

Communicability (C.) is another key to the success of annotation. A mere learning by example can prove to be useful, but it fails in the case of border cases. Unfortunately, however contradictory this may sound, border cases make up a significant percentage of decisions and can be found in every Wall Street sentence.

1.4 Existing Meaning Representations

In this section we will discuss various formalisms of knowledge representation and their conformance to the criteria presented in Section 1.3. Some related annotation projects which are not considered meaning representations are presented in Section 3.1 on page 62.

1.4.1 Representations Based on First Order Logic

The first attempts to formalize natural language were made using the predicate calculus (Frege, 1892). Since then various approaches have been trying to fix the problems of using first order logic purely extensional interpretation of the meaning. First, intensional semantics was developed (Montague, 1972) to introduce the notion of conceivable worlds. This theory was further developed in several directions:

- TIL: Transparent Intensional Logic (Tichý, 1988) aimed at further elaboration of the semantics of conceivable worlds

- Description Logic (Donini *et al.*, 1996) focused on the computational aspects of meaning representation.

- DRT: Discourse Representation Theory (Eijk and Kamp, 1996) focused on the treatment of coreferences, quantifiers, and their interplay.

- Hybrid Modal Logic (Areces and Blackburn, 2001; Areces *et al.*, 2004; Blackburn, 2000, 2001) applied the framework of modal logic to natural language semantics.

All these formalisms have been used to represent real-life sentences. There has been a successful attempt to automatically create DRT structures proposed in Bos (2005). Hybrid Modal Logic has been investigated from the linguistic viewpoint in Kruijff (2001); Novák (2004); Novák and Hajič (2006). The TIL has been subject to automatic transduction (Horák, 2001), but not to manual annotation.

How do these systems fit into our criteria? They are very strong in associativity (I.): every concept is represented by one or more variables and these variables can be looked up easily. Inference friendliness (III.) is guaranteed as to the ease of addition of new knowledge: it can be added by simply adding predicates. On the other hand the robustness with respect to contradictions is addressed only in some of these systems and in general requires non-monotonicity of the reasoning.

Local interpretability (II.) is addressed only in DRT, where the relevant contextual embedding should be present only in the current box. Cognitive adequacy (B.) is the most difficult obstacle which prevents these systems from being manually annotated. The model-theoretic way of thinking and use of quantifiers are largely unintuitive. This is not apparent for sentences which are usually addressed in this literature (e.g., "*Every farmer owns a donkey*"). Nevertheless, it emerges when we try to come up with a predicate calculus representation of an ordinary sentence like "*The U.S. trade representative, Carla Hills, announced . . .* " it seems unintuitive to think about *trade* as a function from possible worlds to a set of objects, which is the typical treatment for nouns.

1.4.2 Representations Based on Linguistic Structures

The meaning representations based on linguistic structures emerged as an extension of dependency syntax (Tesnière, 1934). There are various formalisms, which all share some common features: they start with the text or speech and transform it into formalized layers of representation, where the last layer should be the most suitable for the knowledge representation tasks. They are:

- Functional Generative Description (Sgall *et al.*, 1986), where the highest layer of description is the Tectogrammatical Representation (Hajičová *et al.*, 2000a)

- Robust Minimal Recursion Semantics (Copestake *et al.*, 2005) as a pluggable layer of the framework of Callmeier *et al.* (2004)

- Meaning-Text Theory (Mel'čuk, 1988; Bolshakov and Gelbukh, 2000), which is in many respects similar to the FGD framework (Žabokrtský, 2005).

These approaches have difficulties with respect to the inference friendliness (III.): to include a piece of inferred knowledge, we often have to add a whole new sentence which describes the fact. For example if we are to apply a rule stating a symmetry of a predicate in a logic-based system, we simply add one predicative statement for every instance. In a linguistics-based system, we have to copy the whole statement and transform it into the inverse form.

The next obstacle concerns the cognitive adequacy: the tree constraints force the annotators to choose only one connection where more of them could be applied: in "*They met during the concert on Tuesday.*" the above mentioned systems require the annotator to decide whether *on Tuesday* is connected to *met* or *concert*, although from the knowledge base viewpoint it would be ideal if both *met* and *concert* were connected with the temporal specification under consideration.

1.4.3 Semantic Networks

Semantic networks, as different from the logic-based systems as they may seem, have much in common with them. The semantic network, being a directed graph, can usually be turned into a set of formulae of predicate calculus. The main difference lies in the fact that the relationship between the predicates and the knowledge is not direct: the predicates encode information about the network. The elements of the network then carry their own meaning.

The main advantage of semantic networks is their concept-centeredness. As noted in Helbig (2006, p. 4), the difference is similar to the difference between a logical programming language (e.g., Prolog) and an object oriented programming language (e.g., Java). Every concept should correspond to a cognitive concept and it is assumed that two distinct concepts do not represent the same object, unless there is a piece of information indicating the opposite. On the other hand, in a model-theoretic framework, the model builders tend to create a model as small as possible, therefore collapsing the referents of all variables where possible. This, in effect, often leads to a wrong conclusion.

Individual semantic network formalisms differ in their repertoire of formal means. In practice, two systems have been used for purposes of natural language processing:

- KL-ONE: knowledge representation system (Brachman and Schmolze, 1985)
- MultiNet: Multilayered Extended Semantic Networks (Helbig, 2006)

They satisfy all the criteria presented in Section 1.3 and therefore they are discussed in the remaining chapters.

1.4.4 Semantic Web

A Semantic web is sometimes considered yet another semantic representation. However, it is more a framework allowing us to standardize the representations and exchange the data in a structured format. It is therefore not possible to simply create a semantic web corpus. The technologies being used are the Web Ontology Language (Horrocks and Patel-Schneider, 2004), which allows for standardization and exchange of ontologies, and Resource Description Framework (RDF Core Working Group, 2007), which is an XML-based data format for exchanging predicate-like structures.

Chapter 2

MultiNet: Properties and Modifications

> Consider this example. If one says "Moses did not exist", this may mean various things. It may mean: the Israelites did not have a *single* leader when they withdrew from Egypt or: their leader was not called Moses or, there cannot have been anyone who accomplished all that the Bible relates of Moses—or: etc. etc.— We may say, following Russell: the name "Moses" can be defined by means of various descriptions. For example, as "the man who led the Israelites through the wilderness", "the man who lived at that time and place and was then called 'Moses' ", "the man who as a child was taken out of the Nile by Pharaoh's daughter" and so on. And according as we assume one definition or another the proposition "Moses did not exist" acquires a different sense, and so does every other proposition about Moses.—And if we are told "N did not exist", we do ask: "What do you mean? Do you want to say ...... or ...... etc.?"
>
> Wittgenstein (1953, I/§79)

The representational means of Multilayered Extended Semantic Networks (MultiNet), which are described in Helbig (2006), provide a universally applicable formalism for treatment of semantic phenomena of natural language. To this end, they offer distinct advantages over the use of the classical predicate calculus and its derivatives. The knowledge representation paradigm and semantic formalism, MultiNet, is used as a common backbone for all aspects of natural language processing (be they theoretical or practical ones). It is continually used for the development of intelligent

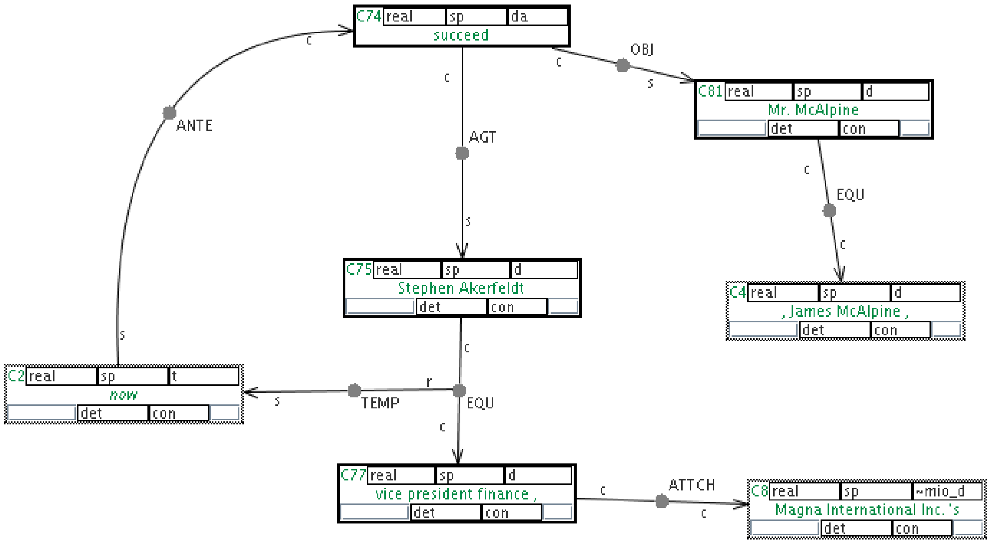


Figure 2.1: MultiNet annotation of sentence "*Stephen Akerfeldt, currently vice president finance, will succeed Mr. McAlpine.*" Nodes C4 and C8 are re-used from preceding sentences.

information and communication systems and for natural language interfaces to the Internet. Within this framework, it is subject to permanent practical evaluation and further development.

The semantic representation of natural language expressions by means of MultiNet is mainly independent of the considered language (Helbig, 2006). In contrast, the syntactic constructs used in different languages to describe the same content are obviously not identical. An example of a MultiNet structure is given in Figure 2.1.

MultiNet is not explicitly model-theoretical and the extensional level is created only in those situations where the natural language expressions require it. It can be seen that the overall structure of the representation differs from a Tectogrammatical representation (TR) in that it is not a tree.

Note that the MultiNet representation crosses the sentence boundaries. First, the structure representing a sentence is created and then this structure is assimilated into the existing representation.

In contrast to CLASSIC (Brachman *et al.*, 1991) and other KL-ONE networks, MultiNet contains a predefined final set of relation types, encapsulation of concepts, and attribute layers concerning the cardinality of objects mentioned in discourse.

2.1 Roots

The MultiNet formalisms are based on the ideas of semantic networks, which started with Associative Networks (Quillian, 1968). Since then, there have been many systems proposed addressing some issues of the original proposal. MultiNet builds upon the Extensional Semantic Networks of Janas and Schwind (1979) and combines their proposal with the Generalized quantifier theory of Barwise and Cooper (1988).

2.2 Overview of Representational Means

2.2.1 Attributes

Every MultiNet concept (i.e., also relations and functions), has several attributes. A more complete description can be found in Helbig (2006, pp. 409).

SORT One of the 51 sorts from the hierarchy. These 51 sorts also include 6 "meaning molecules", i.e. concepts which have more than one sort at the same time (e.g., "certificate" as both discrete object and an ideal object). The sorts are listed on the project wiki page[1] and also in Appendix B.

FACT (only relevant for concepts of SORT subsumed under *o*, *si*, *t*, or *l*) determines the facticity of the entity. The entity can be either real, hypothetical, or non-existent. This attribute is of great importance to the requirement of local interpretability (II on page 21).

GENER (relevant for the same sorts as FACT) distinguishes generic concepts from specific concepts. Consider the following sentence:

A house needs to be maintained regularly. (2.1)

Sentence 2.1 has two readings. The preferred one in this case is with "house" being generic, the less probable reading with a specific house (meaning "there exists a house that needs . . . ").

CARD (only relevant for concepts of SORT subsumed under *o*, *t*, or *l*) determines the cardinality of collections of objects. If CARD is filled in, ETYPE must be at least one.

[1] https://wiki.ufal.ms.mff.cuni.cz/projects:content-annotation:layer-attributes:sort:hierarchy-of-sorts

REFER (relevant for the same sorts as CARD) distinguishes determinate concepts from indeterminate concepts.

VARIA (relevant for the same sorts as CARD) distinguishes fixed concepts from variable ones. Most concepts are constant. Example of a variable concept is "man" in "*Every man loves a woman.*" The "woman" in the sentence is ambiguous with respect to the VARIA attribute. In the preferred reading, "woman" is also variable.

ETYPE (relevant for the same sorts as CARD) distinguishes collections from individuals. Individuals have ETYPE zero, collections have ETYPE one, collections of collections have ETYPE two, collections of collections of collections have ETYPE three, etc. See Section 2.3.4 for further discussion.

2.3 Modifications and Adjustments

In order to efficiently annotate the MultiNet semantic networks, I made several minor changes to the MultiNet specification. The changes are rather technical and they are presented here so that a reader familiar with MultiNet would avoid any confusion that could arise from these modifications.

2.3.1 SUB Relation

In MultiNet, the generic concepts, which belong to the ontology, are clearly distinguished from the specific concepts representing specific individuals and collections. If we talk about "my dog", the concept of my dog is C1 in the Figure 2.2. We know that the dog is one particular instance of the generic concept "dog" (C2).

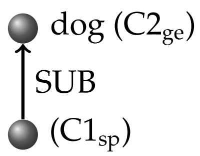


Figure 2.2: The MultiNet subnetwork corresponding to a particular dog C1.

In this way we can map concepts of the discourse to the corresponding concepts of the ontology. However, almost all specific nodes have SUB relation to their generic superconcept, and furthermore they almost always have only one such direct superconcept. Therefore we can simplify the annotation by collapsing the two concepts in Figure 2.2 to only one concept, which is labeled with the appropriate superconcept. In this way we can reduce the number of nodes in the network by almost half, which is certainly a great advantage to manual annotation. This idea is also used in the HaGenLex interface (see Figure 2.7 on page 58), where superconcepts are not explicitly shown as concepts on their own.

This modification does not change the expressivity of the formalism, because if needed, the SUB relation can still be used. The modification can be considered as a shorthand notation collapsing two concepts into one.

2.3.2 Meta-edges

Every relation in MultiNet is a MultiNet concept in itself and it can become a source or target of another relation (Helbig, 2006, p. 452). Formally, for every edge we have a metaconcept representing the relation. There are edges ARG1 and ARG2 from the metaconcept to the source and target nodes of the relation, respectively. The metaconcept is created as needed. A simpler notation used in the book as well as in our annotation is to draw the edges coming directly from or to the other edges like the TEMP edge in Figure 2.1 on page 28.

The difference lies in the fact that while in the original system the simplified notation is only a shorthand for the metaconcept, we do not create any metaconcepts, but simply allow relations to have relations as their sources and targets. This not only eliminates the need for ARG1 and ARG2 relations, but also reduces the cognitive load of annotation by omitting the metaconcepts. This solution also prevents us from creating metaconcepts which might be never used. Examples of meta-edges can be found in the example sentence 3 in Figure A.3 on page 104 (the TEMP and CONF relations).

2.3.3 Meaning Molecules

Meaning molecules in MultiNet terminology are concepts which have more than one sort at the same time. As an example, "parliament" is a collection of individuals, but at the same time it is an institution and it may be a building. These three readings of parliament are called *facets* and they form the meaning molecule. In the original MWR tool (see Section 2.6.1), the user can assign any combination of sorts to any concept. In our annotation, however, we limited ourselves to combinations from a fixed list of plausible combinations[2] listed in Table 2.1. The possibility of choice from six options removes the burden associated with annotating meaning molecules considerably.

Sorts	Example
ideal object, discrete object	certificate, orchestra, school, province, cube
dynamic abstraction, ideal object	globalization, speech
ideal object, discreet object, dynamic abstraction	announcement
ideal object, operational attribute	temperature
static abstraction, ideal object	anarchy
static abstraction, dynamic abstraction	pregnancy

Table 2.1: Admissible meaning molecule types and their examples

2.3.4 ETYPE, QUANT and CARD attributes

Representation of quantities is quite complex in MultiNet. It is achieved using three attributes: ETYPE, QUANT and CARD. ETYPE is a numerical attribute distinguishing individuals (ETYPE=0) from collections (ETYPE=1), collections of collections (ETYPE=2), collections of collections of collections (ETYPE=3), etc. In practice, however, it is completely impossible to determine the appropriate ETYPE value other than 0 or 1. As examples in Helbig (2006, p. 430), organization has ETYPE=1 and umbrella organization has ETYPE=2. Suppose our text contains the expression "IBM[3]". What ETYPE should we assign? IBM has several branches, each of them has several departments, some departments may have multiple project teams, some of them not. I would assign ETYPE=4, but this number can change whenever any of these project teams begins to be divided into two subteams. For these reasons, we limit ourselves to ETYPE=0 for individuals and ETYPE=1 for collections, where we do not consider organizations collections in contexts where the collectiveness of the organization is not emphasized.

[2]The list has been compiled from a more extensive list of semantic categories provided by Sven Hartrumpf.

[3]IBM is a large company in the computer industry.

QUANT attribute is very special among the other attributes in the fact that it does not have a closed set of possible values. Indeed, quantity can be described by very complex phrases. It is implausible to fill whole phrases as an attribute of a concept, because it is against the requirement of the language neutrality of the representation and it requires special treatment of phrases that describe quantities for no good reason. In our annotation we replaced the QUANT attribute with a new QUANT relation, which connects a concept to the expression of quantity.

CARD attribute is used for concepts of ETYPE>0 and it expresses the preextensional cardinality of objects. In MultiNet this attribute is inferred from the value of QUANT attribute, therefore in practice we never fill in this attribute during the annotation.

2.3.5 Function Notation

In MultiNet, the function type is visualized as a property of edges coming from a concept. This in principle allows the use of more than one function within the same concept. In our approach, the function type is a property of the function concept. Arguments of the function are connected to the function node with dashed lines, while relations going from and to the function node are represented by solid lines. Figure 2.3 illustrates the function treatment in the original MultiNet, while our treatment of functions is displayed in Figure A.2 in Appendix A on page 103 (node F48 is a function node). This change makes the annotation more coherent, prevents us from using more than one function for a concept and it is also easier to incorporate the function annotation into the graphical user interface.

2.3.6 Indexed Functions

Apart from ordinary functions, MultiNet contains two families of functions: FLP, for creating locations out of objects, and OP, for arithmetic operations. These two sets of functions are indexed by local prepositions and arithmetic operators, respectively. To unify the treatment and annotation of all functions, we replaced the index of the function by a new argument of the function, i.e. we replace every function of the form $f_j(x, y)$ by function $f(x, y, j)$. From the annotation point of view this ensures uniformity in the treatment of all functions. From the semantic point of view, we extended the annotation possibilities. Originally, there was a fixed set of possible arithmetic operators, and to add a new one, one had to change the semantic network specification. In our new approach, a new operator

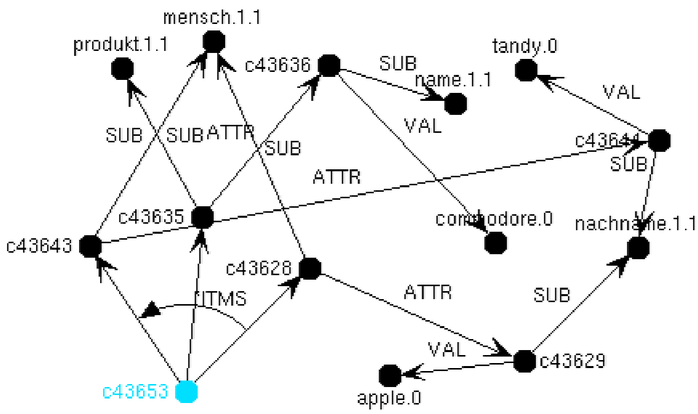


Figure 2.3: Example of a function node in the original MultiNet. The function arguments are the nodes c43628, c43625 and c43643 and the function value is the highlighted node c43653.

can be used the same way as the old ones are. Similarly, if during the annotation we come across a local preposition which is not in the original list of possible FLP indices, we do not have to introduce a new function.

2.3.7 Ternary relations

There are two ternary relations in Helbig (2006): ANLG and DISTG (for concepts analogous and distinguished with respect to a property). Both of them are in fact shorthands for the respective binary relations. By eliminating them we did not lose any expressional power and we greatly reduced the burden to annotators (and the complexity of the annotation tools) by considering only binary relations. This decision was an easy one also because of the fact that these two relations are hardly ever used for representation of real-world sentences.

2.4 Treatment of Selected Phenomena

In this section I discuss some phenomena interesting from the point of view of Functional Generative Description and their treatment in the MultiNet semantic networks.

2.4.1 The Treatment of Topic-Focus Articulation

In describing an information structure of natural language utterances (i.e., what is said about what), we can naturally use the notion of Topic-Focus Articulation (TFA) from Functional Generative Description (Sgall *et al.*, 1986). TFA has been shown to be of importance for the content of utterances, whether the content is encoded in a first order logic (Peregrin, 1995a), Discourse Representation Theory (Kruijff-Korbayová, 1998), Tripartite Structures (Hajičová *et al.*, 1998), or if it is encoded by means of Tectogrammatical Representation (TR) and used in a question answering system (Jirků and Hajič, 1982).

