Scenes-and-Frames Semantics and Its Possibilities in Building a Knowledge Database of the Slovak Language

Martina Ivanová[1,2], Peter Kostelník[3,*]

[1]Ľ. Štúr Institute of Linguistics, Slovak Academy of Sciences, Panská 26, 811 01 Bratislava, Slovakia
[2]Faculty of Arts, University of Prešov, 17. novembra 1, 080 01 Prešov, Slovakia
[3]Xolution, s. r. o., Štefánikova 20, 04001 Košice, Slovakia

Abstract

The presentation will focus on the introduction of the SENSE project (Semantic Analysis of the Slovak Language), which is being carried out as a collaboration of the Ľ. Štúr Institute of Linguistics, Slovak Academy of Sciences and Xolution. The SENSE project aims to design a knowledge database for the Slovak language which would describe how individual words or phrases are transformed into a semantic representation, and the creation of software that can use these datasets to interpret texts that the machine has not been trained on. To achieve this purpose, the Scenes-and-Frames Semantics as introduced in the FrameNet database will be applied. The tools that could assist in the development of such a dataset include databases which have arisen from the research on valency properties as developed in the tradition of Slovak linguistics. The presentation will show how these methodologies can be mutually beneficial. In the presentation, we will also introduce practical examples of the transformation of text into a frame representation through the experimental software in its pilot stage of development.

Keywords

FrameNet, valency analysis, frame semantics, knowledge modeling

1. Introductory remarks

In this paper, the basic principles of the grant project SENSE (SEmantic aNalysis of Slovak languagE) are dealt with. The goal of the project is to create software and a mapping dataset that would be able to work with the Slovak language at the semantic level. The existence of a theoretical framework and a model of semantic representation is one of the prerequisites for semantic text analysis, which remains one of the most difficult tasks in NLP. At the same time, the existence of such a model is a prerequisite for building so-called general artificial intelligence (the goal is to create a model that will enable understanding of natural language as humans understand it). According to the latest neuro- and psycholinguistic research, humans conceptualise continuous experience into discrete fragments, so-called micro-situations [1]. In every language, these representations of events are schematised in the form of abstract models. One of the best-known semantic models is the theoretical approach known as frame semantics, which is being developed as part of the FrameNet project [2].

2. What´s FrameNet got to do with it

One of the basic theoretical and methodological starting points of the concept of frame semantics is the assumption that the meanings of words can be represented by situations, their relationships, and roles. "Meanings are relativised to scenes." This statement illustrates one of the basic conceptual foundations of the concept of frame semantics [3, p. 59]. In this approach, the basic idea is the existence of semantic-syntactic models based on the same semantic-syntactic environment of a unit and on the same modifications (transformations) that a given unit can undergo.

ITAT'25: Information Technologies – Aplications and Theory, September 26–30, 2025, Telgárt, Slovakia
*Corresponding author.
martina.ivanova@unipo.sk (M. Ivanová); peter.kostelnik@xolution.sk (P. Kostelník)
0000-0001-7654-300X (M. Ivanová); 0009-0004-6152-5772 (P. Kostelník)

In the concept of frame semantics, verb frames are understood as conceptual structures that contain references to encyclopaedic knowledge and cultural background. Paraphrases such as “X causes Y to become Z” are defined in this approach as structural (frame) meanings. Since the mapping between semantics and syntax is realised through construction, rather than an isolated lexical unit, there are syntactically relevant aspects of meaning that are activated precisely in the construction. Constructions are thus understood as units that have their own semantics, similar to lexical units [4].

For example, expressions belonging to the `Cause_harm` frame describe situations in which an element with the semantic role `Agent` or `Cause` injures a participant labelled with the semantic role `Victim`. Instead of the role `Victim`, the frame can also express `Body_part`, which is directly affected by the action to the greatest extent. In such cases, the element with the role `Victim` acts as a genitive complement to `Body_part`.

The elements of the frame that are “markers” of the frame, i.e., they evoke its application in the text (e.g., the verbs *beat*, *crash*, *smash*, etc.), are called lexical units. Frames can be of varying complexity and can be composed of different numbers of elements and lexical units. The basic goal in constructing semantic frames is to show how the elements of the frame are related to each other.