An example of a MultiNet structure from Helbig (2006) is given in Figure 2.4. The figure represents the following discourse:

Max gave his brother SEVERAL APPLES. (2.2)
This was a GENEROUS GIFT.
Four of them were ROTTEN.

Note that the MultiNet representation crosses the sentence boundaries. First, the structure representing a sentence is created and then this structure is inserted into and connected to existing representation.

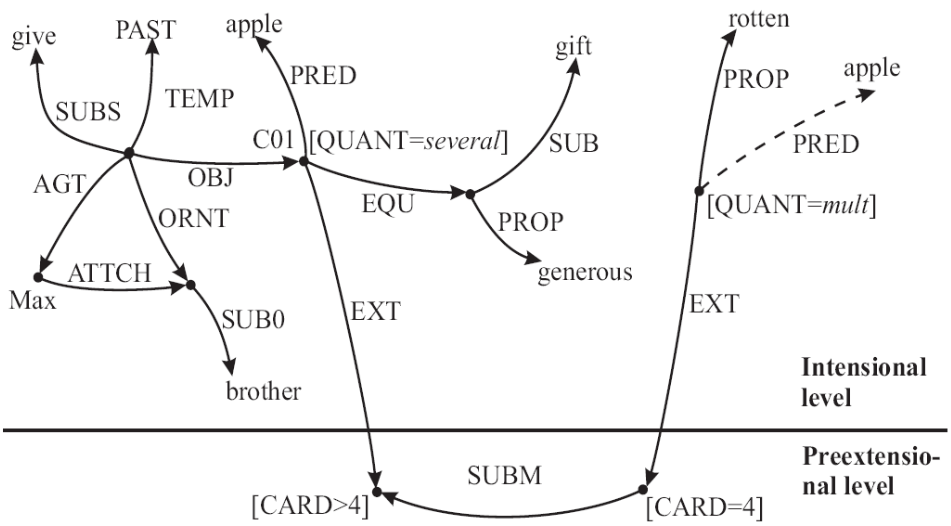


Figure 2.4: MultiNet representation of example discourse (2.2)

Contrastive Topic, Topic, and Focus

Topic-Focus Articulation is represented in TR by two means: first, marking individual nodes c, t, and f for being a part of contrastive topic, topic, and focus, respectively. Second, ordering of nodes by Communicative Dynamism, where the most focused nodes are to the right.

The **contrastive topic** contributes to the structuring of texts, it is one of the most important factors for determining inter-sentential links (Hajičová *et al.*, 2003). We mark the contrastive part of a sentence by c superscript as in example 2.3. The intonation center is capitalized.

The hosting^c team^c was successful at the BEGINNING. The visiting^c team^c succeeded only in the SECOND HALF. (2.3)

The **topic** can be paraphrased as 'what the sentence is about'. Topic is marked by t superscript as in example 2.4.

I^t saw^t him^t YESTERDAY. (2.4)

Finally, **focus** is understood as a part of the sentence that is signaled by the bearer of the intonation center and by its unmarked position at the end of the utterance, it is 'what is said about the topic'. The nodes in focus are marked by f superscript:

We^t need^t more^f WATER^f. (2.5)

TFA in MultiNet

There are several factors by which TFA contributes to the content of the utterance. We divided them into five groups, which will be discussed in the sequel:

1. Marking of definitional knowledge
2. Presuppositions and allegations
3. Scope of quantifiers
4. Degree of exhaustiveness
5. Contrasting

Definitional Knowledge

In semantic networks, the knowledge about a concept is often divided into two parts, the *definitional* part and the *assertional* part (Brachman and Schmolze, 1985). The definitional knowledge about X is the knowledge that should form the output of the question "What is X?" or "Tell me about X".

In MultiNet, this distinction is represented by K_TYPE attribute of the edge connecting X with other concepts.[4]

Many examples in the TFA literature (Sgall *et al.*, 1986; Hajičová *et al.*, 1998) can be at least partially explained by the difference between definitional and assertional knowledge. Typically, definitional knowledge is attributed to concepts representing nodes in topic, while concepts in focus are connected with assertional edges:

$$\text{ENGLISH}^f \text{ is}^t \text{ spoken}^t \text{ in the Shetlands}^t. \tag{2.6}$$

In this example, the speaker clearly conveys definitional knowledge about the Shetlands, not about English. Similarly in the following example:

$$\text{Karl}^t \text{ works}^t \text{ on his}^t \text{ homework}^t \text{ IN THE EVENINGS}^f. \tag{2.7}$$

This utterance should be used when answering general questions about Karl's homework, but by default not about Karl's evenings. In case of a question about Karl's evenings, the answer might be "I only know that Karl works on his homework." This strategy was already used in Jirků and Hajič (1982).

However, there are cases where this rule of thumb does not produce the right attributes:

$$\text{Bees}^t \text{ produce}^t \text{ HONEY}^f. \tag{2.8}$$

In this case it can be said that this is a part of definitional knowledge about bees, but at the same time a part of definitional knowledge about honey. On the other hand, it may be objected that the latter knowledge is not expressed in this sentence, i.e. it is only our background knowledge, that honey is produced by bees.

Consider the following example:

$$\text{On Friday}^t \text{ Geert}^t \text{ went}^f \text{ to the CINEMA}^f. \tag{2.9}$$

[4] Every edge has two K_TYPE attributes, one for each end, this is the attribute adjacent to X.

This time it is not obvious that there is some definitional knowledge encoded in the utterance. But if we ask a question answering system to describe this particular Geert's Friday, we would expect this piece of knowledge as a part of the answer, which suggests that it is definitional. These examples lead to the hypothesis that MultiNet edges going from nodes representing the topic to the nodes representing the focus should be considered definitional for practical purposes.

2.4.2 Presuppositions and Allegations

The influence of TFA on presuppositions has been described in Sgall *et al.* (1986); Peregrin (1995b,a, 1997). Allegation is described in Hajičová (1984). The most famous example of presupposition is probably this one:

The king^t of France^t is^f BALD^f. (2.10)

From a knowledge base point of view, there is not much difference in the treatment of the asserted fact and the presupposed fact. Both of them will become a part of our knowledge. The difference will appear at a point where we discover a contradiction. If the contradiction is caused by a presupposition of a new utterance, it is an indicator that our understanding of the discourse is wrong. However, in practice, there is no need to distinguish presupposed knowledge from asserted knowledge in a knowledge base except for the error detection.

In case of allegation, the situation is a little different. We can compare the typical example of an allegation using the same sentence but with a different TFA (a possible continuation is in parentheses):

(a_1) We^t didn't^t win^t because of ALICE^f. (we won because of Julie)
(a_2) We^t didn't^t win^t because of ALICE^f. (we lost although others played well)
(b) We^t didn't^f WIN^f because of Alice^t. (I told you we would lose)
(c) We^t DIDN'T^f win^t because of Alice^t. (we lost because of Patrick)
(2.11)

The case (a) is an example of allegation. It is ambiguous with respect to the scope of negation – either we won, but not because of Alice (a_1), or we did not win and it was because of Alice (a_2). Sentences (b) and (c) are not ambiguous with this respect. Since the scope of negation is disambiguated on the level of TR by the indices f and t, there is no need for ambiguity resolution in the process of MultiNet annotation if this information can be used. However, there are differences in the representation of negation

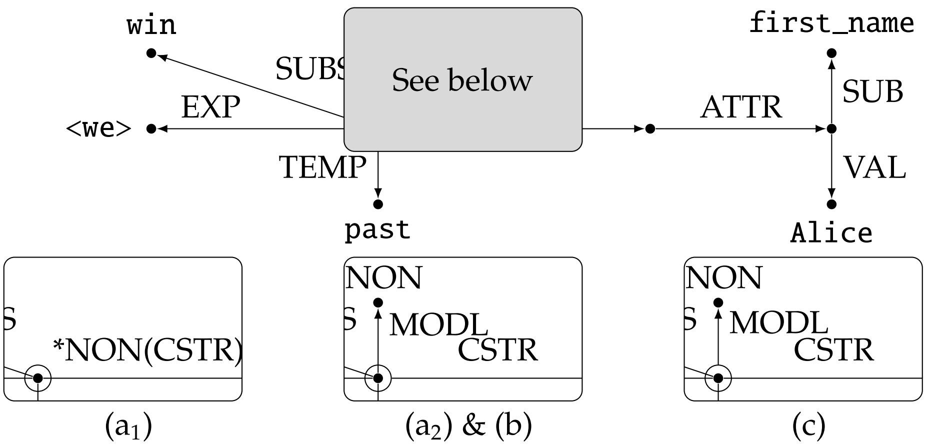


Figure 2.5: MultiNet representation of sentences (2.11) – the shaded area is to be replaced by the corresponding rectangle in the lower part of the figure.

between (a_1), (a_2), (b), and (c). The representations are given in Figure 2.5. The encapsulation of concepts is visualized by circles around vertices: edges going from inside belong to the capsule.

Note that the differences lie mainly in encapsulation of the event and in different positions of the negation. The representations of (a_2) and (b) are identical, although there is a difference in processing: (a_2) contains a presupposition that we did not win, while (b) contains a presupposition that Alice did not do well.

2.4.3 Scope of Quantifiers

The semantic impact of TFA on the scope of quantifiers is described in detail in Hajičová *et al.* (1998). The treatment of this phenomenon in MultiNet is explained in Helbig (2006, p. 206). Both these formalizations offer a formal treatment of this phenomena, therefore the automatic transduction should be straightforward. In the case of the manual MultiNet annotation the annotators do not use much information from the TR because understanding TR would require a significant amount of training.

2.4.4 Degree of Exhaustiveness

Degree of exhaustiveness is connected with the distinction between definitional and assertional knowledge. For example from the sentence (2.6) we can infer as default knowledge (subject to further refinement) that no other language is spoken in the Shetlands. It is, however, difficult to construct these inferences which require a certain amount of background knowledge (we know that a set is to be exhausted, but we do not know what set (e.g., all languages) it is). On the other hand the definitionality is indicated in the K_TYPE attribute, therefore we do not need another means of preserving this sort of TFA semantics.

2.4.5 Contrasting

The contrastive topic has been subdivided into three categories in Hajičová *et al.* (2003). Contrast 1 is the strongest type of contrast: it can be inferred that not only does the focus hold for this topic, but also that it does not hold for other entities if substituted for the contrastive topic. Here is an example:

$$\text{Your}^{t}\ \text{sister}^{c}\ \text{did}^{t}\ \text{very}^{f}\ \text{WELL}^{f}\ \text{in the test}^{t}. \tag{2.12}$$

With contrast, we may infer that there is somebody else (maybe the listener) who did not do well. If this kind of contrast is identified, we should add the inferred knowledge to the network. Without using contrastive topic, we can not draw such an inference. Contrastive types 2 and 3 are weaker forms of contrast which do not influence the resulting content. Their function lies mainly in structuring the content of utterance.

2.5 The Relationship to Tectogrammatical Layer

2.5.1 Nodes and Concepts

If we look at examples of TR and MultiNet structures, at first sight we can see that the nodes of TR mostly correspond to concepts in MultiNet. However, there is a major difference: TR does not include the concept encapsulation. The encapsulation in MultiNet serves for distinguishing definitional knowledge from assertional knowledge about a given node, e.g., in the sentence "The old man is sleeping", the connection to *old* will be in the definitional part of *man*, while the connection to the state *is sleeping* belongs to the assertional part of the concept representing the *man*. In TR, these differences in meaning are represented by differences in Topic-Focus Articulation (TFA) of the corresponding words.

There are also TR nodes that do not require a MultiNet concept (typically, the node representing the verb "be") and TR nodes corresponding to a whole subnetwork, e.g., *Fred* in the sentence "Fred is going home.", where the TR node representing *Fred* corresponds to the subnetwork[5] in Figure 2.6.

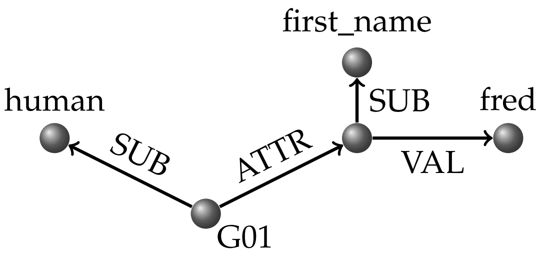


Figure 2.6: The MultiNet subnetwork corresponding to TR node representing *Fred*.

2.5.2 Edges, relations and functions

An edge of TR between nodes that have their conceptual counterparts in MultiNet always corresponds to one or more relations and possibly also some functions. In general, it can be said that the MultiNet representation of a text contains significantly more connections (either as relations, or as functions) than TR, and some of them correspond to TR edges.

2.5.3 Functors and types of relations and functions

There are 67 functor types in TR (see Mikulová *et al.* (2006) for description), which correspond to 94 relation types and 19 function types in MultiNet (Helbig, 2006). The mapping of TR functors to MultiNet is given in table 2.2:

TR functor	MultiNet counterpart
ACMP	*ITMS, *DIFF, ASSOC
ACT	AGT, CSTR, EXP, MEXP, SCAR
ADDR	AVRT, ORNT
ADVS	SUBST, OPPOS
AIM	PURP
APP	ASSOC, ATTCH
APPS	EQU, NAME

continued ...

[5]In fact the concept representing the man is the concept G01, i.e. only one vertex. However, the whole network corresponds to the TR node representing *Fred*.

TR functor	MultiNet counterpart
ATT	MODL
AUTH	AGT, ORIG
BEN	BENF
CAUS	CAUS, JUST, IMPL
CNCS	CONC
CM	*ITMS, MODL
COMPL	PROP except for sentential complements, PROPR, MCONT
COND	COND
CONFR	OPPOS
CONJ	*IMTS, *INTSC, *TUPL, *ALTN1, *UNION, *VEL1
CONTRA	OPPOS
CONTRD	CONC
CPR	*COMP
CRIT	METH, JUST, CIRC, CONF
CSQ	CAUS, JUST
DIFF	*MODP, *OP
DIR1	AVRT, ELMT, ORIGL, ORIG
DIR2	VIA
DIR3	DIRCL, ELMT, AFF
DISJ	*ALTN2, *VEL2
EFF	MCONT, PROP, RSLT
EXT	QMOD, *QUANT, *MODQ
HER	AVRT, ORIG
ID	NAME, PROP, ATTCH
INTT	PURP
LOC	LOC, LEXT, CTXT, RPRS
MANN	MANNR, METH, RPRS, *MODP, *MODS
MAT	ORIGM, *QUANT
MEANS	MODE, INSTR
MOD	MODL
OPER	*OP, TEMP
ORIG	AVRT, INIT, ORIGM, ORIG
PARTL	MODL
PAT	AFF, ATTCH, MCONT, OBJ, PROP, SSPE, SUB, SUPPL
PREC	REAS, OPPOS
REAS	CAUS

continued . . .

TR functor	MultiNet counterpart
REG	CONF, CTXT
RESL	CAUS, GOAL, RSLT
RESTR	*DIFF
RHEM	MODL, *MODQ
RSTR	PROP, PROPR, ATTR, QMOD, VAL, *PMOD, *QUANT
SUBS	SUBST
TFHL	DUR
TFRWH	TEMP
THL	DUR
THO	*QUANT
TOWH	TEMP
TPAR	TEMP, DUR
TSIN	STRT
TTILL	FIN
TWHEN	ANTE, TEMP

Table 2.2: Mapping of TR functors to MultiNet

There are also TR functors with no appropriate MultiNet counterpart: CPHR, DENOM, DPHR, FPHR, GRAD, INTF, PAR, PRED and VOCAT.

The functors summarized in Table 2.2 and their relationship to TR are explained in the following list:

ACMP

ACMP is the adjunct expressing accompaniment (in the broad sense of the word). MultiNet does not contain specific means for representing this language phenomena, it must be treated as a combination of ITMS and DIFF functions and possibly by an additional construction with the concept *"to accompany"*.

ACT

ACT is the actor of a verb. In MultiNet, there are several candidate cognitive roles associated with the actor: AGT – volitional agent (*"Mary hit the ball"*), CSTR – causator (*"The accident delayed the train"*), EXP – experiencer (*"The car broke"*), MEXP – mental experiencer (*"Joe saw a fish"*), SCAR – state carrier (*"The bread cost 1 Euro."*).

ADDR

ADDR is the addressee of the verb. In MultiNet, ORNT relation is used for the cognitive role of orientation to an object.

ADVS

ADVS is a paratactic structure root node – adversative relation. Its MultiNet counterparts are SUBST for substitution and OPPOS for opposition

AIM

AIM is an adjunct expressing purpose. MultiNet has the corresponding PURP relation.

APP

APP is an adnominal adjunct expressing appurtenance. This vague semantical relationship is in MultiNet usually expressed by ATTCH (attachment of objects) and sometimes also by ASSOC (unspecified association).

APPS

APPS is a root node of an appositional structure. APPS can be represented as intensional equality EQU, or by the shorthand relation NAME, which is not used in the annotation.

ATT

ATT is an atomic expression expressing the speaker's attitude. This kind of restriction is modeled using MODL MultiNet relation.

AUTH

AUTH is an adnominal adjunct referring to the author. This relationship can be expressed either by the action of the object creation with the author connected as the AGT of the action, or more vaguely as the origin of the object: ORIG.

BEN

BEN is an adjunct expressing that something is happening for the benefit (or disadvantage) of somebody or something. The MultiNet cognitive role BENF has the corresponding function.

CAUS

CAUS is an adjunct expressing the cause. MultiNet distinguishes two relation in this case: CAUS for causality and JUST for an ethical justification or a socially conventionalized reason for the situation (*"He doesn't smoke here because it is not allowed"*).

CNCS

CNCS is an adjunct expressing concession. The corresponding MultiNet relation is CONC.

CM

CM is a conjunction modifier. This category of language expressions is mostly neglected by MultiNet. It could be partially modeled by a combination of ITMS function and MODL relation.

COMPL

COMPL is an adjunct – predicative complement. It is represented by PROP (except for sentential complements) together with MCONT.

COND

COND is an adjunct expressing a condition. In MultiNet, there is the corresponding relation COND.

CONFR

CONFR is a paratactic structure root node – confrontation. In terms of MultiNet, it is a sign of an OPPOS relation between situations.

CONJ

CONJ is a paratactic structure root node – simple coordination/conjunction. MultiNet has two functions creating collections from individual elements: ITMS for unordered lists and TUPL for ordered lists.

CONTRA

CONTRA is a paratactic structure root node – two entities are in conflict (in a match, fight, etc.). The closest MultiNet counterpart is the OPPOS relation, although it is not clear how to use it directly.

CONTRD

CONTRD is an adjunct expressing confrontation. It is semantically similar to the situations where CONC relation is used.

CPR

CPR is an adjunct expressing comparison. MultiNet function COMP can be used in this context.

CRIT

CRIT is an adjunct expressing a criterion/measure/standard. Corresponding MultiNet relations are METH (method), JUST (justification, see CAUS above for an example), CIRC (circumstance) and CONF (conformity with an abstract frame).

CSQ

CSQ is a paratactic structure root node – consequential relation. MultiNet counterparts are CAUS (causal relationship), JUST (justification, see CAUS above for an example), and GOAL.

DIFF

DIFF is an adjunct expressing a difference (between two entities, states, etc.). MultiNet uses OP function for mathematical operations and MODP for modification of properties.

DIR1

DIR1 is a directional adjunct answering the question "*where from?*". It can be represented by ORIGL relation in the case of locations or by ORIG relation in the case of generalized locations.

DIR2

DIR2 is a directional adjunct answering the question "*which way?*". It can be represented using VIA relation.

DIR3

DIR3 is a directional adjunct answering the question "*where to?*". In the case of location, DIRCL relation can be used. There are also examples where ELMT relation is more appropriate (e.g., "*He belongs to our greatest specialists*").

DISJ

DISJ is a paratactic structure root node – disjunctive relation. MultiNet uses functions VEL2 for alternative situations and ALTN2 for other cases.

EFF

EFF is the effect of an action represented by a verb. It can be represented using MCONT (mental content, "*to hear the clock tick*"), PROP (property, "*They made each other's life bearable*"), and RSLT (result, "*She changed her hairstyle from curly hair to straight hair*").

EXT

EXT is an adjunct expressing extent. MultiNet contains QMOD relation (quantitative modification), QUANT attribute, and MODQ function (modification of a property by a quantity).

HER

HER is an adjunct expressing inheritance. MultiNet does not contain a specific relation for this purpose. The meaning can be partially captured using AVRT and ORIG relations.

ID

ID is a nominative of identity and explicative genitive. NAME relation can be used for the nominative of identity, while PROP and ATTCH relations can be used for the explicative genitive.

INTT

INTT is an adjunct expressing intention. The closest MultiNet counterpart is the PURP relation expressing the purpose of an action.

LOC

LOC is a locative adjunct answering the question "*where?*". In the case of locations, appropriate MultiNet relations are LOC (location) and LEXT (local extent). In more abstract contexts, CTXT relation can be used (e.g., "*We have serious loopholes in the educational field.*").

MANN

MANN is an adjunct expressing the manner. MultiNet distinguishes relations MANNR (manner) and METH (method).

MAT

MAT is an adnominal argument referring to the content of a container. MultiNet does not contain a specific relation for this case, but the relation ORIGM (material origin) can be used in some cases.

MEANS

MEANS is an adjunct expressing a means. In MultiNet, MODE (mode of a situation) and INSTR (instrument) relations can be used.

MOD

MOD is an atomic expression with a modal meaning. It corresponds to MultiNet MODL (modality of a situation) relation.

OPER

OPER is a paratactic structure root node referring to a mathematical operation or interval. Mathematical operators are expressed using OP functions, while intervals can be expressed using temporal relations such as TEMP (e.g., "*Monday to Friday*").

ORIG

ORIG is the origo argument of a verb. Its MultiNet counterparts are AVRT (averting of a situation from an object, e.g., "*They bought the machines from the producer*"), INIT (initial situation or entity, e.g., "*She remade the puppet from the jester into the devil*"), ORIGM (material origin, e.g., "*They made furniture out of wood*"), and ORIG (mental or informational origin, e.g., "*He heard about the accident from the teacher*").

PARTL

PARTL is an effective root node of an independent interjectional clause. It can be partially modeled by the MultiNet MODL relation.

PAT

PAT is the patient of the verb. It can be represented using the following MultiNet relations: AFF (affected object, "*He ate the soup.*"), ATTCH (attachment of objects, "*a chance of winning*"), GOAL ("*He reached the end*"), OBJ ("*He sold the house*"), PROP (property, "*George is good*"), SSPE (state specifier, "*He resembles his mother*"), and SUB (conceptual subordination, "*The cat is a mammal*").

PREC

PREC is an atomic expression referring to the preceding context. Depending on the type of expression, it should be represented using relations between situations, such as REAS (reason) and OPPOS (relation specifying an opposition).

REAS

REAS is a paratactic structure root node – causal relation. Causal relations are represented using CAUS (relation between cause and effect).

REG

REG is an adjunct expressing that something is asserted with respect to something else. If it is with respect to an abstract frame, CONF relation can be used. Otherwise, CTXT (contextual embedding of a situation) is appropriate.

RESL

RESL is an adjunct expressing the result/effect of something. Depending on the content of the expression, various MultiNet relations can be used: RSLT (result), CAUS (e.g., "*they pushed the car in such a way that it ended up in a ditch*"), GOAL (e.g., "*The author is trying to write it in such a way that anybody can read it*").

RESTR

RESTR is an adjunct expressing an exception / restriction. Exceptions can be formally represented as set differences (the exceptions are subtracted from the set of all relevant concepts). MultiNet contains the DIFF function for this purpose.

RHEM

RHEM is a rhematizer. The importance of rhematizers lies above all in the topic-focus articulation (see Section 2.5.5). If this kind of treatment does not suffice to capture the content, MODL relation can be used.

RSTR

RSTR is an adnominal adjunct modifying its governing noun. It expresses PROP (relation between object and a property), ATTR relation (assignment of attributes to objects, e.g., "*The tree is 4 m in height*"), or PMOD function (modification of object by associative properties, e.g., "*philosophical dictionary*").

SUBS

SUBS is an adjunct expressing that somebody or something is substituted for somebody or something else. It corresponds to SUBST relation (relation specifying a substitute for an entity).

TFHL

TFHL is a temporal adjunct answering the question *"for how long?"* It corresponds to the DUR relation (relation specifying a temporal extension).

TFRWH

TFRWH is a temporal adjunct answering the question *"from when?"* It can be represented by the MultiNet TEMP relation (Relation specifying a temporal frame).

THL

THL is a temporal adjunct answering the questions *"how long?"* and *"after how long?"*. MultiNet relation DUR (relation specifying the temporal extension) can be used to represent it.

THO

THO is a temporal adjunct answering the questions *"how often?"* and *"how many times?"*. MultiNet attribute QUANT for the relevant situation should be able to represent the meaning.

TOWH

TOWH is a temporal adjunct answering the question *"to when?"*. In MultiNet, we can use TEMP relation in combination with the RSLT in cases like e.g., *"He postponed the class to Friday."*.

TPAR

TPAR is a temporal adjunct answering the questions *"in parallel/simultaneously with what?"* and *"during what time?"*. As other purely temporal natural language expressions, these can be represented by an interplay of TEMP and DUR relations.

TSIN

TSIN is a temporal adjunct answering the question *"since when?"*. MultiNet has a special relation for this purpose: STRT (relation specifying the temporal beginning).

TTILL

TTILL is a temporal adjunct answering the question *"until when?"*. Its MultiNet counterpart is the FIN relation specifying the temporal end.

TWHEN

TWHEN is a temporal adjunct answering the question *"when?"*. An expression like this can be represented using MultiNet relation TEMP.

Table 2.3 shows the mapping from MultiNet relations to TR functors (a star denotes a MultiNet function as opposed to MultiNet relation). The mapping is the inverse with respect to the mapping in Table 2.2.

MultiNet	**TR counterpart**
Relations:	
AFF	PAT, DIR3
AGT	ACT, AUTH
ANTE	TWHEN
ASSOC	ACMP, APP
ATTCH	APP, ID, PAT
ATTR	RSTR
AVRT	ORIG, HER, ADDR, DIR1
BENF	BEN
CAUS	CAUS, CSQ, RESL, REAS
CIRC	CRIT
CONC	CNCS, CONTRD
COND	COND
CONF	REG, CRIT
CSTR	ACT
CTXT	REG, LOC
DIRCL	DIR3
DUR	TFHL, THL, TPAR
ELMT	DIR3, DIR1
EQU	APPS
EXP	ACT
FIN	TTILL
GOAL	see RSLT, DIRCL and PURP
IMPL	CAUS

continued . . .

MultiNet	TR counterpart
INIT	ORIG
INSTR	MEANS
JUST	CAUS, CRIT, CSQ
LEXT	LOC
LOC	LOC
MANNR	MANN
MCONT	PAT, EFF, COMPL
MERO	see PARS, ORIGM, *ELMT, *SUBM and TEMP
METH	MANN, CRIT
MEXP	ACT
MODE	see INSTR, METH and MANNR
MODL	MOD, ATT, CM, PARTL, RHEM
NAME	ID, APPS
OBJ	PAT
OPPOS	CONTRA, ADVS, CONFR, PREC
ORIG	ORIG, DIR1, AUTH, HER
ORIGL	DIR1
ORIGM	ORIG, MAT
ORNT	ADDR
PROP	COMPL, EFF, PAT, RSTR, ID
PROPR	COMPL, RSTR
PURP	AIM, INTT
QMOD	EXT, RSTR
REAS	see CAUS, JUST and IMPL
RPRS	LOC, MANN
RSLT	EFF, RESL
SCAR	ACT
SITU	see CIRC and CTXT
SOURC	see INIT, ORIG, ORIGL, ORIGM and AVRT
SSPE	PAT
STRT	TSIN
SUB	PAT
SUBST	SUBS, ADVS
SUPPL	PAT
TEMP	TWHEN, TFRWH, TOWH, TPAR
VAL	RSTR

continued . . .

MultiNet	**TR counterpart**
VIA	DIR2
Functions:	
*ALTN1	CONJ
*ALTN2	DISJ
*COMP	CPR, grammateme DEGCMP
*DIFF	RESTR, ACMP
*INTSC	CONJ
*ITMS	CONJ, ACMP, CM
*MODP	MANN, DIFF
*MODQ	RHEM, EXT
*MODS	MANN
*NON	grammateme NEGATION
*OP	DIFF, OPER
*ORD	grammateme NUMERTYPE
*PMOD	RSTR
*QUANT	EXT, MAT, RSTR, THO
*SUPL	grammateme DEGCMP
*TUPL	CONJ
*UNION	CONJ
*VEL1	CONJ
*VEL2	DISJ

Table 2.3: Mapping of MultiNet relations to TR functors and other representational means.

There are also MultiNet relations and functions with no counterpart in TR (stars at the beginning denote a function): ANLG, ANTO, CHEA, CHPA, CHPE, CHPS, CHSA CHSP, CNVRS, COMPL, CONTR, CORR, DISTG, DPND, EXT, HSIT, MAJ, MIN, PARS, POSS, PRED0, PRED, PREDR, PREDS, SETOF, SYNO, VALR, and *FLPJ.

From the tables 2.2 and 2.3, we can conclude that although the mapping is not one to one, the preprocessing of the input text to TR highly reduces the problem of the appropriate text to MultiNet transformation. However, it is not clear how to solve the remaining ambiguity.

2.5.4 Grammatemes and layer information

TR has at its disposal 15 grammatemes, which can be conceived as node attributes. Note that not all grammatemes are applicable to all nodes. The grammatemes in TR roughly correspond to layer information in MultiNet, but also to specific MultiNet relations.

1. NUMBER. This TR grammateme is transformed to QUANT, CARD, and ETYPE attributes in MultiNet.

2. GENDER. This syntactical information is not transformed to the semantic representation with the exception of occurrences where the grammateme distinguishes the gender of an animal or a person and where MultiNet uses SUB relation with appropriate concepts.

3. PERSON. This grammateme is reflected in cognitive roles connected to the event or state and is semantically superfluous.

4. POLITENESS has no structural counterpart in MultiNet. It can be represented in the conceptual hierarchy of SUB relation.

5. NUMERTYPE distinguishing e.g. "three" from "third" and "one third" is transformed to the way in which this number is connected to the network.

6. INDEFTYPE corresponds to QUANT and VARIA layer attributes.

7. NEGATION is transformed to both FACT layer attribute and *NON function combined with modality relation MODL.

8. DEGCMP corresponds to *COMP and *SUPL functions.

9. VERBMOD: *imp* value is represented by MODL relation to imperative, *cdn* value is ambiguous not only with respect to facticity of the condition but also with regard to other criteria distinguishing CAUS, IMPL, JUST and COND relations which can all result in a sentence with a *cdn* verb. Also the FACT layer attribute of several concepts is affected by this value.

10. DEONTMOD corresponds to MODL relation.

11. DISPMOD is a purely syntactical grammateme and has no counterpart in MultiNet.

12. ASPECT has no direct counterpart in MultiNet. It can be represented by the interplay of temporal specification and RSLT relation connecting an action to its result.

13. TENSE is represented by relations ANTE, TEMP, DUR, STRT, and FIN.

14. RESULTATIVE has no direct counterpart and must be expressed using the RSLT relation.
15. ITERATIVENESS should be represented by a combination of DUR and TEMP relations where a temporal concept has QUANT layer information set to *several*.

2.5.5 TFA, quantifiers, and encapsulation

In TR, the information structure of every utterance is annotated in terms of Topic-Focus Articulation (TFA):

1. Every autosemantic word is marked `c`, `t`, or `f` for contrastive topic, topic, or focus, respectively. The values can distinguish which part of the sentence belongs to topic and which part to focus.
2. There is an ordering of all nodes according to communicative dynamism (CD). Nodes with lower values of CD belong to topic and nodes with greater values to focus. In this way, the degree of "aboutness" is distinguished even inside the topic and the focus of sentences.

MultiNet, on the other hand, does not contain any representational means devoted directly to representation of the information structure. Nevertheless, the differences in the content of sentences differing only in TFA can be represented in MultiNet by other means. The TFA differences can be reflected in these categories:

- Relations connecting the topic of the sentence with the remaining concepts in the sentence are usually a part of definitional knowledge about the concepts in the topic, while the relations going to the focus belong to the assertional part of knowledge about the concepts in the focus. In other words, TFA can be reflected in different values of K_TYPE attribute.
- TFA has an effect on the identification of presuppositions (Peregrin, 1995a) and allegations (Hajičová, 1984). In the case of presuppositions, we need to know about them in the process of assimilation of new information into the existing network in order to detect presupposition failures. In the case of allegation, there is a difference in the FACT attribute of the allegation.
- The TFA has an influence on the scope of quantifiers (Peregrin, 1995b; Hajičová *et al.*, 1998). This information is fully transformed into the quantifier scopes in MultiNet.

2.6 Existing Tools

2.6.1 MWR

MWR (MultiNet Wissens-Repräsentation[6]) is a tool allowing the user to parse a sentence using the Wocadi parser (see Section 2.6.2) and to visualize and edit the resulting MultiNet network (Gnörlich, 2000). It provides convenient and quick access to the parser that translates from German sentences or phrases to MultiNets. However, this tool has not been designed primarily for annotation and the visualization largely prefers compactness of the network by hiding all the attributes from the user. Also, intersentential links are difficult to create.

2.6.2 Wocadi

Wocadi is a rule-based parser of German sentences into the MultiNet semantic networks (Helbig, 1986; Helbig and Hartrumpf, 1997). It relies heavily on the dictionary HaGenLex (see Section 2.6.3). The parser is hard to evaluate because of the non-existence of gold-standard data. The only parameter measured has been the parser coverage. However, the parsing and MultiNet itself proved to be a viable tool in the CLEF 2007 answer validation task for German (Hartrumpf *et al.*, 2007).

2.6.3 HagenLex

There is a semantically oriented computer lexicon HaGenLex (Hartrumpf *et al.*, 2003), which is also used in the Wocadi parser (see Section 2.6.2). English and German search is supported, with outputs in English (around 3,000 semantic networks) and/or German (more than 25,000 semantic networks, currently). An example sentence for the German verb "borgen.1.1" ("borrow") and its automatically generated and validated semantic representation is displayed in Figure 2.7. The quality of example parses is assured by comparing the marked-up complements in the example to the ones in the semantic network. In the rare case that the parse is not optimal, it will not be visible to annotators.

[6] En: MultiNet Knowledge Representation

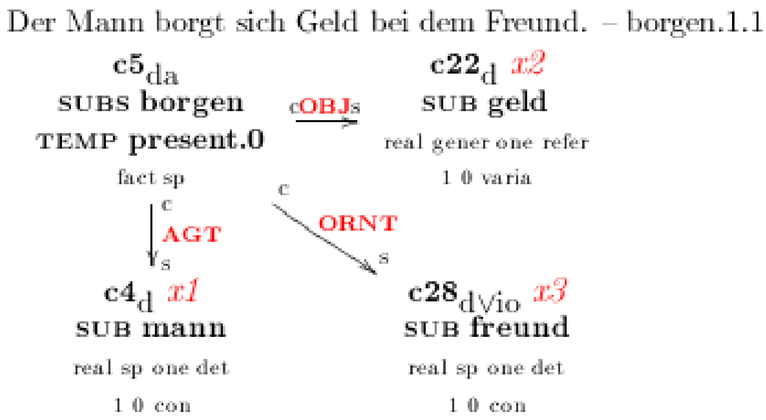


Figure 2.7: HaGenLex entry showing an example sentence for the German verb "borgen.1.1" ("borrow"). The sentence is literally "The man borrows himself money from the friend."

2.7 PML Format of MultiNet

The Prague Dependency Treebank native format is an XML based PML: Prague Markup Language (Pajas and Štěpánek, 2005). The MultiNet annotation layer format is described by its XML schema, which is included in the annotation tool *cedit*[7]. The format of the file is quite simple. An example of a simple file containing three nodes is shown here[8] :

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<MNetwork xmlns="http://ufal.mff.cuni.cz/pdt/pml/">
    <head>
        <schema href="kdata.xml"/>
        <references>
            <reffile href="wsj_0020.t" id="t" name="tdata"/>
        </references>
    </head>
    <body>
        <MConcept id="C1">
            <fact>real</fact>
            <gener>sp</gener>
            <sort>ad</sort>
            <trMapping>
                <LM>
                    <trReference>t#EnglishT-wsj_0020-s3-t25</trReference>
                    <left>49</left>
                    <top>356</top>
                </LM>
                <LM>
                    <trReference>t#EnglishT-wsj_0020-s4</trReference>
```

[7] `resources/cdata.xml` in the distribution

[8] The network only demonstrates the file format and does not make any sense, e.g., the sort of C2 is incompatible with ANTE relation.