FrameNet consists of several components:

- Definition: contains a description of each frame; a semantic frame is defined as a semantic representation of an event, situation, or relationship that consists of several elements, each of which has its own semantic role in the frame
- Semantic type: specification of the semantic type, e.g. `Event` in the case of the `Cause_harm` frame
- Frame elements: a frame element is a type of participant (role) in a given frame with certain types of semantic links; Frame elements are classified as core, peripheral, or extra-thematic; in the case of the `Cause_harm` frame, the core elements include the roles `Agent`, `Body_part`, `Cause`, `Victim`, while the peripheral and extra-thematic elements include, for example, `Duration`, `Instrument`, `Frequency`, etc.
- Frame-frame relations: these reflect the method of hierarchization when postulating frames (these are inheritance relations, which indicate the possibility of defining a frame as a subframe of a more generally postulated frame, e.g. the `Cause_harm` frame inherits characteristics from the `Cause_benefit_or_detriment` frame, and the `Cause_harm` frame in turn inherits characteristics from the `Corporal_punishment` frame), but also the possibility of modifying the lexical units of a given frame using inchoative, causative, and other units
- Lexical units: a lexical unit is a word with a fixed meaning that evokes a given frame
- Examples of annotated sentences

FrameNet has had a considerable impact on the field of computational linguistics [5]. Above all, it has paved the way for the task of automatic semantic role labelling (ASRL) introduced by the seminal work of Gildea and Jurafsky [6].

3. What´s valency got to do with it

In linguistic studies, the concept of valency has been developed for a long time. Valency theory originated in the work of the French structuralist Lucien Tesnière [7], in whose theory of dependency grammar valency plays a considerable role. Valency theory takes an approach towards the analysis of sentences that focuses on the role that certain words play in sentences with respect to the necessity of occurrence of certain other elements. The term valency comprises elements that are word-specific in the sense that their occurrence cannot be explained in terms of generalisable properties; rather, their occurrence is dependent on an individual lexical item (called a governing element, or predicator). Complements are those elements that satisfy the valency requirements of a predicator at the formal syntactic level. However, the syntactic valency finds its corollary at the level of semantics, so that a distinction between syntactic and semantic valency can be made. The Berkeley FrameNet project has led to a particular treatment of the concept of valency. The FrameNet project is dedicated to producing valency descriptions of frame-bearing lexical units (LUs), in both semantic and syntactic terms, and it bases this work on attestations of word usage taken from a very large digital corpus. The semantic descriptors of each valency pattern are taken from frame-specific semantic role names (called frame elements), and the syntactic terms are taken from a restricted set of grammatical function names and a detailed set of phrase types.

The treatment of valency in the FrameNet database differs from certain other electronic lexical resources in several ways, by:

- relying on corpus evidence
- basing the semantic layer of valency on an understanding of the cognitive frames that motivate and underlie the meanings of each lexical unit
- semantic frames are regarded as primary for the description and analysis of meaning (and its syntactic relevance), and semantic roles are defined in terms of their semantic frames
- recognising various kinds of discrepancy between units on the semantic/functional level and patterns of syntactic form [8].

Despite several fundamental differences, it is possible to identify a number of points of contact where the conceptual foundations of frame semantics and the valency approach converge in the Slovak valency dictionary (Valenčný slovník slovenských slovies na korpusovom základe, VSSSKZ [9]).

(1) As the title shows, the valency parameters in VSSSKZ are based on the corpus data, similarly to FrameNet.

(2) VSSSKZ does not work with so-called cognitive role shifting; the definition of roles is cognitive in its nature and is based on the semantic classification of the verbs. The principle of cognitive role shifting leads to the direct object being labelled identically as `Patient` in Czech valency dictionary VALLEX [10], even for lexical units from two different semantic classes, e.g. *naložit 1 (location)* and *naložit 2 (providing)*, while in FrameNet and VSSSKZ, the object participant is assigned a different semantic role in these cases (the semantic role `Theme` in the `Placing` frame and the semantic role `Goal` in the `Filling` frame in FrameNet, and the semantic role `Manipulator` in the class of manipulative verbs and the semantic role `Modifier` in the semantic class of modifying verbs in VSSSKZ).