```
22                     <left>598</left>
23                     <top>588</top>
24                 </LM>
25             </trMapping>
26         </MConcept>
27         <MConcept id="C2">
28             <sort>aq</sort>
29             <trMapping>
30                 <trReference>t#EnglishT-wsj_0020-s3-t27</trReference>
31                 <left>0</left>
32                 <top>539</top>
33             </trMapping>
34         </MConcept>
35         <MFunction id="F3">
36             <fact>hypo</fact>
37             <gener>gen</gener>
38             <sort>ad</sort>
39             <ftype>ALTN1</ftype>
40             <arguments.rf>
41                 <LM>C2</LM>
42                 <LM>C1</LM>
43             </arguments.rf>
44             <trMapping>
45                 <trReference>t#EnglishT-wsj_0020-s3</trReference>
46                 <left>352</left>
47                 <top>399</top>
48             </trMapping>
49         </MFunction>
50         <MRelation id="R4">
51             <sort>si</sort>
52             <rtype>ANTE</rtype>
53             <goingFrom.rf>C1</goingFrom.rf>
54             <goingTo.rf>C2</goingTo.rf>
55         </MRelation>
56     </body>
57 </MNetwork>
```

The file contains a preamble (1-8) identifying the file containing the tectogrammatical annotation (6). The network contains two regular concepts C1 and C2, a function concept F3 and a relation R4. The concept C1 (10-26) is of sort *ad* (13) and it is used in two sentences (16, 21) with different graphical coordinates (17-18, 22-23). The concept C2 (27-34) of sort *aq* (28) has no further attributes and is used only in one sentence (29-33). The function F3 (35-49) is an ALTN1 function (39). It has two arguments. The first argument is C2 (41) and the second argument is C1 (42). There is an edge going from C1 to C2. It represents an ANTE relation (50-55).

The order of concepts, functions and relations in the file is not important. In case of metaedges there is nothing complicated going on in the file. One of the references (as 53 or 54) would be referring to another relation instead of a concept.

Chapter 3

Manual Annotation

> Given the two ideas 'fat' and 'lean', would you be rather inclined to say that Wednesday is fat and Tuesday lean, or the other way round? (I incline to choose the former.) Now have "fat" and "lean" some different meaning here from their usual one?—They have a different use.—So ought I really to have used different words? Certainly not that.—I want to use *these* words (with their familiar meanings) here.—Now, I say nothing about the causes of this phenomenon. They *might* be associations from my childhood. But that is a hypothesis. Whatever the explanations,—the inclination is there.
>
> Asked "What do you really mean here by 'fat' and 'lean'?"—I could only explain the meanings in the usual way. I could *not* point to the examples of Tuesday and Wednesday.
>
> Wittgenstein (1953, II/xi)

The MultiNet annotation task comprises several stages. First, the annotator typically starts with an empty graph. He or she can see the list of all concepts used in the current article, the tectogrammatical tree, and the sentence itself (see the annotation tool main screen in Figure 3.6 on page 72). It is possible to navigate through sentences and search them. The annotators' task is to:

1. create nodes of the network corresponding to the tectogrammatical nodes,

2. create any function nodes necessary and determine the function arguments,

3. determine and fill in the sort of each node,
4. connect nodes and function nodes with appropriate relations creating a connected graph (connected in the undirected sense),
5. find the concepts which have already occurred in the previous sentences, reuse them in the current sentence and connect to the referring nodes,
6. fill in all the remaining attributes of all nodes.

The annotator repeats this for every sentence. In the case of the first sentence it is necessary to create indexical referents such as "now" "I", "you", and "here" (Jakobson, 1971; Montague, 1972; Kaplan, 1979). In practice, however, the newspaper sentences in our initial sample never contain any mention of these concepts except for implicit relations to "now". This concept is created for the first sentence and later it is subject to coreference.

In effect, the whole article is one semantic network, although the annotator can always see only the part relevant to the current sentence (Novák, 2007b). To avoid confusion, every concept displays its unique concept identification number.

3.1 Related Annotation Projects

Manual annotation in the realm of textual data began with simple linguistic annotation and is becoming still more complex. The annotation paradigms range from linear, such as part of speech tagging (Kučera, 1980; Hajič *et al.*, 2001) and named entity tagging (Tanabe *et al.*, 2005), to structured annotations. The structural annotation can be divided into syntax-based and semantics-based. In the case of the syntax based project, one can annotate the constituent structure (Marcus *et al.*, 1993) or the dependency structure (Mikulová *et al.*, 2006; Hajič *et al.*, 2006; Čmejrek *et al.*, 2004; Hajič *et al.*, 2004). The semantic annotation projects have always been grounded in linguistic origins and widely inspired by Fillmore (1968, 1977) and Jackendoff (1990). In these projects the main attention is paid to the verbs and their arguments (Kingsbury *et al.*, 2002; Lopatková *et al.*, 2006) and there are separate projects annotating the noun structure (Meyers *et al.*, 2004).

It is notable that no other logic-based formalism has been subject to this kind of annotation effort. There has been a huge gap between the projects listed above and the semantics as it is used in the field of artificial intelligence and discussed in Section 1.4 on page 23.

3.2 Tools

The MultiNet annotation process for texts already annotated on the tectogrammatical layer is supported by various tools introduced in this section. The annotators are free to use all the tools and resources described below to produce the appropriate markup (Novák, 2007a).

3.2.1 Annotation Tool *Cedit*

The core annotation is facilitated by the *cedit* tool, which uses PML (Pajas and Štěpánek, 2005), an XML file format, as its internal representation (Novák, 2007a). The annotation tool is an application with a graphical user interface (GUI) implemented in Java™SDK 6 (Sun Microsystems, Inc., 2007) using the Swing GUI (Elliott *et al.*, 2002). The *cedit* tool is platform independent and directly connected to the annotators' wiki (see Section 3.2.4), where the annotators can access the definitions of individual MultiNet relations, functions and attributes, as well as examples, counterexamples and discussion concerning the entity in question. If the wiki page does not contain the required information, the annotator is encouraged to edit the page with his/her questions and comments. The overall *cedit* frame layout is derived from the Tred tool (Hajič *et al.*, 2001) to facilitate the interoperability of annotation tools for different layers of PDT.

The *Cedit* Package Structure

The package structure of *cedit* is as follows:

- `cdata` – package containing API for loading, manipulating and storing the semantic networks.
- `cedit` – package containing the annotation graphical user interface (GUI)
  - `cedit.components` – subpackage containing graphical components of GUI (classes inheriting from `javax.swing.JComponent`).
  - `cedit.dialogs` – subpackage containing dialogs used in the GUI (classes inheriting from `javax.swing.JDialog`).
- `tdata` – package containing API for loading and representing tectogrammatical trees and the underlying sentences.

- `utils` – package with various utilities (logging, spline curves, loading and saving of user preferences)
  - `utils.eval` – subpackage containing classes for evaluation of annotator agreement as described in Chapter 4 on page 75.

Cedit Features

The *cedit* GUI is displayed in Figure 3.6 on page 72. The annotator can see the sentence in the upper part of the frame. The corresponding tectogrammatical tree with collapsed multiword expressions (Bejček *et al.*, 2006) is available on the left side. The right side of the frame is occupied by the list of all used concepts with a text field for quick search. There is a toolbar and a status bar on the top and on the bottom of the frame, respectively. The central part displays the editable semantic network. Nodes are connected with the edges and there are also editable links to the tectogrammatical tree. The available keyboard shortcuts are presented in Table 3.1 and the mouse gestures are listed in Table 3.2 on page 73.

All the concept attributes (see Section 2.2.1 on page 29) are always visible and editable with a canonical positioning of the attributes as displayed in Figure 3.1. The attributes are automatically hidden and shown as needed according to the concept sort.

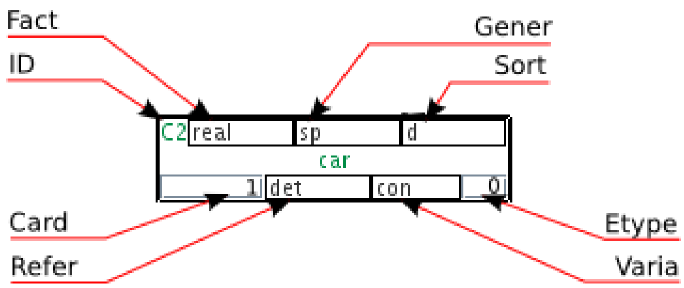


Figure 3.1: MultiNet concept displayed in *cedit*. Depending on the concept sort, the upper row, both rows, or only the sort is displayed and available to the annotator.

3.2.2 Online Lexicon

The annotators in the semantic annotation project have the option to look up examples of MultiNet structures in an online version of the semantically

Key	Functionality
Enter	Display the dialog with the attributes of the selected node
Up	Select the next graph element on the way up
Down	Select the next graph element on the way down
Left	Select the next graph element to the left
Right	Select the next graph element to the right
Escape	Deselect the selected node
Backspace	Select previously selected graph element
Numpad 1	Select the next edge in the clockwise direction
Numpad 0	Select the next edge in the counter-clockwise direction
F4	Select an element in the network
Shift + Left	Move node 10 pixels to the left
Shift + Up	Move node 10 pixels up
Shift + Right	Move node 10 pixels to the right
Shift + Down	Move node 10 pixels down
Ctrl + Shift + Left	Move node 1 pixel to the left
Ctrl + Shift + Up	Move node 1 pixel up
Ctrl + Shift + Right	Move node 1 pixel to the right
Ctrl + Shift + Down	Move node 1 pixel down
Alt + o	Open a network file
Ctrl + s	Save the network
Ctrl + a	Save the network as a new file
Ctrl + e	Export a PNG image of the network of the current sentence
Ctrl + w	Close the network file
Alt + q	Quit *cedit*
Ctrl + z	Undo
Ctrl + Shift + z	Redo
Insert	Create new node
Shift + Insert	Create new function node
Delete	Delete the selected node or edge
Alt + Right	Go to the next sentence
Alt + Left	Go to the previous sentence

Table 3.1: List of keyboard shortcuts in *cedit*.

oriented computer lexicon HaGenLex (see Section 2.6.3 on page 57). The annotators can use lemmata (instead of entering IDs formed by the lemma and a numerical suffix) for the look-up. This increases the recall of the dictionary look-up.

3.2.3 Online Parser

Sometimes the annotator needs to look up a phrase or something more general than a particular noun or verb. In this case, the annotator can use the workbench for (MultiNet) knowledge bases (MWR, see Section 2.6.1 on page 57). The English version of the parser will be connected in the near future, too.

3.2.4 Wiki Knowledge Base

A wiki (Leuf and Cunningham, 2001) is used collaboratively to create and maintain the knowledge base used by all the annotators. In this project we use Dokuwiki (Badger, 2007). The entries of individual annotators in the wiki are logged and a feed of changes can be observed using an RSS reader. The *cedit* annotation tool allows users to display appropriate wiki pages of individual relation types, function types and attributes directly from the tool using their preferred web browser by right-clicking the item in a combo box.

3.3 Problems and Solutions

Although some relationships between elements of FGD and elements of MultiNet are quite straightforward (Novák, 2006), there remain a number of problems associated with the MultiNet annotation.

What is real?