(3) Similarly to FrameNet, VSSSKZ is characterized by the systematic classification of all lexical items into semantic classes. For example, in situations encoding perceptual events, FrameNet distinguishes between the frames `Perception_experience` and `Perception_active`. The semantic frame `Perception_experience` corresponds to a situation where the perceptual activity is not intentional, while the semantic frame `Perception_active` denotes a situation where perceivers intentionally focus their attention on an entity or phenomenon in order to gain a perceptual experience. The semantic specification is then reflected in the definition of the elements of the frame, cf. the semantic frame `Perception_experience` with the roles `Perceiver_passive` and `Phenomenon` and the frame `Perception_active` with the roles `Perceiver_agentive` and `Phenomenon`. A similar approach can also be found in VSSSKZ, in which process perceptual verbs (e.g., *vidieť*) and action perceptual verbs (e.g., *pozorovať*) are treated as units from two distinct semantic classes. The left-intentional participant is assigned the role of processual perceiver in case of process perceptual verbs, and the role of agentive perceiver in action perception verbs.

FrameNet works with situation-specific terms such as injury, perpetrator, etc., when defining semantic roles, rather than limiting itself to traditionally defined thematic roles such as agent, patient, goal, etc. This approach argues, on the one hand, that the number of traditionally defined semantic roles is not sufficient for semantic description and, on the other hand, that it is difficult to postulate criteria for mapping that would allow specific specifications to be "fitted" into traditionally defined roles. The VSSSKZ does not abandon traditionally defined roles but works with two levels of abstraction. Within the 34 semantic micro-groups, the semantic macro-roles are defined on the left side of the intention: agentive – processual – stative (the criterion is the status of the verbal lexeme in terms of action, process, or state) are defined on the left side of the intention, and on the right side of the intention, the macro-roles of `Patient – Result – Target – Source – Content – Relation` are delimited. VSSSKZ proposes roles that are neither as general as the semantic roles proposed in the Czech valency dictionary VALLEX [10], nor as specific as the thousands of potential verb-specific roles.

(4) In FrameNet, valency patterns are described in the Valence Pattern Tables for individual units. Both FrameNet and VSSSKZ provide the means of assigning partial interpretations to valents that are conceptually present, but syntactically unexpressed. FrameNet distinguishes three types of missing elements, abbreviates DNI, INI, and CNI. DNI stands for "definite null instantiation" and marks FEs that are unrealised but which have to be recoverable from the context. An example is the FE RECIPIENT in a sentence such as *"The prize was given by an international committee"*. INI stands for "indefinite null instantiation" and covers FEs that are merely existentially bound, an example of which is the FE THEME in *"Sure, I gave to the Red Cross last year – everybody did"*. CNI stands for "constructional null instantiation" and marks all omissions licensed by a syntactic construction. A typical case is the omission of agentive FEs in the imperative construction, as in *"Give a generous gift to Goodwill today"*.

In the schema describing the semantic valency structure (so-called intention structure, IŠ) of the Slovak verb *dať 3* (see Table 1), the possibility not to express the participant with the semantic role `Theme` (labelled as AKC – abbreviation for acceptor, the label for the semantic role used in VSSSKZ) is marked by brackets. This situation is also reflected by the corpus examples.

Table 1
Valency structure of the verb *dať 3* in VSSSKZ.

VŠ: Sn – VF – Sa – (Sd) – ADVtemp – ADVmot **IŠ: AGdon – PRED – MANT – (AKC)** Sn: [HUM] AGdon Sa: [KONKR/PARS ANIM] MANT Sd: [ANIM] AKC
Okrem toho musia chovatelia dávať vzorky na veterinárnu kontrolu.
Počas Silvestra, aj keď som mu samozrejme dal lieky na upokojenie, veľmi utrpel.
Pred plánovaným chirurgickým výkonom môže dať pacient krv sám sebe, ak nemá závažné ochorenie, ktoré to vylučuje.

In the FrameNet approach, frame elements are categorised as either core or non-core elements. Core elements are defined as elements that are conceptually necessary for a given frame. In contrast, peripheral elements of the frame are not unique to the given frame and can usually occur in any frame (typically expressions of time, place, manner, purpose, attitude, etc.), and extra-thematic elements of the frame have no direct relation to the situation identified with the frame, but provide new information, often showing how an event represented by one frame is part of an event involving another frame [2]. In VSSSKZ non-valency complements that are typically associated with the given verb are present in the valency structure (VŠ) and marked by non-bold script (in this case, temporal ADVtemp and purpose ADVmot adverbials are marked as non-valency, i.e., non-core, complements for the given verb).