Every MultiNet concept of SORT subsumed under the sort *o*, *si*, *t*, or *l* has the FACT attribute, distinguishing real object from nonexistent and hypothetical objects. Now consider the sentence "*The flight has been canceled.*" Does the flight exist? In other words, given a question like "Is there a flight from A to B?", what is the correct answer?

a. "Yes, there is."

b. "Yes, but it has been canceled."

c. "No, there is no flight."

d. "No, the flight has been canceled."

It seems that answers a. and c. are not quite appropriate, but the preference of either b. or d. is not so clear and in such a case we may well obtain different values of the FACT attribute of "flight" from different people: those inclined to answer with b. would assign FACT=real and those inclined to d. would assign FACT=hypo or FACT=non.

Sort restrictions

Every MultiNet relation and every function has its *signature* which puts constraints on the properties of the involved concepts. These constraints are usually expressed as restrictions on the sorts.

For example, the ITMS function requires all its arguments to be of the same sort. This poses, however, a problem in the case of real-world sentences:

She said there is "growing realization" around the world that denial of intellectual-property rights harms all trading nations, and particularly the "creativity and inventiveness of an (offending) country's own citizens."

In the above sentence, the author puts the "creativity and inventiveness" into the same collection as "nations", but these two clearly do not share the same sort. The solution we use is to ignore the signature in these cases. This, however, puts an extra burden on annotators, because it is in a conflict with the instruction to always conform to the signatures.

Unexpected modifications

The expression "task force" would normally be annotated as a modification by an associative property (see Figure 3.2 on page 68). However, when the expression changes to "intellectual-property task force", this approach is no longer possible, because the task is of the sort *aq* (associative quality), while the function MODP for modification of properties allows us to modify only qualities subsumed under the sort *p* (properties in the broader sense). The desired solution is depicted in Figure 3.3, but it is violating the current restrictions.

Similarly, the annotation for "compatible law" will have to be substantially different than for "a law compatible with international standards" due to the modification of "compatible".

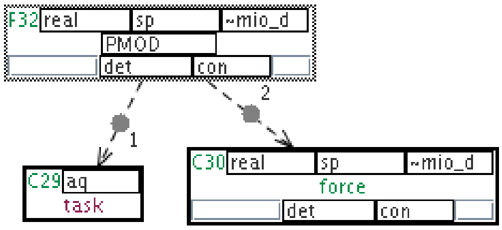


Figure 3.2: Annotation of the "task force". The force is modified by the associative property "task". The concept F32 represents the "task force".

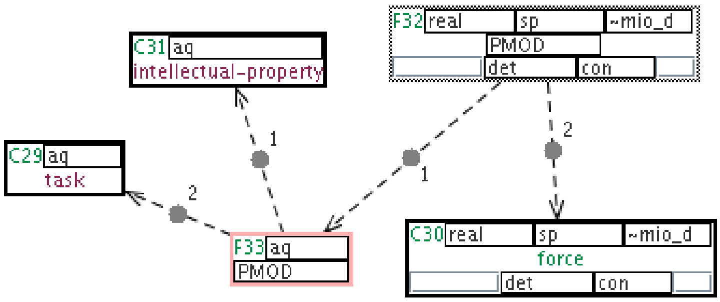


Figure 3.3: Annotation of the "intellectual-property task force". The "task" is modified by "intellectual-property", which violates the constraints on MODP (the color of the node indicates the violation to the annotator).

Temporal fuzziness

There are a number of MultiNet representational means devoted to the temporal ordering of situations, namely ANTE, FIN, STRT and DUR relations. They, however, fail to represent the vague temporal specifications of natural language, such as "earlier this year". This expression certainly does not mean yesterday, almost surely not last week and probably not last month. The natural way to represent this would be a probability density function over dates in the year, but this is not supported in MultiNet, and furthermore it is virtually impossible to annotate on a larger scale.

Vague references

"For instance" is a very common expression in *The Wall Street Journal*. It means that the following situation or a participant in the situation is an example of something already mentioned. There are two problems associated with this. First, the "something already mentioned" may be a whole sentence or paragraph. Ellipsis is also quite common (e.g., "*Banks seem to love their clients. For instance, HSBC spokesman said that the amount spent on customer loyalty programs has tripled over the last 16 months.. . .*"). Second, there are two different ways to annotate expressions like this. Consider a sentence like "*For instance, John loves Mary.*", where we know that "for instance" refers to some emotions represented by the concept with ID=C23. Either we can consider "instance" as a state "being an instance of something" and "love" is in this state (Figure 3.4), or we can say that "love" simply is the same object as the "instance" and use the EQU relation (Figure 3.5). It is difficult to come up with a rule which would consistently produce only one annotation.

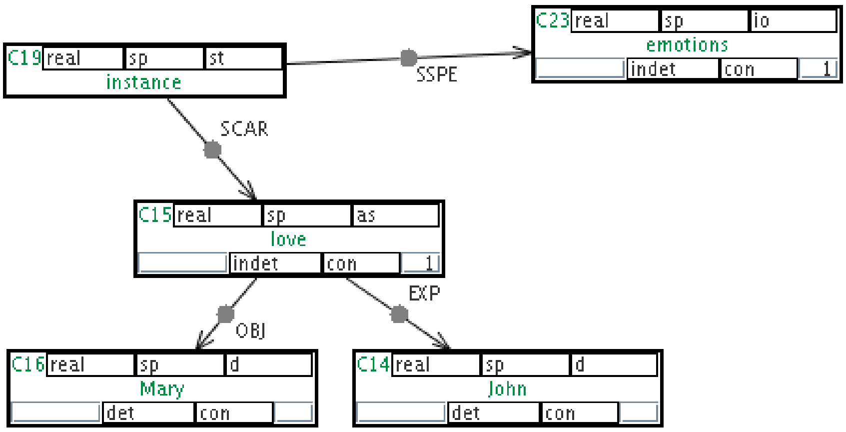


Figure 3.4: "For instance, John loves Mary." Love is the State CARrier of the state "to be an instance of".

Another example of a vague expression is the sentence *Mr. Stronach will direct an effort to reduce overhead and curb capital spending "until a more satisfactory level of profit is achieved and maintained," Magna said.* The expression "more satisfactory level" is annotated using the COMP function for comparatives. This function takes a property and an object, with which we compare. The property here is "satisfactory", but what is the level we are comparing with? Is it the current level of profit, the planned level of profit or the level of profit that would be achieved without Mr. Stronach's effort?

Sometimes, we have difficulty with seemingly unambiguous expressions. For example, in "next spring", the "next" is an indexical expression and requires an anchor. But is the "next spring" "next" with respect to "today", "this month", "this year", or "this spring"? It is hard to standardize the annotation of these kinds of expressions.

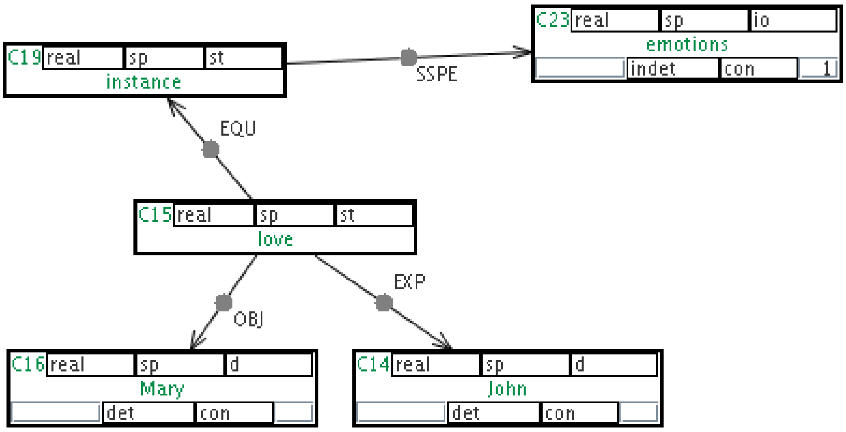


Figure 3.5: "For instance, John loves Mary." Love is an instance of emotions. The state C19 has no SCAR.

Quantities

MultiNet has systematic means to represent quantities of objects. It is tempting to use these for annotation of an expression like "*$ 570 million mortgage-backed securities mutual fund*". Here, however, the fund is only one. The amount is something related to the fund. One annotator can create an "assets" node, another one can add an attribute "value" or "size". Again, it is difficult to create a rule which would help all annotators to produce the same annotation.

Another example of a quantity-related problem is the expression "*the countries she placed under varying degrees of scrutiny*", where there is a mismatch between the syntactical and semantical structure. What is worse, the plural of "degrees" suggests that there were more degrees. At the same time, it is not true that all these degrees were "varying". It is then a question of how to represent the word "varying".

Specificity vs. Genericity

The evaluation shows (see Section 4.5.1 on page 89) that the agreement on the degree of genericity is very problematic. This is very likely caused by the vagueness of the definition (Helbig, 2006, p. 420):

> The attribute value [GENER = ge] specifies only that the concept in question has the aspect of generalization as a meaning component and thus, in contrast to concepts characterized by

[GENER = sp], such a concept does not apply to a special element or a special group of elements.

The examples associated with the definition always discuss discreet objects such as dogs, cars and men. Situations and abstractions from situations are not discussed. The typical confusing example is “certificate” in an expression like “certificate policy”.

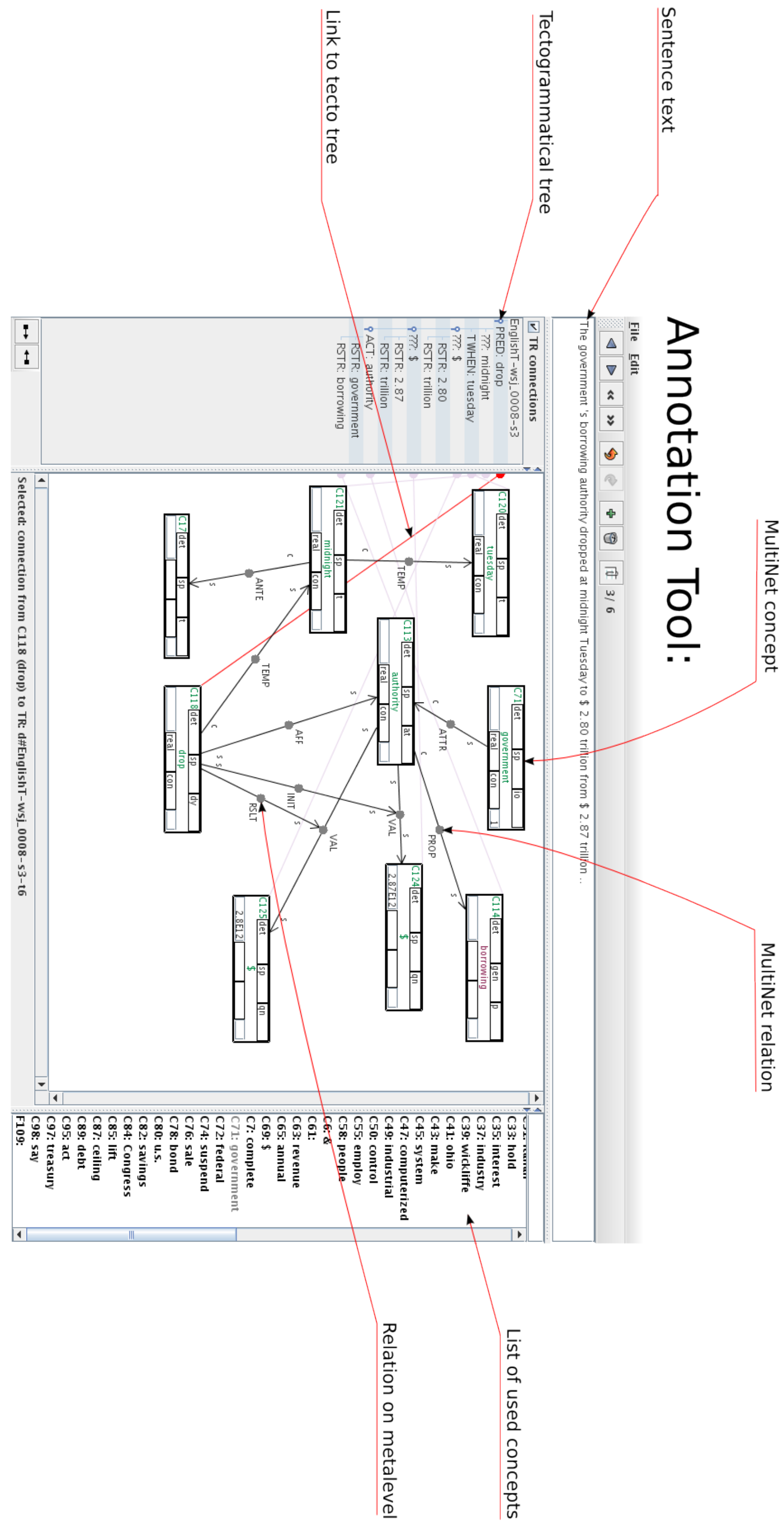


Figure 3.6: The main editor frame of *cedit*.

Gesture	Source	Target	Action
Double click		Tectogrammatical node	Create new node connected to this tectogrammatical node
Shift + Double click		Tectogrammatical node	Create new function node connected to this tectogrammatical node
Drag	Tectogrammatical node handle	Tectogrammatical node	Change the link between a node and the tectogrammatical tree
Drag	Tectogrammatical node handle	Corresponding graph node or edge	Delete the link between the node and the tectogrammatical tree
Double click		Concept list	Reuse the concept in this sentence
Double click		Tectogrammatical node handle	Toggle show TR connections
Drag	Node or Function node	Free space in the graph pane	Move the node
Drag	Edge	Free space in the graph pane	Change the curve of the edge
Drag	Node	Node	Create a new straight relation edge
Drag	Function Node	Node	Add an argument to the function
Ctrl + Drag	Function Node	Node	Create a new straight relation edge
Drag	Edge	Originating node	Delete the edge
Drag	Edge	Other Node	Change the target of the relation
Ctrl + Drag	Edge	Other Node	Create a metaedge going from the edge
Ctrl + Shift + Drag	Edge	Other Node	Change the source node of the relation
Ctrl + Drag	Node	Edge	Create a metaedge going to the edge
Drag	Node or edge	Tectogrammatical tree	Create a link from the node or edge to the tree or change an existing one
Drag	Node	Concept List	Hide the node in the concept list
Double click		Node or edge	Display the dialog with the attributes of the selected element
Right click		Item in a combo box	Display the wiki page with the contextual help

Table 3.2: List of available mouse editing actions in *cedit*.

Chapter 4

Evaluation

> If someone were to draw a sharp boundary I could not acknowledge it as the one that I too always wanted to draw, or had drawn in my mind. For I did not want to draw one at all. His concept can then be said to be not the same as mine, but akin to it. The kinship is that of two pictures, one of which consists of colour patches with vague contours, and the other of patches similarly shaped and distributed, but with clear contours. The kinship is just as undeniable as the difference.
>
> Wittgenstein (1953, I/§76)

The natural question to ask in the realm of manual annotation is the following:

What is the quality of the annotation?

It turns out that this question is extremely difficult to answer in a principled way. In order to measure the quality of one network it is necessary to have the *gold standard annotation* at hand. In absence of such a gold standard, the next natural option is to measure the consistency of the annotation process by *inter-annotator agreement*.

4.1 Evaluation Metrics

Human annotations are usually evaluated against each other to measure the consistency of the annotation. The most common measures of agreement are accuracy (number of correct decisions divided by the number of all decisions) and F-measure (harmonic mean between the recall and the precision). However, these approaches suffer from the fact that some annotation agreement is present simply by chance. This fact was the reason to propose annotation agreement metrics corrected for the agreement by chance. First, the Scott's π (Scott, 1955) and Cohen's κ (Cohen, 1960) were introduced. They were later generalized to the K coefficient of agreement (Carletta, 1996).

For three reasons I do not use any of these corrections in this thesis:

1. The agreement metrics itself is difficult to develop and as noted in Chapter 5, to obtain the most appropriate agreement score there is still much to do.

2. The agreement by chance is difficult to compute in such a complex situation. The probability that two annotators will produce exactly the same oriented graph with the same size and all the attributes is virtually zero.

3. The measures have no clear probabilistic interpretation (Artstein and Poesio, 2007).

When a stable level of annotator agreement is achieved and maintained, and the agreement measure is robust with respect to equivalent annotations, the metrics extended for hierarchical values should be used. An example is Krippendorf's α (Krippendorff, 1980).

4.2 Evaluation Data

The initial evaluation presented in this section has been carried out on a portion of *The Wall Street Journal* articles from the Penn Treebank (Marcus *et al.*, 1993), which have been annotated on all the FGD layers and are available as the Prague English Dependency Treebank (Hajič *et al.*, Est 2009). Initially, some sentences were used during the training of annotators. These sentences were removed from the evaluation sample. The evaluation sample contains 67 annotated sentences (1793 words), annotated by two annotators, of which 46 sentences (1236 words) were annotated by three independent annotators. All annotators are native English speakers.

4.3 Structural Agreement

The structural agreement is measured for every sentence in isolation in two steps. First, the best match between the two annotators' graphs is found. Most of the graph nodes are connected to the tectogrammatical tree (see Figure 4.8 on page 97) and for the remaining nodes, all possible one-to-one mappings are constructed and the optimal mapping w.r.t. the F-measure is selected. Second, the optimal mapping is used to compute the agreement.

Formally, we start with a set of tectogrammatical trees containing a set of nodes N. The annotation is a tuple $G = (V, E, T, A)$, where V are the vertices, $E \subseteq V \times V \times P$ are the directed edges and their labels (e.g., agent of an action: AGT $\in P$), $T \subseteq V \times N$ is the mapping from vertices to the tectogrammatical nodes, and finally A are attributes of the nodes. We simplified the problem by ignoring the mapping from edges to tectogrammatical nodes, the metaedges, and the MultiNet edge attribute *knowledge type*. Analogously, $G' = (V', E', T', A')$ is another annotation of the same sentence and our goal is to measure the similarity $s(G, G') \in [0, 1]$ of G and G'.

To measure the similarity we need a set Φ of admissible one to one mappings between vertices in the two annotations. A mapping is admissible if it connects vertices which are indicated by the annotators as representing the same tectogrammatical node:

$$\begin{aligned}
\Phi \;=\; & \Big\{ \phi \subseteq V \times V' \Big| \\
& \forall_{\substack{n \in N \\ v \in V \\ v' \in V'}} \Big(\big((v, n) \in T \wedge (v', n) \in T' \big) \rightarrow (v, v') \in \phi \Big) \\
\wedge \; & \forall_{\substack{v \in V \\ v', w' \in V'}} \Big(\big((v, v') \in \phi \wedge (v, w') \in \phi \big) \rightarrow (v' = w') \Big) \\
\wedge \; & \forall_{\substack{v, w \in V \\ v' \in V'}} \Big(\big((v, v') \in \phi \wedge (w, v') \in \phi \big) \rightarrow (v = w) \Big) \Big\}
\end{aligned} \tag{4.1}$$

In Equation 4.1, the first condition ensures that Φ is constrained by the mapping induced by the links to the tectogrammatical layer. The remaining two conditions guarantee that Φ is a one-to-one mapping.

Then we can define the annotation agreement s as

$$s_{(G, G', m)} = F_m(G, G', \phi^*) \tag{4.2}$$

where ϕ^* is the optimal mapping between nodes of alternative annotations:

$$\phi^* = \underset{\phi \in \Phi}{\operatorname{argmax}}(F_m(G, G', \phi)) \tag{4.3}$$

and F_m is the F1-measure:

$$F_m(G, G', \phi) = \frac{2 \cdot m(\phi)}{|E| + |E'|} \tag{4.4}$$

where $m(\phi)$ is the number of edges that match given the mapping ϕ. We use four variants of m, which gives us four variants of F and consequently four scores s for every sentence:
Directed unlabeled:

$$m_{du}(\phi) = \Big|\Big\{(v, w, \rho) \in E \Big| \exists_{v',w' \in V', \rho' \in P}\Big((v', w', \rho') \in E' \\ \wedge (v, v') \in \phi \wedge (w, w') \in \phi\Big)\Big\}\Big| \tag{4.5}$$

Undirected unlabeled:

$$m_{uu}(\phi) = \Big|\Big\{(v, w, \rho) \in E \Big| \exists_{v',w' \in V', \rho' \in P}\Big(\\ \big((v', w', \rho') \in E' \vee (w', v', \rho') \in E'\big) \\ \wedge (v, v') \in \phi \wedge (w, w') \in \phi\Big)\Big\}\Big| \tag{4.6}$$

Directed labeled:

$$m_{dl}(\phi) = \Big|\Big\{(v, w, \rho) \in E \Big| \exists_{v',w' \in V'}\Big((v', w', \rho) \in E' \\ \wedge (v, v') \in \phi \wedge (w, w') \in \phi\Big)\Big\}\Big| \tag{4.7}$$

Undirected labeled:

$$m_{ul}(\phi) = \Big|\Big\{(v, w, \rho) \in E \Big| \exists_{v',w' \in V'}\Big(\\ \big((v', w', \rho) \in E' \vee (w', v', \rho) \in E'\big) \\ \wedge (v, v') \in \phi \wedge (w, w') \in \phi\Big)\Big\}\Big| \tag{4.8}$$

These four $m(\phi)$ functions give us four possible F_m measures, which allows us to have four scores for every sentence: s_{du}, s_{uu}, s_{dl} and s_{ul}.

Figure 4.1 shows that the inter-annotator agreement is not significantly correlated with the position of the sentence in the annotation process. This suggests that we excluded enough of the initial sentences where the agreement was gradually increasing until it reached this stable level.

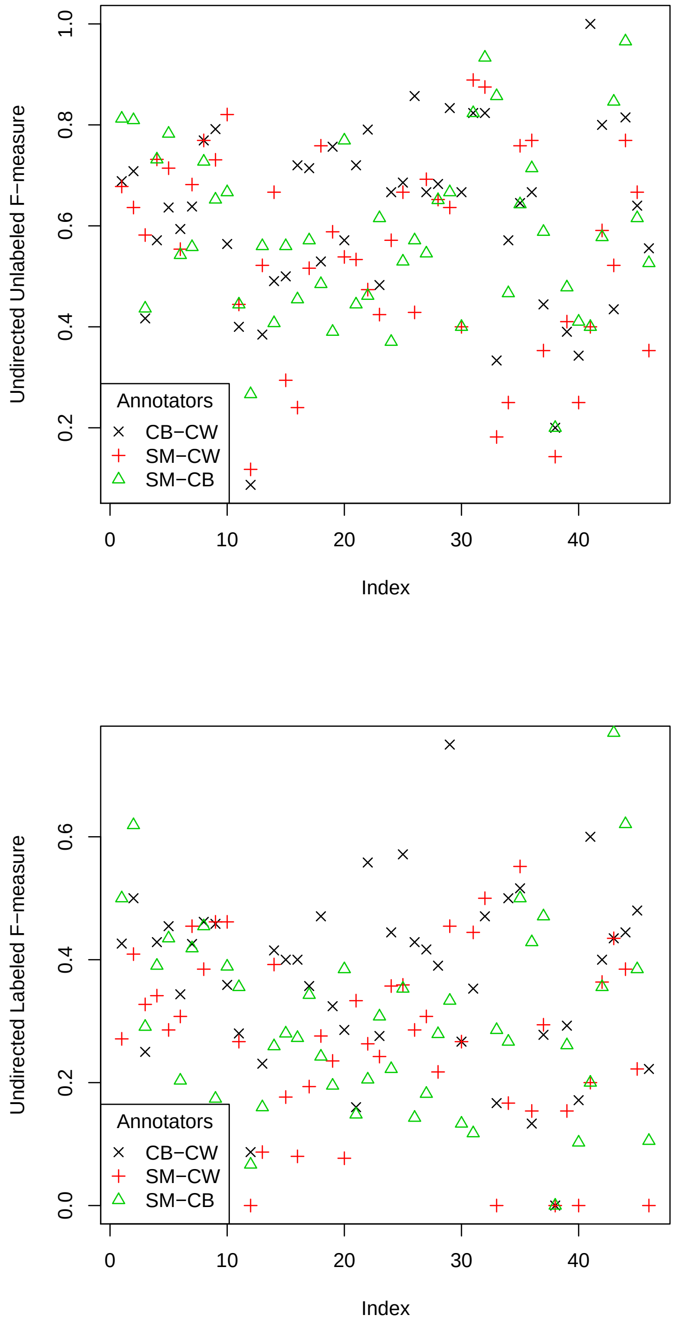


Figure 4.1: Inter-annotator agreement over time. Upper: unlabeled, Lower: labeled. Each point represents a sentence; CB, CW, and SM are the annotators' IDs.

Sample	Annotators	Agreement F-measure			
		s_{uu}	s_{du}	s_{ul}	s_{dl}
Smaller	CB-CW	61.0	56.3	37.1	35.0
Smaller	SM-CB	54.9	48.5	27.1	25.7
Smaller	SM-CW	58.5	50.7	31.3	30.2
Smaller	average	58.1	51.8	31.8	30.3
Larger	CB-CW	64.6	59.8	40.1	38.5

Table 4.1: Inter-annotator agreement in percents. The results come from the two samples described in the Section 4.2.

Figure 4.2 shows that the agreement is not correlated with the sentence length. It means that longer sentences are on average no more difficult than short sentences. The variance decreases with the sentence length as expected.

In Figure 4.3 I show the comparison of directed and labeled evaluations with the undirected unlabeled case. By definition, the undirected unlabeled score is the upper bound for all the other scores. The directed score is well correlated and not very different from the undirected score, indicating that the annotators did not have much trouble with determining the correct direction of the edges. This might be in part due to support from the formalism and the *cedit* tool: each relation type is specified by a sort signature; a relation that violates its signature is reported immediately to the annotator. On the other hand, labeled score is significantly lower than the unlabeled score, which suggests that the annotators have difficulties in assigning the correct relation types (see Section 4.5.3 on page 92 for discussion). The correlation coefficient between s_{uu} and s_{ul} (approx. 0.75) is also much lower than than the correlation coefficient between s_{uu} and s_{du} (approx. 0.95).

Figure 4.4 compares individual annotator pairs. The scores are similar to each other and also have a similar distribution shape.

A more detailed comparison of individual annotator pairs is depicted in Figure 4.5. The graph shows that there is a significant positive correlation between scores, i.e., if two annotators can agree on the annotation, the third annotator is likely to also agree, but this correlation is not a very strong one. The actual correlation coefficients are shown under the main diagonal of the matrix.

All the results are summarized in Table 4.1.

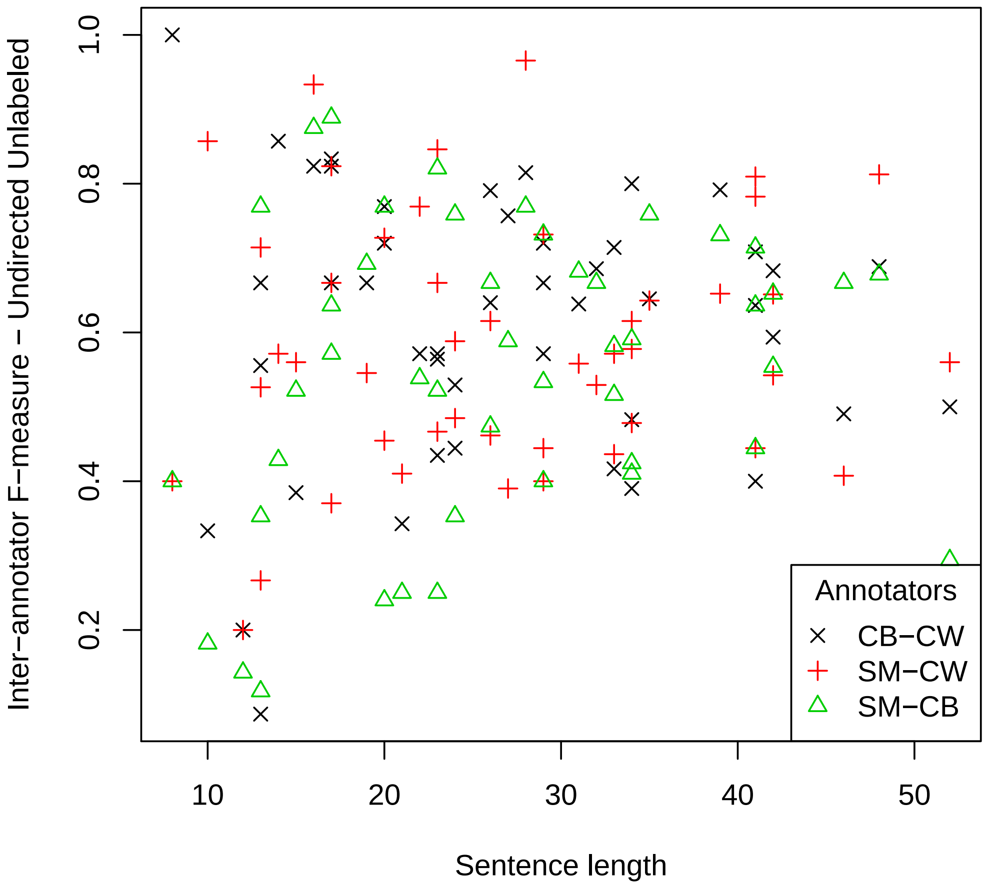
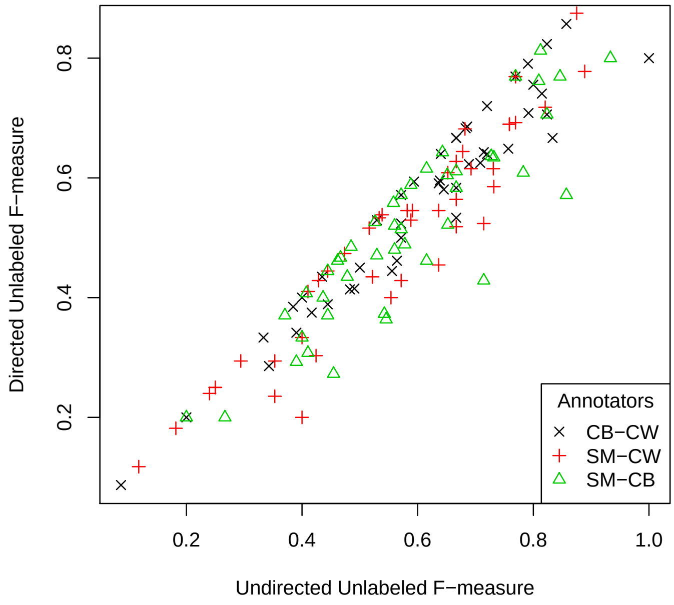


Figure 4.2: Inter-annotator agreement depending on the sentence length.



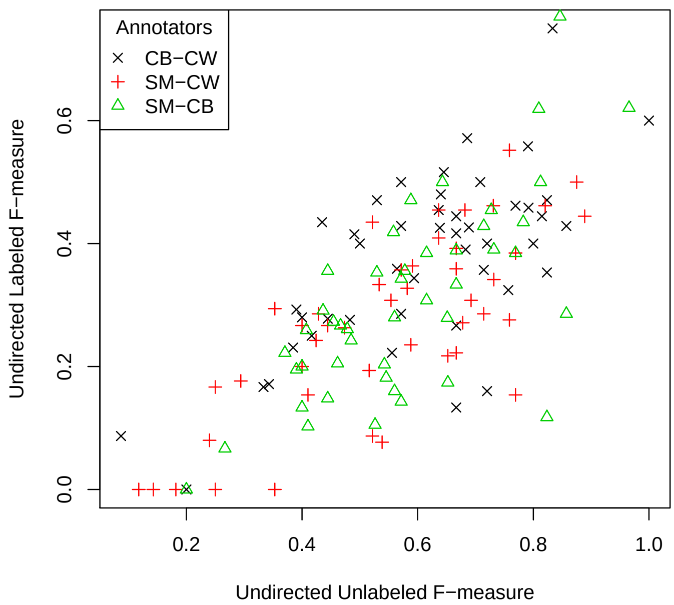


Figure 4.3: Upper: Directed vs. undirected inter-annotator agreement. Lower: Labeled vs. unlabeled inter-annotator agreement.

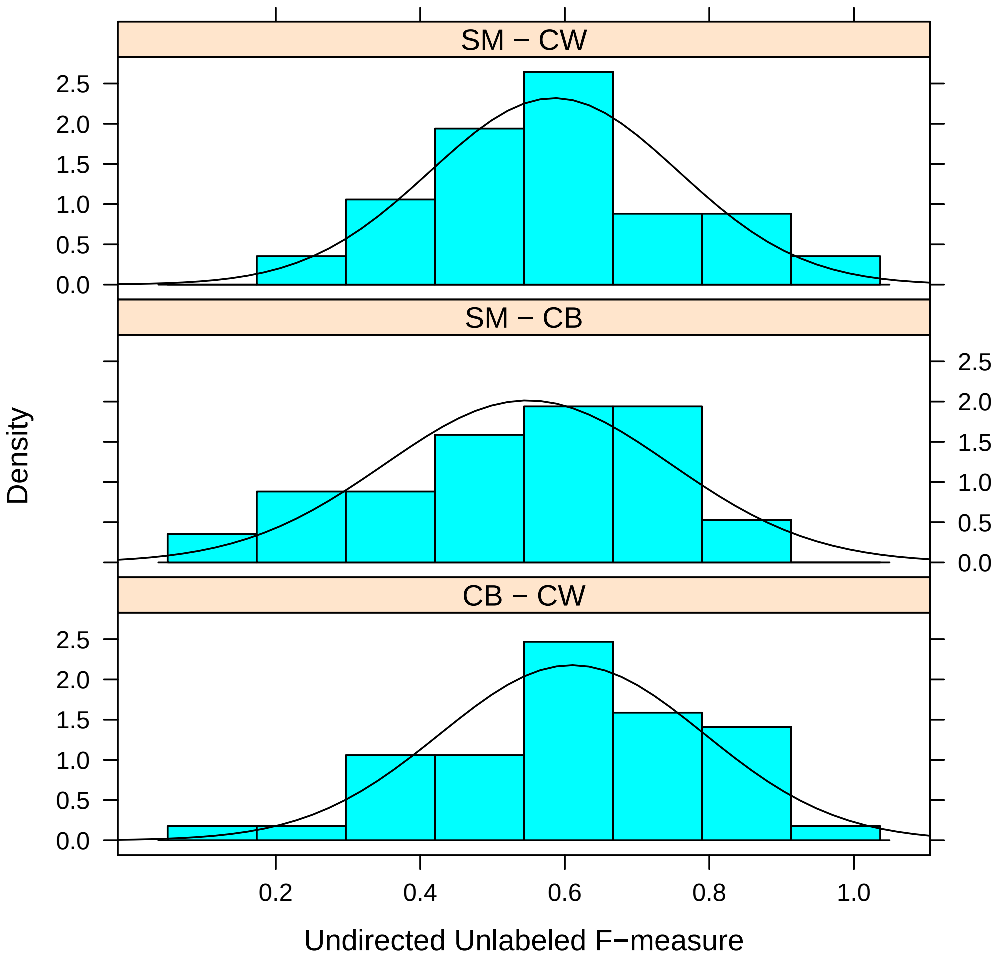


Figure 4.4: Comparison of individual annotator pairs.

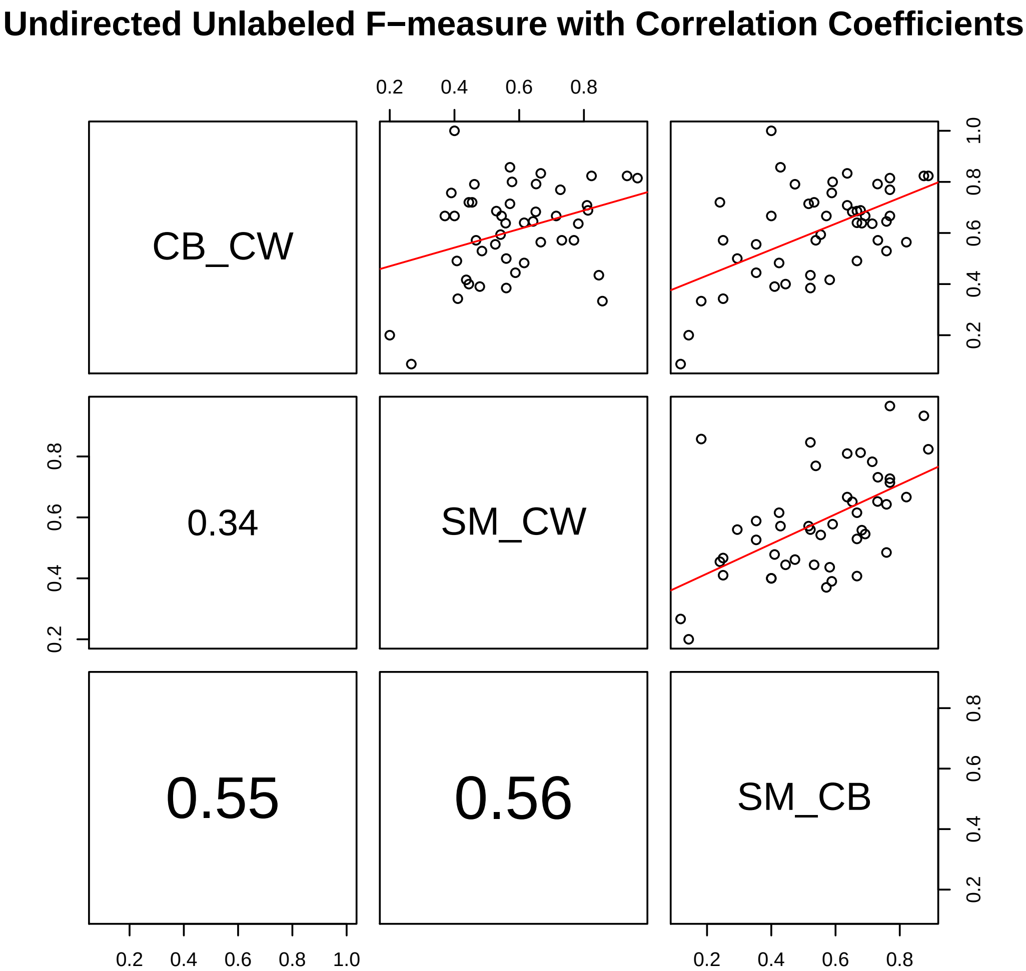


Figure 4.5: Correlation of individual annotator pairs agreements.

4.4 Inter-sentential Coreferences

The coreference annotation is an inherent part of the PDT (Mikulová *et al.*, 2006; Hajičová *et al.*, 2000b). However, in our annotation we do not use the coreference information from the tectogrammatical layer to be able to look for the differences in these two approaches to coreference.

To evaluate the agreement in coreferences and attribute values, we need a suitable measure F_m as defined in Equation 4.4 on page 77 to obtain the best possible mapping ϕ^*. Ideally, we should combine the measures presented in Equation 4.5 through Equation 4.8, because none of these measures gives us the best possible mapping: The m_{uu} measure would not prefer mappings where the labels and the edge directions match. On the other hand, the m_{dl} measure would not map nodes where there is a mismatch in the label or in the direction. To overcome this, we can use a combined measure $m_a(\phi)$, defined in Equation 4.9.

$$m_a(\phi) = |M_{dl}| + \frac{3}{4} \cdot |M_{wl}| + \frac{1}{2} \cdot |M_{dw}| + \frac{1}{4} \cdot |M_{ww}| \tag{4.9}$$

where $|M_{dl}|$ is the number of edges where both direction and the label matches:

$$\begin{aligned} M_{dl} = \Big\{(v, w, \rho) \in E \Big| \exists_{v',w' \in V'} :& \\ & (v', w', \rho) \in E' \wedge (v, v') \in \phi \wedge (w, w') \in \phi\Big\} \end{aligned} \tag{4.10}$$

$|M_{wl}|$ is the number of edges, where the direction is wrong but the label matches:

$$\begin{aligned} M_{wl} = \Big\{(v, w, \rho) \in E \Big| \exists_{v',w' \in V'} :& (v, w, \rho) \notin M_{dl} \\ & \wedge (w', v', \rho) \in E' \wedge (v, v') \in \phi \wedge (w, w') \in \phi\Big\} \end{aligned} \tag{4.11}$$

$|M_{dw}|$ is the number of edges, where the direction is the same but the labels differ:

$$\begin{aligned} M_{dw} = \Big\{(v, w, \rho) \in E \Big| \exists_{v',w' \in V', \rho' \in P} :& (v, w, \rho) \notin (M_{dl} \cup M_{wl}) \\ & \wedge (v', w', \rho') \in E' \wedge (v, v') \in \phi \wedge (w, w') \in \phi\Big\} \end{aligned} \tag{4.12}$$

Finally, $|M_{ww}|$ is the number of edges, where both the direction and the label differ:

$$M_{ww} = \Big\{(v, w, \rho) \in E \Big| \exists_{v', w' \in V', \rho' \in P} : (v, w, \rho) \notin (M_{dl} \cup M_{wl} \cup M_{dw}) \\ \wedge (w', v', \rho') \in E' \wedge (v, v') \in \phi \wedge (w, w') \in \phi \Big\} \tag{4.13}$$

The coefficients in Equation 4.9 were chosen arbitrarily to prefer mappings of the edges with more matching parameters and at the same time mappings where there are at least some structural correspondences. The relation type received more weight than the edge direction, because the agreement in the edge label is more informative than the agreement in the edge direction.

In the rest of this chapter, all the presented results are obtained using the optimal mapping ϕ^* found using m_a defined in Equation 4.9 as m in Equation 4.4 for each sentence. The coreference evaluation algorithm looks at every concept which occurs in more than one sentence. The first occurrence is identified and the concept is mapped to the other annotator's corresponding concept. In every subsequent occurrence the mapping is used to find the other annotator's concept. If the concept is the same as in the case of the first occurrence, we count this occurrence as a matching coreference. The complete algorithm is presented in Figure 4.6.

The results are summarized in Table 4.2 and Figure 4.7. There are four categories of coreferences: "mismatch": here the other annotator uses a different concept as the coreference target. "missingR0": the first mention of the concept has no counterpart in the best matching alternative annotation. "noMap": the concept which is coreferring to a previous sentence has no mapping to the alternative annotation. "ok": The coreferring concept used in the sentence is mapped to a concept in the alternative annotation which corefers with the concept mapped to the referent of this concept.

The error analysis shows that most of the errors are due to either insufficient guidelines, or an error in the annotation structure which leads to erroneous mapping of the alternative annotations.

An example of insufficient guidelines is the article F20 for the annotator pair CB-CW. One of the early sentences contains an apposition "*The U.S. trade representative, Carla Hills, . . .*" Carla Hills is subsequently referred to in almost every sentence, but while one annotator uses *Carla Hills* from the first sentence as the coreference target, the other uses the expression *The U.S. trade representative.*

Input: Alternative annotations $G = (V, E, T, A)$ and $G' = (V', E', T', A')$, mapping ϕ^*_s for every sentence s
Output: List of coreference agreements and disagreements