The challenges of semantics-processing tasks lie in the necessity to move away from carefully hand-crafted, domain- dependent systems toward robustness and domain independence [6, p. 245]. In the next section, the semantic analyser for Slovak is described, the tool for identifying the semantic relationships, or semantic roles, filled by constituents of a sentence within a semantic and valency frame.

4. Dataset

One of the goals of the SENSE project is to create a dataset for mapping the Slovak language into the FrameNet format. We have chosen a declarative approach, which does not focus on creating annotated sentence examples but rather on building a formal knowledge model.

The core of this model consists of semantic frames, defined as independent, modular units. Each unit contains a set of formal rules and templates that explicitly define the mapping between surface syntactic structures and the semantic roles of the given frame. The aim is therefore not to deliver a Slovak version of FrameNet in the form of an annotated corpus, but to provide a set of formal specifications that describe semantic frames and define the exact mechanism for their realisation (instantiation) from text. These structures serve as a directly usable knowledge base for automatic semantic analysis.

The chosen approach offers several significant advantages:

- Complete control and transparency. The knowledge is explicitly defined in the form of readable rules. This allows for full control over the semantic interpretation process and eliminates the "black box" nature typical of statistical models.
- High scalability and flexibility. The system can be easily extended. Adding support for new linguistic expressions with the same meaning only requires the definition of a new materialisation alternative. Similarly, correcting and refining existing definitions is straightforward.
- Support for modularity and domain adaptation. The approach naturally supports modularity. It is possible to create specialised, domain-specific knowledge bases that may contain different interpretations of general frames or define entirely new, narrowly focused frames. These modules can be combined and adapted as needed.
- Flexible maintenance and immediate deployment of changes. One of the key practical requirements is the ability to dynamically extend and repair the system's knowledge on the fly. In practice, it is often necessary to deploy improvements almost immediately. The declarative approach allows this because any change or addition of a rule takes effect instantly. This is a fundamental advantage over statistical approaches, where even minimal changes in training data require a complete and time-consuming retraining of the entire model.

A disadvantage, however, is that the dataset will not be well-suited for processing by statistical models, as it does not contain explicit annotated examples.

We anticipate that in the future it will be possible to extend the frame dataset with annotated texts and contribute to the creation of a Slovak FrameNet in its standard format. We also expect that the created knowledge model can be used as a supporting tool for the semi-automatic construction of a large annotated corpus.

5. Semantic analyser

5.1. Motivation

Xolution has many years of experience in developing and implementing chatbots designed for various specific task domains. Based on practical experience from deployments in commercial environments, it has become evident that an effective domain-specific chatbot must be capable of handling several typologically different tasks, which require varying degrees of interaction, understanding, and integration with external systems. For this reason, we approach chatbot design as a modular system, where each module implements a separate logic for solving a particular class of tasks. Such a design enables flexibility, easy expansion, and adaptation of the system to the specific requirements of the domain. Typical and most frequent tasks include, for example:

- Answering complex questions that require access to external structured data sources (ontologies, databases): *"Find me at least three-star hotels in Košice near the city centre with Wi-Fi, breakfast, and a price for a double room up to 100 EUR."*

- Handling specific tasks where the chatbot takes the initiative in the dialogue and asks questions, e.g., diagnosing a device malfunction, providing step-by-step assistance in problem solving, or data collection such as filling out forms or generating documents.
- Interacting with external systems (directly or via API calls), for example, adding entries to a calendar, sending emails, or saving service requests.

Since the chatbot always processes text input, all these tasks share a common denominator. In order for the system to fulfil a request, it must have a mechanism that converts free text into a structured form suitable for machine processing.

First, each request requires the recognition of the user's intent. Let us take the example of searching for accommodation *"Book me a double room in a three-star hotel in Košice with breakfast for up to 100 EUR"*. The system needs to extract the intent and relevant entities from the text. The example of target intent structure is described in Table 2.

Table 2
Intent example.

`intent:`	find-hotel-booking
`stars`	3
`location`	Košice
`room-type`	double
`meal-plan`	breakfast
`price-limit`	100
`check-in-date`	?
`check-out-date`	?

The identified structure represents a clear formal specification of the request, ready for automatic processing. The system knows which items are missing and need to be obtained to complete the request. Based on the recognised intent, it can also send the completed request for further processing to a specific module responsible for handling the task. Working with formal structures derived from free text offers several advantages, for example:

- Increased accuracy and reliability. The chatbot stops guessing and start working with facts. By recognising specific intents, ambiguities are eliminated, e.g., *"Book me a hotel/pizza"*.
- Simplified chatbot logic. When the chatbot operates with a clearly defined intent schema, universal and general implementations of service mechanisms can be used. For example, a mechanism for obtaining missing information in a request (so-called slot-filling) will work in the same way for every intent. Additionally, the logic is separated from the content and can be modified independently.
- Easy integration with external systems. Converting a clearly defined structure into an API call or a database query is relatively straightforward.
- Personalisation. If the user's history of actions is known, preferences frequently repeated in the past can be suggested.
- Support for multiple languages. Text in any supported language is always mapped to the same structure, so the request processing logic remains unchanged.
- Support for modularity. Individual expert modules can be shared by different chatbots.
- Better scalability and maintenance. To extend the system with new capabilities, it is sufficient to define a new intent, its structure, and the method of mapping text to that structure.

Mapping, or the conversion of text into machine-processable structures, is a critical part of input understanding. There are many approaches and solutions to this task, such as platform-as-a-service tools (DialogFlow [11], Wit.ai [12]), open-source frameworks (Rasa NLP [13]), libraries (SpaCy [14]), or fine-tuning language models (BERT for text annotation or GPT directly for structure generation). At

Xolution, we use a proprietary rule-based sequence resolver that takes into account morphological data, synonyms, and named entities.

Text for intent recognition can be formulated in many ways, for example: *"find/search hotel/accommodation/facility, ..."*, *"I want to stay at ..."*, *"I'm looking for a room, ..."*, *"I would like a bed, ..."* or *"hotel Košice, ..."*. Approaches based on isolated examples share a common disadvantage: recognising each intent requires many examples. To add support for a new formulation, statistical models need to be completely retrained, while rule-based systems often become opaque and prone to rule conflicts as the knowledge model grows. Furthermore, example-based modeling may not fully capture the overall meaning of the text and can lead to false positives, for example: *"I wasn't thinking about booking a hotel in Košice."*

To increase robustness, accuracy, and interpretability of converting text into target structures, we therefore propose the use of frame semantics. We selected FrameNet as the target format, which models typical situations along with their participants and supplementary semantic roles (circumstances, place, time, manner) through frames. Compared to other well-known standards (such as VerbNet [15] or PropBank [16]), FrameNet offers richer and finer semantic differentiation of frames (e.g., distinguishing between buying, selling, donating, or exchanging). Thanks to the semantic modeling of participants and roles, it is more readable and comprehensible to humans. Representing text using frames offers many anticipated advantages, for example:

- A comprehensive and readable view of the input text. Text converted into a frame-based representation takes the form of a graph with a clearly distinguished structure of main and subordinate clauses. Individual parts of the text are represented by frames, which also enables more precise detection of the user's intents.
- Significant reduction in training examples. Different textual formulations with the same meaning are semantically represented by the same frames. Training examples are modelled using the resulting semantic structure and are no longer dependent on specific formulations.
- Greater transparency, interpretability, and explainability of the system's decisions. If the system does not behave as expected, it is much easier to identify and fix errors, while also clearly explaining how the system arrived at its decision.
- Support for multilingual solutions. Since the system operates at the level of frame semantics, it is possible to adapt additional languages without changing the system logic.

5.2. Implementation

The proposed approach to semantic analysis is conceptually inspired by formalisms for deep semantic representation, particularly Abstract Meaning Representation [17] and the more recent Universal Meaning Representation [18]. Both of these representations are based on converting natural language text into a structured, directed acyclic graph that captures the semantic structure of frames as well as the description of complex structured objects.

Let us consider the following example sentence (some translations of examples into English may be awkward, as we aim to preserve the syntactic constructions used in Slovak): *"Peter, ktorý v Xolution vyvíja chatboty, sediac na zelenej záhrade videl, že Zuzana písala knihu pre deti."* / *"Peter, who develops chatbots at Xolution, sitting in the green garden, saw that Zuzana was writing a book for children."*

The desired target form for the sentence is illustrated in Figure 1, where the representation of the main meaning is shown on the left and the semantic adjuncts on the right. The main meaning is identified as the PERCEPTION-EXPERIENCE frame (someone perceives something), whose `experiencer` is *Peter*, and the perceived `phenomenon` is not a simple entity but an entire additional event described by the STATEMENT frame (someone tells or writes something).