```
foreach v ∈ V subject to |{n ∈ N; (v, n) ∈ T}| > 1 do
    Find the first sentence s_0 containing the concept v;
    Find the v'_0 ∈ V' such that (v, v'_0) ∈ φ*_{s_0};
    if there is no such v'_0 then
        print("missingR0 for " + v);
    else
        foreach n ∈ N where (v, n) ∈ T except for the n in s_0 do
            Find the sentence s containing the node n;
            Find the v' ∈ V' such that (v, v') ∈ φ*_s;
            if there is no such v' then
                print("noMap for " + v + " at " + n);
            else
                if v' = v'_0 then
                    print("ok for " + v + " at " + n);
                else
                    print("mismatch of " + v' + " and " + v'_0);
```

Figure 4.6: Comparing of two alternative coreference annotations. The asymptotic algorithmic complexity is $\mathcal{O}(|V| + |T|)$ because every inner loop iterates over different sets of n.

4.5 Labeling Agreement

4.5.1 Layer Attributes

Every MultiNet concept has several attributes specifying its nature. The key attribute is the SORT of the concept (e.g., action, concrete object, unit of measurement, . . .). The annotator fills in the other attributes according to the SORT: not all sorts of concepts have all the attributes. As described in Helbig (2006), sorts are divided into three categories, where the first category has all attributes, the second category has only FACT and GENER, and the last category has no other attributes except for the SORT itself.

FACT

FACT attribute has three values: real, hypothetical and non-real. Non-real case is used in the case of a double negation, which was not observed in the data (e.g., *It's not true that I won't go there*). From the 1828 filled FACT

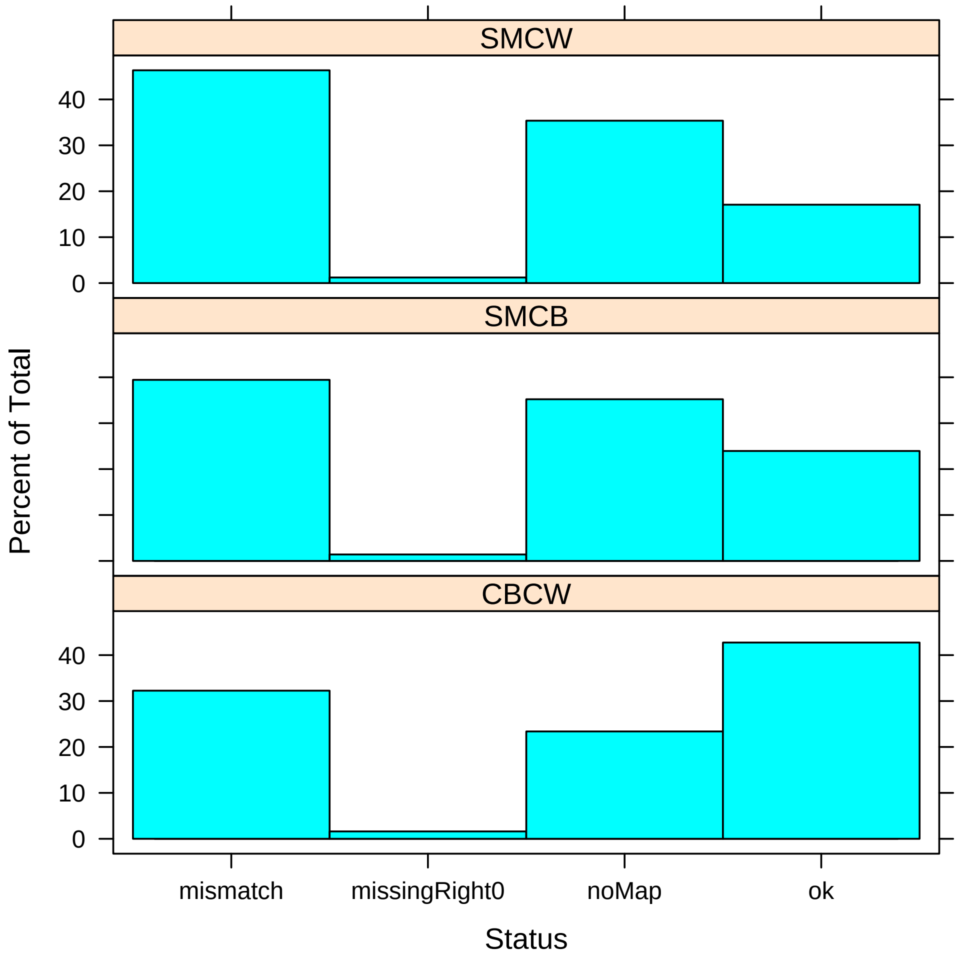


Figure 4.7: The agreement for all pairs of annotators, CBCW, SMCB, and SMCW. The data shows high variance w.r.t. the distribution of "ok" cases.

Article	Status Pair	mismatch	missingR0	noMap	ok
F20	CB-CW	17	1	5	14
	SM-CB	15	0	24	11
	SM-CW	23	1	22	4
F21	CB-CW	3	0	6	8
	SM-CB	6	0	1	3
	SM-CW	5	0	2	3
F22	CB-CW	9	1	7	8
	SM-CB	7	1	0	3
	SM-CW	10	0	5	7
F26	CB-CW	6	0	5	7
F27	CB-CW	5	0	6	16

Table 4.2: Experimental results of coreference annotation agreement evaluation.

attributes, 96.2% were real, the remaining 3.8% being hypothetical. There were 91 disagreements.

An example of real/hypothetical disagreement is the following sentence:

"He said Mexico could be one of the next countries to be removed from the priority list because of its efforts to craft a new patent law."

The disagreement was about the action "*remove*", i.e., one annotator understood that there will be some removing, while another one considered this action as merely hypothetical. See Section 3.3 on page 66 for further discussion.

GENER

GENER attribute has two values: generic and specific. Out of 1842 attributes filled in, 1426 (77.4%) were specific, the remaining 22.6% generic. There were 465 mismatches. This disagreement comes mainly from overusing the generic value of the attribute caused by the vague instructions regarding the attribute (see Section 3.3 on page 70 for further discussion).

In the following sentence, the expressions with a mismatching degree of genericity are boldfaced:

*"Gary Hoffman, a Washington lawyer **specializing** in intellectual-property **cases**, said the threat of U.S. **retaliation**, combined with a growing **recognition** that protecting **intellectual property** is in a country's own interest, prompted the improvements made by South Korea, Taiwan and Saudi Arabia."*

Frequency	**SORT**	Description
416	io + d	io and d combination
386	d	discrete object
369	da	action
198	io	ideal object
89	tq	total quality
85	ad + io	ad (dynamic abstraction) and io combination
85	t	temporal situation descriptor
76	p	property (general)
75	nq	nonmeasurable quality
67	nu	numeric quantificator
62	st	static situation (state)

Table 4.3: Most frequent sorts in the sample data. + denotes a combination of several simple sorts.

SORT

SORT attribute has a whole hierarchy of values. 48 of them were actually used in the data. Table 4.3 shows the most frequent sorts in the sample.

The total number of filled sorts is 2428 and the estimated entropy is 4.15. 1287 disagreements were found. The evaluation of SORT values is complicated by two facts:

1. They form a hierarchy, i.e., some of the values are closer to each other than others. Sometimes the disagreement can be caused by the fact that one annotator simply selected a subsort of the other annotator's sort.

2. There are several "meaning molecules" – collections of sorts (e.g., "*certificate*" is both an ideal object and a discrete object). The situation, where the first annotator selects the molecule and the second annotator only its part is also considered a mismatch.

The most common mismatched pairs are presented in Table 4.4.

A closer look at the values reveals that the most confusing pairs of sorts are usually the ones quite close to each other in the hierarchy. In Table 4.4 it is the case for all pairs except for local situation descriptor vs. ideal object + discrete object: this happens for geographical units, where some contexts tend to introduce them as institutions (e.g., "*Israel refused to . . .* "), while other contexts suggest a location (e.g., "*It happened in Israel.*"). In this case there is a tension between the annotators' effort to assign a consistent

Count	First Annotator	Second Annotator
209	d (discrete object)	io (ideal object) + d
89	io	io + d
55	d	io
33	da (action)	st (static situation (state))
28	io	ad (dynamic abstraction) + io
26	nq (nonmeasurable quality)	p (property (general))
23	nq	tq (total quality)
22	ad + io	io + d + ad
21	aq (associative quality)	tq (total quality)
21	io + d	io + d + ad
18	p	tq
18	t (temporal situation descriptor)	ta (temporal abstractum)
15	l (local situation descriptor)	io + d
15	ad + io	io + d
13	da	ad + io

Table 4.4: Most often mismatching pairs of SORT values. + denotes a combination of several simple sorts (a *meaning molecule*).

sort to the same kind of concepts across all articles and the contexts which strongly prefer one reading over another.

REFER

REFER attribute has two possible values: determinate and indeterminate. In the sample there are 1368 nodes with a filled REFER attribute, 1018 (74.4%) of them determinate. A mismatch was found in 384 cases. The mismatches appear in contexts where there is no explicit article present, i.e. "*U.S. claims some success in its trade* ***diplomacy****.*" In this example, the the REFER of "*diplomacy*" differs.

VARIA

VARIA attribute has two possible values: variable and constant. Variable concept is a concept, which is seemingly referring to a single object, but in fact the object is not a single object, but a whole collection (e.g., *book* in "*Every student will get a book*"). There were 1370 values filled in the sample. 1031 (75.3%) of them were constant. 403 values were in a mismatch. From this summary it is apparent that the definition of what is constant is not very clear. The annotators tend to mark some concepts as variable because they vary over time (e.g., "*They removed it from the **list** of countries*").

4.5.2 Functions

There are 14 different functions used in the data set. The most frequently used functions are ITMS (a function creating a collection out of individual concepts), PMOD (modification of a concept by an associative property (e.g., "*trade diplomacy*")), and QUANT (function creating quantity from a number and a measurement unit). There are 270 occurrences of a function and 62 mismatches. The sample entropy of FTYPE distribution is 2.86. There were only three pairs in the confusion matrix with more than five occurrences:

- ITMS vs. VEL1: 15 times. VEL1 is the "or" function while ITMS can be considered as the "and" function. All the mismatches were caused by only one annotator, who strongly preferred the use of VEL1 over ITMS, e.g. in "*the threat of U.S. retaliation, combined with a growing recognition of . . . , prompted the improvements . . .* "
- MODQ vs. QUANT: 14 times. MODQ generates a modified quantity by applying a graduator to the original quantity. This confusion is caused by overusing MODQ function where QUANT would be appropriate.
- ITMS vs. PMOD: 9 times. This confusion emerges in situations where there are complicated structures with several functions and the annotators get confused, e.g. in "*all trading nations, and particularly the creativity and inventiveness of an offending country's own citizens*", where there are nested ITMS functions combined with PMOD.

4.5.3 Relations

All concepts in the semantic network are connected to the rest of the network with edges representing their relations. The edges are labeled with a relation type (RTYPE). There are 60 different RTYPEs in the sample with the sample entropy of 4.49. RTYPEs are the most important labels in the network because they mark the roles of all concepts in the overall meaning. The most commonly used RTYPEs are summarized in Table 4.5.

The sample contains 608 RTYPE mismatches. The most frequent ones are presented in Table 4.6.

Frequency	**RTYPE**	Description
263	AGT	Agent
254	EQU	Equality
208	PROP	Property
208	ANTE	Temporal precedence
198	OBJ	Object of a dynamic situation
179	ATTCH	Attachment (e.g., Paul's sister)
177	MCONT	Mental content of a situation
127	ASSOC	Loose association
72	QUANT	Quantity
57	TEMP	Temporal embedding
42	DUR	Duration
37	ORNT	Orientation of a situation
29	RSLT	Result
29	AFF	Affected object
28	PURP	Purpose
27	BENF	Beneficiary
26	SCAR	State carrier
26	MODL	Modality
24	OPPOS	Opposing situation
22	LOC	Location

Table 4.5: Most frequent RTYPEs in the sample. The total number of all RTYPE instances is 2328.

Frequency	RTYPEs	Description
40	ASSOC vs. PROP	Association vs. Property
32	MCONT vs. OBJ	Mental content vs. Object
20	ATTCH vs. PROP	Attachment vs. Property
17	DUR vs. TEMP	Duration vs. Temporal embedding
16	AFF vs. OBJ	Affected object vs. Object
16	ATTCH vs. EQU	Attachment vs. Equality
15	ATTCH vs. MCONT	Attachment vs. Mental content
14	ASSOC vs. ATTCH	Association vs. Attachment
12	ASSOC vs. MCONT	Association vs. Mental content
10	ASSOC vs. QUANT	Association vs. Quantity
9	AGT vs. EXP	Agent vs. Experiencer
9	AGT vs. MEXP	Agent vs. Mental experiencer
9	OBJ vs. ORNT	Object vs. Orientation
8	AGT vs. SCAR	Agent vs. State carrier
8	ATTCH vs. POSS	Attachment vs. Possession

Table 4.6: Most frequently confused pairs of relations. There were 608 disagreements in total.

Table 4.7 shows the RTYPEs which appear most frequently in a mismatch. The table shows that the most confusing relations are MCONT, PROP, OBJ, ATTCH, and AGT. ASSOC relation is used in situations where the annotators do not know which relation to use. It is a kind of backing off relation. In the rest of this section we will discuss possible causes for the disagreements about these relations. We will ignore the ASSOC relation which itself is a sign of a confusion.

MCONT

Mental content is confused above all with OBJ and ATTCH. The confusion with OBJ is caused by the similarity of their syntactical distributions. E.g., “*to issue a review*”: “*review*” can be considered both as an OBJ and as the MCONT of “*to issue*”. Similarly, in the case of “*movie producer*” it is tempting to call the relation MCONT, marking the movie as the mental content. In the latter example, however, the MCONT edge should start at the concept representing the action of production, not at the producer, who is the AGT of the action.

Frequency	RTYPE	Description
116	MCONT	Mental content
112	ASSOC	Association
105	PROP	Property
103	OBJ	Object
99	ATTCH	Attachment
77	AGT	Agent
39	EQU	Equality
39	TEMP	Temporal embedding
30	ORNT	Orientation
29	AFF	Affected object
29	ANTE	Temporal precedence
27	DUR	Duration
26	CTXT	Contextual embedding
26	SCAR	State carrier
25	QUANT	Quantity
24	RSLT	Result
22	PURP	Purpose
21	EXP	Experiencer
18	BENF	Beneficiary
18	MEXP	Mental experiencer

Table 4.7: Most frequently confused individual RTYPEs. There were 608 mismatches, i.e. total of 1216 RTYPEs in a mismatch.

PROP

PROP is most frequently confused with ATTCH. One example is the construction "*their own authors*". It is not quite clear what the semantics of "*own*" is in this case. One annotator marked "*own*" as the property of the authors. Another example is "*creditor banks*", where the actual relation between the "*creditor*" and the "*bank*" is rather complex.

OBJ

OBJ is typically confused with MCONT (discussed in Section 4.5.3) and AFF. The difference between AFF and OBJ should lie in the fact that AFF changes the object upon which it is acted, while OBJ should not be physically changed. On page 446 of Helbig (2006) it is noted that this distinction has problematic border cases and these are indeed found in our sample. Examples of these border cases are "*to discourage their scientists*", "*to initial* *the agreement*" and "*to improve its standing*".

ATTCH

ATTCH is often confused with PROP (discussed in Section 4.5.3), EQU and MCONT (discussed in Section 4.5.3). There is no reason to confuse ATTCH with EQU, because these two relations are clearly distinguishable. The EQU relation is used for coreference annotation and ATTCH for conceptual attachment. A closer look shows that the mismatches are caused by incorrect mappings caused by mismatches in the EQU neighborhood. EQU, as the coreference relation, is often used in connection with concepts attached to a different sentence than the current one. This fact makes the concepts subject to arbitrary mapping in the situations where there is no clear correspondence. These situations arise especially when the coreferring node is a node not connected to the tectogrammatical tree (e.g., a PMOD function node).

AGT

AGT is confused quite randomly with many relations. There are 77 mismatches, but the three most frequently mismatching relations, EXP, MEXP, and SCAR, occur only 9, 9, and 8 times, respectively. The reason for confusion between these relations is apparent: they all connect a situation (typically expressed by a verb) to the concept expressed by the subject of the sentence. We have already shown in Table 4.4 that the the mismatch of action and static state belongs to the top ranking ones. This fact explains the mismatches between AGT and SCAR, the former being used for actions, while the latter for states.

The difference between AGT on one side and EXP and MEXP on the other side is a more subtle one. AGT should be an active participant in the situation. Border cases found in the sample include both AGT-EXP mismatches ("***they** failed to **honor** the copyrights*") and AGT-MEXP mismatches ("***she** hasn't **deemed** any cases bad enough to. . .* ").

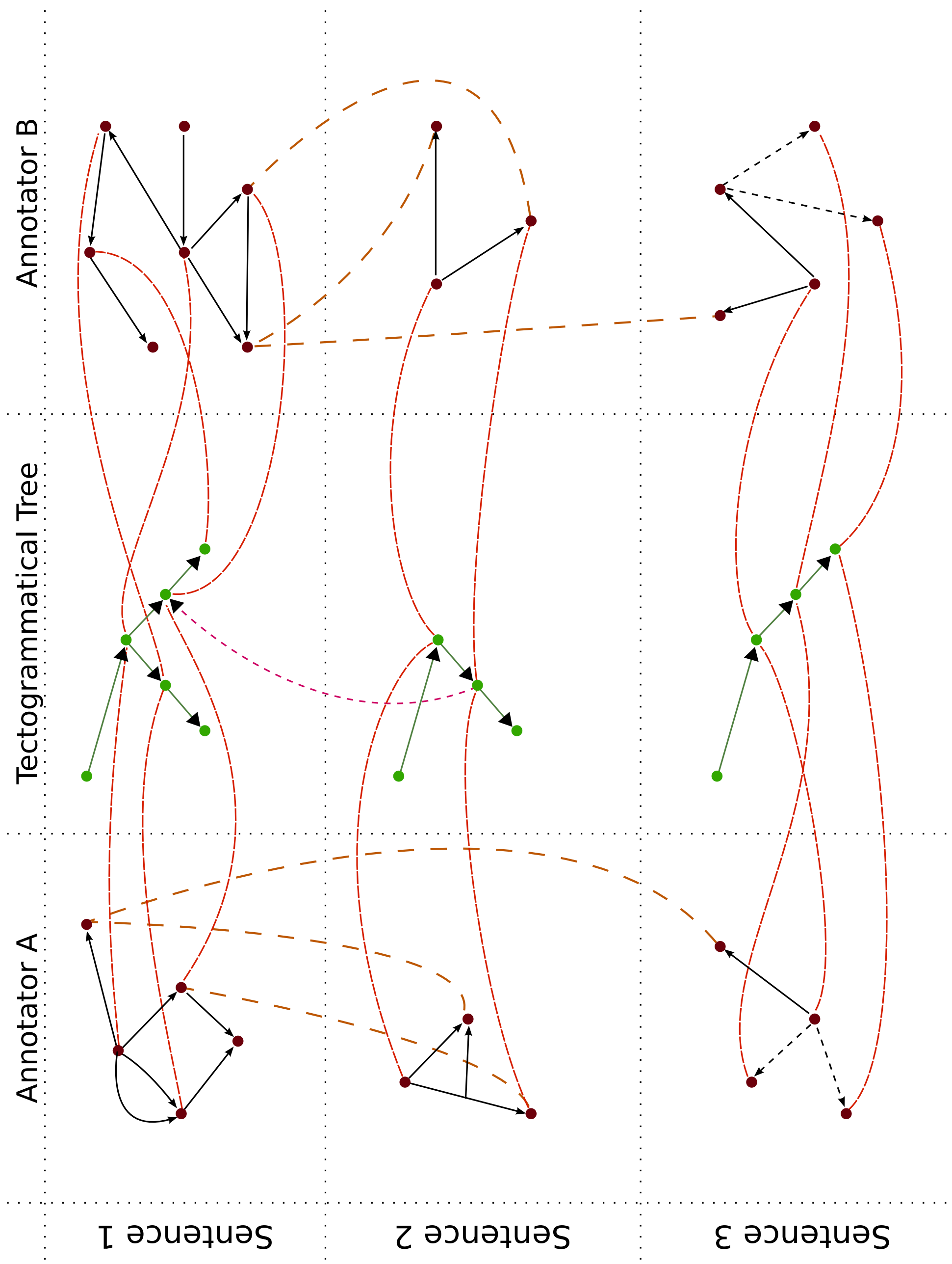


Figure 4.8: The comparison of alternative annotations. Dashed undirected lines in MultiNet connect various occurrences of the same concept. The connections to the TR are created by the annotator, but not all MultiNet concepts are anchored to TR.

Chapter 5

Conclusion

> But if you say: "How am I to know what he means, when I see nothing but the signs he gives?" then I say: "How is *he* to know what he means, when he has nothing but the signs either?"
>
> Wittgenstein (1953, I/§504)

The evaluation of the inter-annotator agreement shows that the annotation process in this form is too complex to be practically used as a process leading to a usable knowledge base. There are three independent possible improvements:

1. Automatic preprocessing. The annotation from scratch increases the frustration of the annotators. The reduced amount of work per sentence would significantly help the speed. The annotation would become more consistent because in cases where the annotators are not sure, they would leave the structure produced by the preprocessing. On the other hand this approach can easily lead to artificially high similarity between the Tectogrammatical Representation and the semantic network.

2. Division of labor. If the annotation task would be split into several steps and every annotator would be assigned to a more specialized task, it would reduce the required amount of training and speed up the annotation. However, the main burden associated with the annotation is to develop the structure of the network, relation labels and concept sorts. These three tasks are rather entangled and not easily separated.

3. High quality lexicon. The HaGenLex lexicon (see Section 2.6.3) is currently sparse and not very useful except for simple structures. The instant access to a relevant example during the annotation would certainly be useful.

In the end, two important questions remain to be answered:

1. What new information do we gain by having the semantic network annotation?

2. Can we add this information even in the framework of Tectogrammatical representation?

The following list is trying to address these questions.

- Temporal ordering of situations and states: The TR would have to allow for a new kind of link similar to the ANTE relation in MultiNet. The temporal relationships among various states and actions in the discourse (Prior, 1967; Reichenbach, 1947) require proper formal means (Moens and Steedman, 1988; Němec, 2006; Panevová *et al.*, 1971).

- Specific vs. generic distinction (see Section 2.2.1 on page 29): The TR would have to allow for a new attribute distinguishing generic from specific concepts.

- Bridging anaphora: This can be achieved only by allowing TR annotators to add the new sentences necessary to explicate the type of bridging.[1]

- Multi-word expressions: This is already being solved for TR in Bejček *et al.* (2006).

- Distinguishing locations from location-like expressions which share some semantical properties of locations, but are not describing a physical location (Jackendoff, 1990). An example of a location-like expression is "He belongs to our greatest specialists.": This would require a completely new treatment for DIR1, DIR2, DIR3, and LOC functors of TR.

[1] E.g., "The house was small and the roof was red". In this case we have to add a new sentence "The house has a roof" to be able to represent the coreferences. In MultiNet, we only add one PARS ("part of") relation connecting the roof to the house.

- Encapsulation of concepts: This MultiNet feature is yet to be analyzed. The definitions regarding encapsulation were so vague that we decided not to annotate the knowledge types at the moment.

As a further step, the rules of inference should be applied to the annotated network to eliminate the disagreement caused by the fact that one annotator explicitly annotated something that was omitted by the other annotator because he or she was aware of the inference rule. The simplest inferences are regarding the symmetry and transitivity of some relations (EQU in Figure 5.1 being an example of symmetric relation of equivalence and ANTE in Figure 5.2 being an example of transitive relation (temporal succession)). A very preliminary list of inference rules is presented in Helbig (2006).

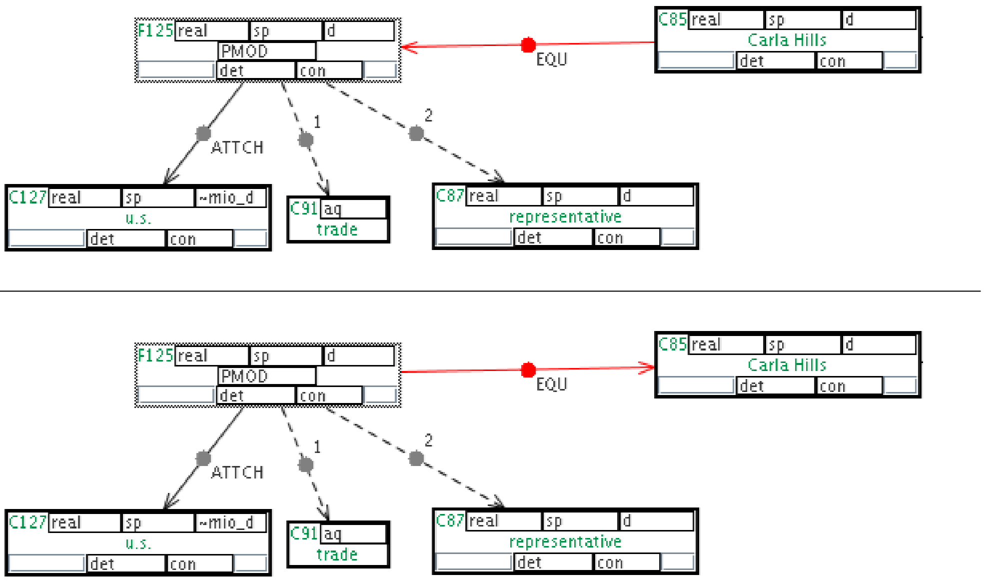