The target format is expected to be capable of representing the meaning of text using frames, whose roles may include other frames or complex structured entities. Structured entities - such as physical objects - can be described by attributes and properties, which may in turn contain other entities or frames. The expected output is therefore a recursive structure, where frames may contain entities, and entities may contain frames.

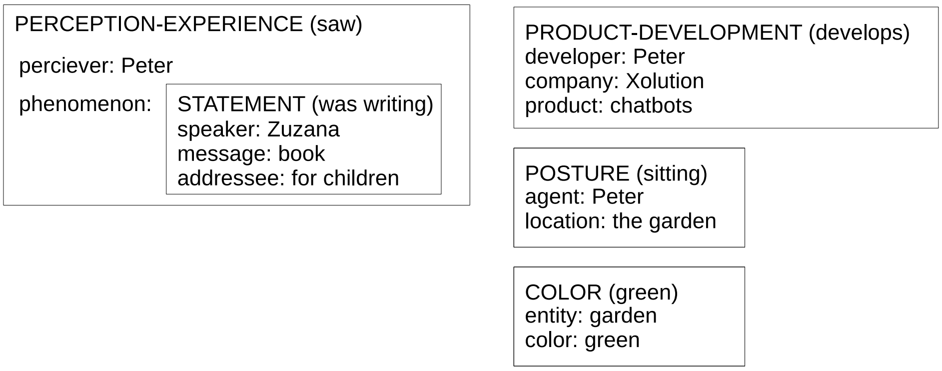

Figure 1: Example of desired outcome.

5.2.1. Text Annotation

Text analysis begins with the annotation of individual words and fixed phrases. The annotation process itself is carried out in several consecutive steps:

1. Identification of basic data types using regular expressions (regex). The parser in various formats recognises numbers, dates, times, emails, URLs, phone numbers, emoticons, etc.
2. Morphological analysis assigns to each word annotations for potentially multiple parts of speech with corresponding morphological tags.
3. Application of gazetteers for recognising basic simple named entities (Named Entity Recognition, NER), such as first names, geographic names, abbreviations, product code identifiers, etc.
4. Identification of taxonomic classes, where one word may be mapped to multiple taxonomic concepts. A word does not have to be mapped to any taxonomy. In such cases, the word is assigned a general taxonomic type. Taxonomic classes are also assigned to entities recognised by NER.
5. Identification of frames that may be evoked by the given word.

As mentioned, the prototype uses taxonomies that organise individual concepts into hierarchical structures. Taxonomies help to create a clear, systematic arrangement of concepts, facilitating search, understanding, knowledge management, and inference mechanisms. Separate taxonomies are used for organising entities and frames. A simple example of taxonomies is illustrated in Figure 2.

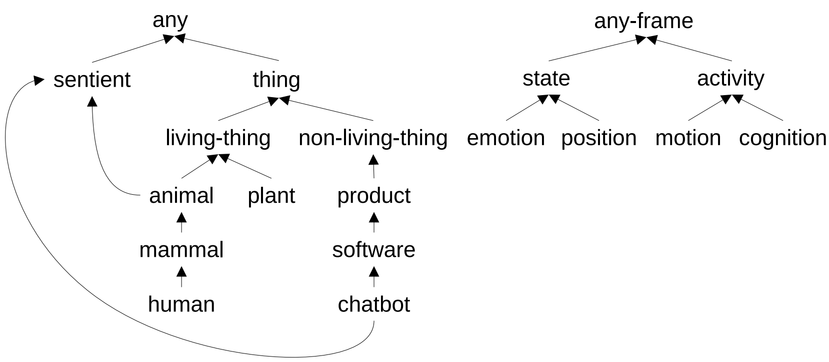

Figure 2: Example of entity and frame taxonomy.

Taxonomies themselves support multiple inheritance. Likewise, in the case of polysemy, one word can be classified into multiple taxonomic classes. An important advantage of using taxonomies is the ability to model additional knowledge (the definition of frames and complex entities) directly using taxonomic concepts instead of individual words. This approach significantly generalises, simplifies, and clarifies the process of knowledge modeling.

5.2.2. Syntactic analysis

The proposed prototype utilises a syntactic parser based on Augmented Transition Networks (ATN) [19], implemented in Xolution. The parser is adapted to the syntactic constructions of the Slovak language, taking into account its rich morphology, constructional variability associated with meaning change (which is addressed for some verbal units in the VSSSKZ), free word order, and a high degree of ambiguity. During analysis, the parser allows for on-the-fly validation and pruning of unpromising structures, which both improves the efficiency of the parsing process and reduces the number of output alternatives. It is also possible to consult individual syntactic structures with external knowledge during the analysis, enabling a focus on more meaningful constructions. The parser recognizes main and subordinate clauses, including their conjunctions, and pays special attention to syntactic networks for phrase analysis.

The analyser generates multiple possible syntactic structures as output. These are transformed into dependency graphs, which serve as input to the semantic analysis of frames and structured entities. The Universal Dependencies convention [20] is used to label edges in the dependency graphs. For more precise modeling of the inflection of the Slovak language, some edge labels have been extended with additional information, such as grammatical case (fall number) (e.g., obl1, …, obl7).

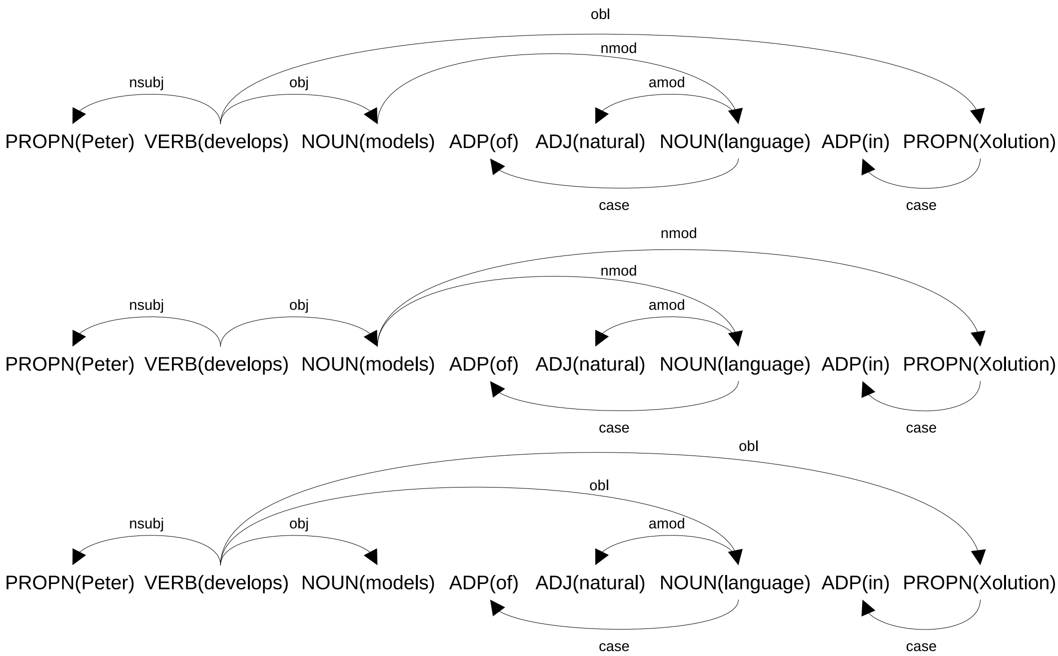

Figure 3: Examples of dependency graph alternatives.

In practice, generating multiple alternative parses is not a drawback but a necessity, especially in the case of phrases. As an example, consider the well-known issue of prepositional phrase attachment ambiguity. Three examples of different syntactic alternatives for the sentence *"Peter develops models of natural language at Xolution."* are shown in Figure 3. To create a correct semantic representation, all alternatives must be analysed. For the example sentence, the correct syntactic construction is represented by the first graph in the figure, where the word *"models"* is expanded by the prepositional phrase *"of natural language"* and the phrase *"in Xolution"* is attached to the main verb. However, if the main verb was changed, the situation could differ, and another syntactic alternative might be more appropriate.