Figure 5.1: Example of a symmetrical relation. The network represents the expression “U.S. trade representative Carla Hills”. Both annotations are semantically equivalent but the graphs differ in the highlighted relation.

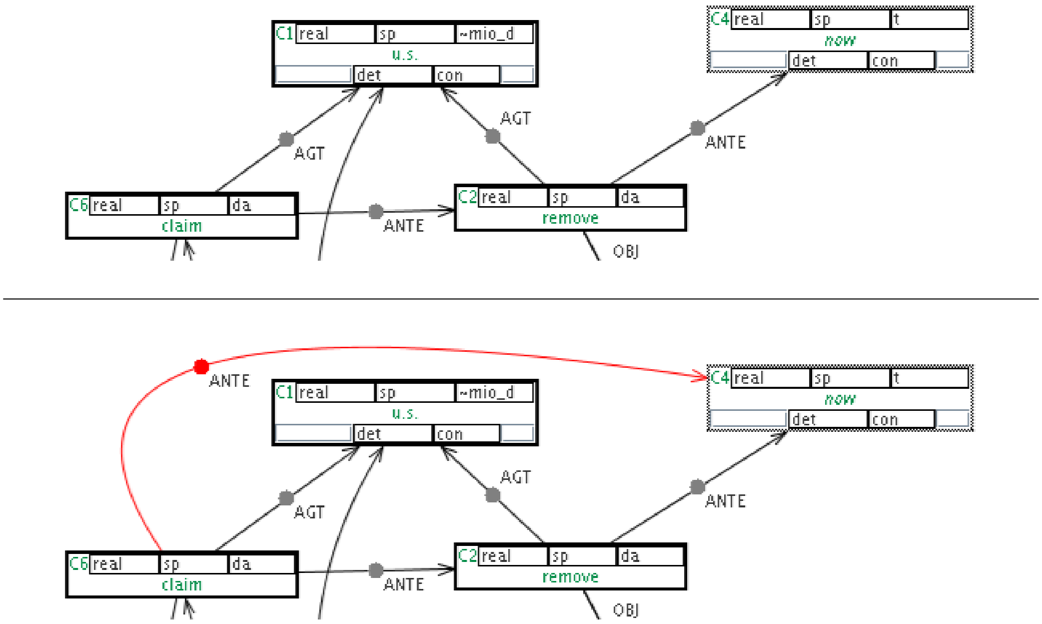


Figure 5.2: Example of a transitive relation. The network represents a part of the sentence “The U.S., claiming some success in its trade diplomacy, removed . . . ”. Both annotations are semantically equivalent but the graphs differ in the highlighted relation.

Appendix A

Example Annotations

The following three sentences are originally consecutive in the text. Note that some concepts appear in more than one sentence (they can be identified by the same ID).

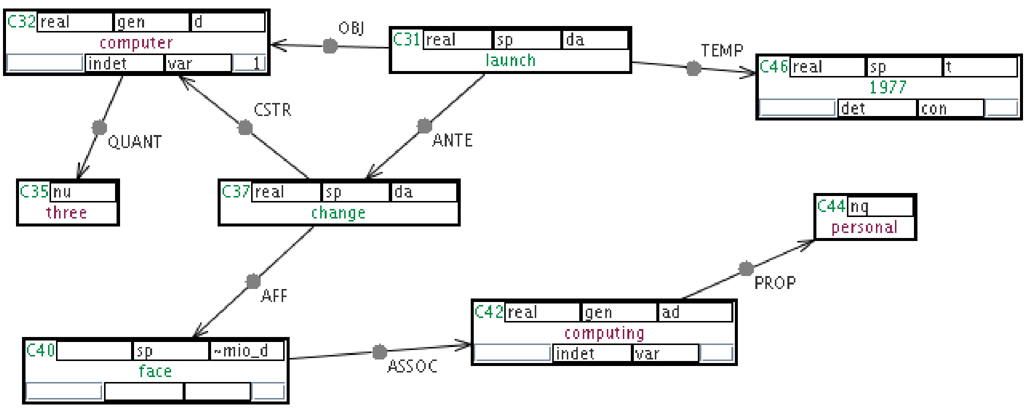


Figure A.1: Sentence 1: "Three computers that changed the face of personal computing were launched in 1977."

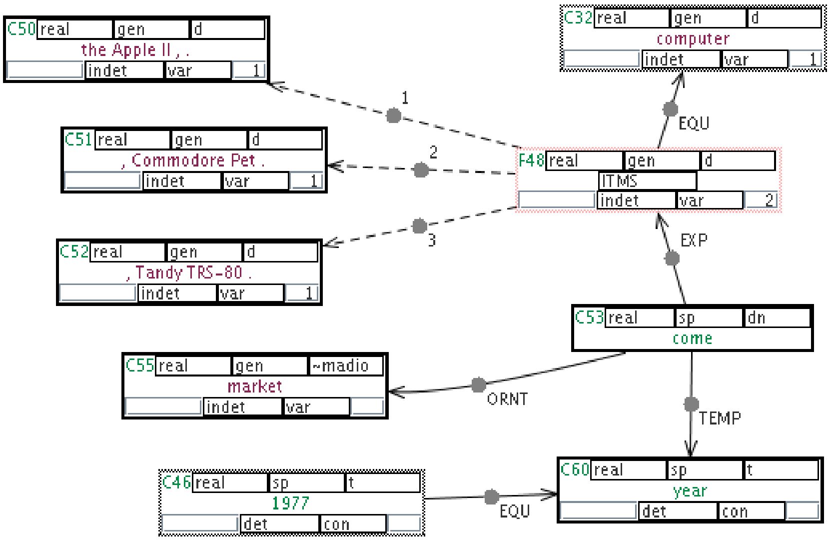


Figure A.2: Sentence 2: "That year the Apple II, Commodore Pet and Tandy TRS-80 came to market."

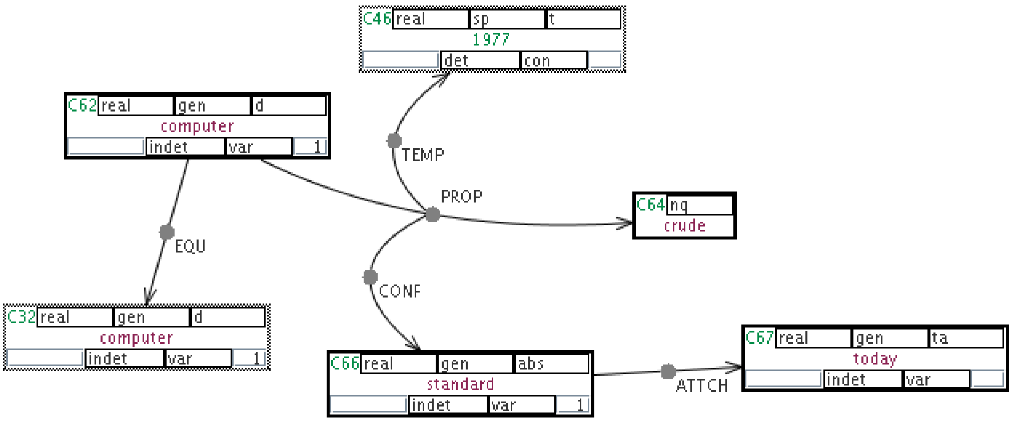


Figure A.3: Sentence 3: "The computers were crude by today's standards."

Appendix B

Overview of Sorts

This appendix gives an overview of the hierarchy of sorts used in the annotation. A more detailed description can be found in Helbig (2006) and on the project wiki page[1].

ent entity

- **o** object
  - **co** concrete object
  - **d** discrete object
  - **s** substance
  - **ab** abstract object
    - **at** attribute
    - **oa** operational attribute
    - **na** nonoperational attribute
  - **re** relation
  - **io** ideal object
  - **ta** temporal abstractum
  - **mo** modality
    - **abs** abstraction from situation
    - **ad** dynamic abstraction
    - **as** static abstraction
- **si** situation

[1] https://wiki.ufal.ms.mff.cuni.cz/projects:content-annotation:layer-attributes:sort:hierarchy-of-sorts

- **st** static situation (state)
  - **dy** dynamic situation (event)
  - **da** action
  - **dn** happening

- **sd** situation descriptor
- **t** temporal situation descriptor
- **md** modal situation descriptor
- **l** local situation descriptor

- **ql** quality
  - **p** property (general)
    - **gq** gradable quality
    - **mq** mesurable quality
    - **nq** nonmesurable quality
  - **tq** total quality
- **rq** relational quality
  - **fq** functional quality
  - **aq** associative quality
  - **oq** operational quality

- **gr** graduator
- **lg** qualitative graduator
- **ng** quantitative graduator

- **qn** quantity
- **m** measurement
- **me** unit of measurement
  - **qf** quantificator
  - **nn** nonnumeric quantificator
  - **nu** numeric quantificator

fe formal entity

Meaning molecules with examples:

˜mio_d io, d: certificate, orchestra, school, province, cube

˜**madio** ad,io: globalization, speech

˜**mio_dad** io, d,ad: announcement

˜**miooa** io,oa: temperature

˜**masio** as,io: anarchy

˜**masad** as,ad: pregnancy

List of Figures

List of Tables

Bibliography

Carlos Areces and Patrick Blackburn. Bringing them all Together. *Journal of Logic and Computation*, 11(5):657–669, 2001. 23

Carlos Areces, Patrick Blackburn, and Maarten Marx. Hybrid Logics: Characterization, Interpolation and Complexity. To appear in the Journal of Symbolic Logic, http://www.loria.fr/projets/hylo/Papers/jsl.pdf, 2004. 23

Ron Artstein and Massimo Poesio. Inter-coder agreement for computational linguistics. *Computational Linguistics*, submitted, 2007. 76

Mike Badger. Dokuwiki – A Practical Open Source Knowledge Base Solution. *Enterprise Open Source Magazine*, 2007. 66

Jon Barwise and Robin Cooper. Generalized Quantifiers and Natural Language. In J. Kulas, J. H. Fetzer, and T. L. Rankin, editors, *Philosophy, Language, and Artificial Intelligence: Resources for Processing Natural Language*, pages 241–303. Kluwer, Boston, 1988. 29

Eduard Bejček, Petra Möllerová, and Pavel Straňák. The Lexico-Semantic Annotation of PDT. In *Lecture Notes in Computer Science, Proceedings of the 9th International Conference, TSD 2006*, number 4188 in Lecture Notes In Computer Science, pages 21–28. Springer-Verlag Berlin Heidelberg, 2006. 64, 100

Patrick Blackburn. Representation, Reasoning, and Relational Structures: A Hybrid Logic Manifesto. *Logic Journal of the IGPL*, 8(3):339–625, 2000. 23

Patrick Blackburn. Modal Logic As Dialogical Logic. *Synthese*, 127(1 - 2):57–93, April 2001. 23

Igor Bolshakov and Alexander Gelbukh. The Meaning-Text Model: Thirty Years After. *International Forum on Information and Documentation*, 1:10–16, 2000. 25

Johan Bos. Towards Wide-Coverage Semantic Interpretation. In *Proceedings of Sixth International Workshop on Computational Semantics IWCS-6*, pages 42–53, 2005. 24

Ronald Brachman and James Schmolze. An Overview of the KL-ONE Knowledge Representation System. *Cognitive Science*, 9:171–216, 1985. 26, 37

Ronald J. Brachman, Deborah L. McGuinness, Peter F. Patel-Schneider, Lori Alperin Resnick, and Alex Borgida. Living with CLASSIC: When and How to Use a KL-ONE-like Language. In John Sowa, editor, *Principles of Semantic Networks: Explorations in the representation of knowledge*, pages 401–456. Morgan-Kaufmann, San Mateo, California, 1991. 28

Ulrich Callmeier, Andreas Eisele, Ulrich Schäfer, and Melanie Siegel. The DeepThought Core Architecture Framework. In *Proceedings of LREC*, May 2004. 22, 24

Jean Carletta. Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*, 22(2):249–254, 1996. 76

Martin Čmejrek, Jan Cuřín, Jiří Havelka, Jan Hajič, and Vladislav Kuboň. Prague Czech-English dependecy treebank: Syntactically annotated resources for machine translation. In Maria Teresa Lino, Maria Francisca Xavier, Fátima Ferreira, Rute Costa, and . . . ., editors, *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC'2004, Lisbon, Portugal, 26-28 May 2004*. ELRA - European Language Resources Association, 05 2004. 62

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46, 1960. 76

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan Sag. Minimal Recursion Semantics: An Introduction. *Research on Language and Computation*, 3(4):281–332, December 2005. 24

Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Andrea Schaerf. Reasoning in Description Logics. In Gerhard Brewka, editor, *Principles of Knowledge Representation*, pages 191–236. CSLI Publications, Stanford, California, 1996. 23

Jan Eijk and Hans Kamp. Representing Discourse in Context. In Johan Benthem and Alice Meulen, editors, *Handbook of Logic and Language*, pages 179–237. Elsevier, Amsterdam, 1996. 23

Jason Eisner and Damianos Karakos. Bootstraping Without the Boot. In *Proceedings of HLT/EMNLP*, pages 395–402, Vancouver, Canada, October 2005. 21

James Elliott, Robert Eckstein, Marc Loy, David Wood, and Brian Cole. *Java Swing*. O'Reilly, 2002. 63

Charles J. Fillmore. The Case for Case. In Emmon Bach and Robert T. Harms, editors, *Universals of Linguistic Theory*, pages 1–88. Holt, Rinehart, and Winston, New York, 1968. 62

Charles J. Fillmore. The Case for Case Reopened. In Peter Cole and Jerrold M. Sadock, editors, *Grammatical Relations*, volume 8 of *Syntax and Semantics*, pages 59–81. Academic Press, New York, San Francisco, London, 1977. 62

Gottlob Frege. Über Sinn und Bedeutung. In Mark Textor, editor, *Funktion - Begriff - Bedeutung*, volume 4 of *Sammlung Philosophie*. Vandenhoeck & Ruprecht, Göttingen, 1892. 23

Carsten Gnörlich. MultiNet/WR: A Knowledge Engineering Toolkit for Natural Language Information. Technical Report 278, University Hagen, Hagen, Germany, 2000. 57

Petr Hájek and Tomáš Havránek. *Mechanizing Hypothesis Formation; Mathematical Foundations for a General Theory*. Springer-Verlag, Berlin, Heidelberg, New York, 1978. 21

Jan Hajič, Barbora Vidová-Hladká, and Petr Pajas. The Prague Dependency Treebank: Annotation Structure and Support. In *Proceedings of the IRCS Workshop on Linguistic Databases*, pages 105—114, Philadelphia, Pennsylvania, 2001. University of Pennsylvania. 63

Jan Hajič, Otakar Smrž, Petr Zemánek, Jan Šnaidauf, and Emanuel Beška. Prague Arabic Dependency Treebank: Development in Data and Tools. In Mahtab Nikkhou, editor, *Proceedings of the NEMLAR International Conference on Arabic Language Resources and Tools*, pages 110—117, Cairo, 2004. ELDA. 62

Jan Hajič. Building a Syntactically Annotated Corpus: The Prague Dependency Treebank. In E. Hajičová, editor, *Issues of Valency and Meaning. Studies in Honour of Jarmila Panevová*, pages 106–132. Karolinum, Charles University Press, Prague, Czech Republic, 1998. 19

Jan Hajič, Barbora Vidová-Hladká, Jarmila Panevová, Eva Hajičová, Petr Sgall, and Petr Pajas. *Prague Dependency Treebank v. 1.0*. Linguistic Data Consortium, CDROM No. LDC2001T10 edition, 2001. 62

Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, and Marie Mikulová. Prague Dependency Treebank 2.0. CD-ROM, Linguistic Data Consortium, LDC Catalog No.: LDC2006T01, Philadelphia, Pennsylvania, 2006. 62

Jan Hajič, Kristýna Čermáková, Lucie Mladová, Anja Nedolužko, Jiří Semecký, Jana Šindlerová, Josef Toman, and Zdeněk Žabokrtský. Prague English Dependency Treebank (in progress), Est. 2009. 76

Eva Hajičová and Petr Sgall. Linguistic meaning and knowledge representation in automatic understanding of natural language. In *Proceedings of the 8th conference on Computational linguistics*, pages 67–75, Morristown, NJ, USA, 1980. Association for Computational Linguistics. 22

Eva Hajičová, Petr Sgall, and Barbara Partee. *Topic-Focus Articulation, Tripartite Structures, and Semantic Content*. Kluwer, Dordrecht, 1998. 35, 37, 39, 56

Eva Hajičová, Jarmila Panevová, and Petr Sgall. A Manual for Tectogrammatic Tagging of the Prague Dependency Treebank. Technical Report TR-2000-09, ÚFAL MFF UK, Prague, Czech Republic, 2000. in Czech. 24

Eva Hajičová, Jarmila Panevová, and Petr Sgall. Coreference in Annotating a Large Corpus. In *Proceedings of the Second LREC Conference*, pages 497–500, Athens, Greece, 2000. 85

Eva Hajičová, Petr Sgall, and Kateřina Veselá. Information Structure and Contrastive Topic. In Wayles Brown, Ji-Young Kim, Barbara H. Partee, and Robert A. Rothstein, editors, *Formal approaches to Slavic linguistics. The Amherst Meeting 2002*, pages 219–234, Ann Arbor, 2003. 36, 40

Eva Hajičová. Presupposition and Allegation Revisited. *Journal of Pragmatics*, 8:155–167, 1984. 38, 56

Sven Hartrumpf, Hermann Helbig, and Rainer Osswald. The Semantically Based Computer Lexicon HaGenLex – Structure and Technological Environment. *Traitement Automatique des Langues*, 44(2):81–105, 2003. 57

Sven Hartrumpf, Ingo Glöckner, and Johannes Leveling. University of Hagen at QA@CLEF 2007: Coreference Resolution for Questions and

Answer Merging. In *Results of the CLEF 2007 Cross-Language System Evaluation Campaign, Working Notes for the CLEF 2007 Workshop*, Budapest, Hungary, 2007. 57

Hermann Helbig and Sven Hartrumpf. Word Class Functions for Syntactic-Semantic Analysis. In *Proceedings of the 2nd International Conference on Recent Advances in Natural Language Processing (RANLP'97)*, pages 312–317, Tzigov Chark, Bulgaria, September 1997. 57

Hermann Helbig. Syntactic-Semantic Analysis of Natural Language by a New Word-Class Controlled Functional Analysis. *Computers and Artificial Inteligence*, 5(1):53–59, 1986. 57

Hermann Helbig. *Knowledge Representation and the Semantics of Natural Language*. Springer, Berlin, Germany, 2006. 20, 21, 25, 26, 27, 28, 29, 31, 32, 34, 35, 39, 41, 70, 87, 95, 101, 105

Aleš Horák. *The Normal Translation Algorithm in Transparent Intensional Logic for Czech*. PhD thesis, Faculty of Informatics, Masaryk University, Brno, Czech Republic, 2001. 24

Ian Horrocks and Peter F. Patel-Schneider. Reducing OWL Entailment to Description Logic Satisfiability. *Journal of Web Semantics*, 1(4):345–357, 2004. 26

Ray Jackendoff. *Semantic Structures*. The M.I.T. Press, Cambridge, MA, 1990. 62, 100

Roman Jakobson. *Selected Writings*. Mouton, The Hague, 1971. 62

Jurgen M. Janas and Camilla B. Schwind. Extensional Semantic Networks: Their Representation, Application, and Generation. In N. V. Findler, editor, *Associative Networks: Representation and Use of Knowledge by Computers*, pages 267–302. Academic Press, Orlando, 1979. 29

Petr Jirků and Jan Hajič. Inferencing and search for an answer in TIBAQ. In *Proceedings of the 9th conference on Computational linguistics – Volume 2*, pages 139–141, Prague, Czechoslovakia, 1982. 35, 37

David Kaplan. On the Logic of Demonstratives. *Journal of Philosophical Logic*, 8(1):81–98, 1979. 62

Sergei Karcevskij. Du dualisme asymétrique du signe linguistique. *Travaux du Cercle linguistique de Prague*, 1:88–93, 1929. 22

Paul Kingsbury, Martha Palmer, and Mitch Marcus. Adding Semantic Annotation to the Penn TreeBank. In *Proceedings of the Human Language Technology Conference (HLT'02)*, 2002. 62

Klaus Krippendorff. *Content analysis: an introduction to its methodology*. Sage Publications, Newbury Park, CA, 1980. 76

Ivana Kruijff-Korbayová. *The Dynamic Potential of Topic and Focus: A Praguian Approach to Discourse Representation Theory*. PhD thesis, ÚFAL, MFF UK, Prague, Czech Republic, 1998. 35

Geert Jan Kruijff. *A Categorial-Modal Logical Architecture of Informativity*. PhD thesis, Faculty of Mathematics and Physics, Charles University, 2001. http://www.coli.uni-sb.de/ gj/dissertation.phtml. 24

Henry Kučera. Computational analysis of predicational structures in English. In *Proceedings of the 8th conference on Computational linguistics*, pages 32–37, Morristown, NJ, USA, 1980. Association for Computational Linguistics. 62

Bo Leuf and Ward Cunningham. *The Wiki Way. Quick Collaboration on the Web*. Addison-Wesley, Reading, Massachusetts, 2001. 66

Dekang Lin and Patrick Pantel. Discovery of Inference Rules for Question-Answering. *Natural Language Engineering*, 7(4):343–360, 2001. 20

Markéta Lopatková, Zdeněk Žabokrtský, and Karolína Skwarska. Valency Lexicon of Czech Verbs: Alternation-Based Model. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, volume 3, pages 1728–1733. ELRA, 2006. 62

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19(2):313–330, 1993. 62, 76

Igor A. Mel'čuk. *Dependecy Syntax: Theory and Practice*. State University of New York Press, 1988. 25

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. Annotating Noun Argument Structure for NomBank. In *Proceedings of LREC-2004*, Lisbon, Portugal, 2004. 62

Marie Mikulová, Alevtina Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolářová, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, Magda Razímová, Petr Sgall, Jan Štěpánek, Zdeňka Urešová, Kateřina Veselá, and Zdeněk Žabokrtský. Annotation on the tectogrammatical level in the Prague Dependency Treebank. Annotation manual. Technical Report 30, ÚFAL MFF UK, Prague, Czech Rep., 2006. 41, 62, 85

Marc Moens and Mark Steedman. Temporal Ontology and Temporal Reference. *Journal of computational Linguistics*, 14:15–28, 1988. 100

Richard Montague. Pragmatics and Intensional Logic. *Semantics of Natural Language*, 1972. 23, 62

Václav Novák and Jan Hajič. Perspectives of Turning Prague Dependency Treebank into a Knowledge Base. In *Proceedings of the LREC Conference*, Genova, Italy, 2006. 24

Václav Novák. Towards Logical Representation of Language Structures. *The Prague Bulletin of Mathematical Linguistics*, 82:5–86, 2004. 24

Václav Novák. On Distance between Deep Syntax and Semantic Representation. In *Proceedings of Frontiers in Linguistically Annotated Corpora*, pages 78–85, Sydney, Australia, 2006. 66

Václav Novák. Cedit – Semantic Networks Manual Annotation Tool. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 11–12, Rochester, New York, April 2007. Association for Computational Linguistics. 63

Václav Novák. Large Semantic Network Manual Annotation. In *Proceedings of 7th International Workshop on Computational Semantics*, pages 355–358, Tilburg, Netherlands, 2007. 62

Petr Němec. Annotation of Temporal Relations within a Discourse. In *Lecture Notes in Computer Science, Proceedings of the 9th International Conference, TSD 2006*, number 4188 in Lecture Notes In Computer Science, pages 181–188. Springer-Verlag Berlin Heidelberg, 2006. 100

Petr Pajas and Jan Štěpánek. A Generic XML-Based Format for Structured Linguistic Annotation and Its Application to Prague Dependency Treebank 2.0. Technical Report 29, UFAL MFF UK, Praha, Czech Republic, 2005. 58, 63

Jarmila Panevová, Eva Benešová, and Petr Sgall. Čas a modalita v češtině. *AUC - Philologica Monographia*, 34, 1971. 100

Jaroslav Peregrin. Topic, Focus and the Logic of Language. In *Sprachtheoretische Grundlagen für die Computerlinguistik (Proceedings of the Goettingen Focus Workshop, 17. DGfS)*, Heidelberg, 1995. IBM Deutschland. 35, 38, 56

Jaroslav Peregrin. Topic-Focus Articulation as Generalized Quantification. In P. Bosch and R. van der Sandt, editors, *Proceedings of "Focus and natural language processing"*, pages 49–57, Heidelberg, 1995. IBM Deutschland. 38, 56

Jaroslav Peregrin. Topic and Focus in a Formal Framework. In Barbara H. Partee and Petr Sgall, editors, *Discourse and Meaning*. John Benjamins Publishing Company, Amsterdam/Philadelphia, 1997. 38

Arthur N. Prior. *Past, Present and Future*. Oxford University Press, Oxford, UK, 1967. 100

M. Ross Quillian. Semantic Memory. In Marvin Minsky, editor, *Semantic Information Processing*, pages 216–270. MIT Press, Cambridge, MA, 1968. 29

RDF Core Working Group. Resource Description Framework (http://www.w3.org/RDF/), 2007. 26

Hans Reichenbach. *Elements of Symbolic Logic*. University of California Press, Berkley CA, 1947. 100

Ferdinand de Saussure. *Cours de linguistique générale*. Payot, Paris, 1916. 13

William A. Scott. Reliability of Content Analysis: the Case of Nominal Scale Coding. *Public Opinion Quarterly*, 19:321–325, 1955. 76

Petr Sgall, Eva Hajičová, and Jarmila Panevová. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. D. Reidel, Dordrecht, The Netherlands, 1986. 17, 24, 35, 37, 38

Petr Sgall, Jarmila Panevová, and Eva Hajičová. Deep Syntactic Annotation: Tectogrammatical Representation and Beyond. In Adam Meyers, editor, *Proceedings of the HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 32–38, Boston, Massachusetts, May 2004. Association for Computational Linguistics. 17

Sun Microsystems, Inc. *Java Platform, Standard Edition 6*. Sun Microsystems, Inc., 2007. http://java.sun.com/javase/6/webnotes/README.html. 63

Lorraine Tanabe, Natalie Xie, Lynne H. Thom, Wayne Matten, and W. John Wilbur. GENETAG: a tagged corpus for gene/protein named entity recognition. *BMC Bioinformatics*, 6 Suppl 1, 2005. 62

Lucien Tesnière. Comment construire une Syntaxe. *Bulletin de la Faculté des Lettres de Strasbourg*, pages 219–229, 1934. 24

Pavel Tichý. *The Foundations of Frege's Logic*. Walter de Gruyter & Co, Berlin/New York, 1988. 23

Ludwig Wittgenstein. *Philosophische Untersuchungen*. Suhrkamp Verlag, Frankfurt am Main, 1953. 13, 17, 27, 61, 75, 99

Zdeněk Žabokrtský. Resemblances between Meaning-Text Theory and Functional Generative Description. In *Proceedings of the 2nd International Conference of Meaning-Text Theory*, pages 549–557, 2005. 25

Index