The used ATN parser is relatively efficient at analysing common sentences but fails to recognise supplementary colloquial structures, such as various parenthetical clauses, parataxis, or embedded discourse, for example: *"I went, **peter said,** to work"*, *"find me, **could you please,** information"* or *"find me, **hello, btw,** information"*.

For the purposes of the prototype, the parser employed is sufficient, but in practice, it is necessary to be able to analyse arbitrarily complex texts. Therefore, we are also experimenting with a neuro-symbolic approach, where a neural parser is responsible for segmenting the text into main clauses, subordinate clauses, and insertions, and the symbolic parser performs a detailed analysis of phrase alternatives.

5.2.3. Frames

FrameNet defines a semantic frame as a conceptualised type of situation, event, or state that includes the semantic roles of participants and properties of the described concept (so-called frame elements). In the proposed prototype, a frame is defined as a structure that describes:

- Expected semantic roles (so-called frame elements): actors, attributes, properties. For each role, recommended taxonomic classes can be defined. An object mapped to a role may or must belong to one of the specified classes. It is also possible to define the valence of a role, which determines the importance of that role for the frame. In FrameNet, roles are divided into core and non-core elements. For greater flexibility, the prototype allows the valence to be defined numerically.
- Frame modality. It often happens that a frame describes a more general situation where it is necessary to distinguish finer shades of meaning. An example is the frame PERCEPTION-EXPERIENCE (someone perceives something), which collectively describes general perception, smell, taste, hearing, sight, touch, etc. Modalities allow these nuances to be distinguished. Modalities can be hierarchically organised and are defined in a separate taxonomy assigned to the frame.
- Constraints on the co-occurrence of roles that define the validity of the frame. Some roles must appear together within a frame, while others may be mutually exclusive.

Table 3
Example of frame definition.

PRODUCT-DEVELOPMENT		
`role`	`taxonomy`	`valency`
`developer`	`person, developer`	`0`
`product`	`product`	`0`
`company`	`company`	`1`
`beneficiary`	`any`	`2`
`location`	`any`	`2`

An illustrative example of the PRODUCT-DEVELOPMENT frame definition is shown in Table 3. The frame describes who develops which product, at which company, for whom, and in which location.

Each frame has defined alternatives for its realisation. Each alternative describes a way of mapping a specific surface syntactic structure to a frame representation. This makes it possible to construct a frame from differently formulated texts, such as *"mám strach" / "I have fear"* or *"bojím sa" / "I'm scared"*. Different formulations may require different syntactic structures to be recognised in order to correctly identify frame roles. Each alternative includes frame triggers (i.e., the words that evoke the frame) and mapping rules that associate syntactic structures with the corresponding roles.

Triggers can be specified as combinations of main and auxiliary words. For example, there is a clear semantic difference between *"Peter **vyvíja**" / "Peter develops"* and *"Peter **sa vyvíja**" / "Peter is evolving"*. A specific modality can be assigned to a given trigger.

When modeling alternatives, it is important to consider not only different sentence formulations but also the morphological type of the trigger. For instance, we want various formulations illustrated in Table 4 to map to the same frame: PRODUCT-DEVELOPMENT, where the `developer` is *Peter* and the `product` is *chatbots*.

For each frame role, a set of syntactic structures is defined that can be mapped to that role. The system searches for these syntactic structures within the corresponding syntactic substructure. For

Table 4
Formulations for different morphological triggers.

Slovak	English	Trigger type
petrov **nn:vývoj** chatbotov	peter's **nn:development** of chatbots	nominal
nn:vývoj chatbotov petrom	**nn:development** of chatbots by peter	nominal
chatboty **amod:vyvíjajúci** peter	chatbots **amod:developing** peter	attributive modifier
peter **vbg:vyvíjajúci** chatboty	peter **vng:developing** chatbots	adjective, active voice
chatboty **vbn:vyvinuté** petrom	chatbots **vbn:developed** by peter	adjective, passive voice
peter **g:vyvíjajúc** chatboty	peter **g:developing** chatbots	gerund
peter chcel **inf:vyvíjať** chatboty	peter wanted to **inf:develop** chatbots	infinitive
peter **vb:vyvíja** chatboty	peter **vb:develops** chatbots	main verb

Table 5
Example of frame materialization definition.

PRODUCT-DEVELOPMENT trigger: develop, build, create		
`role`	scope: main verb	`scope: nominal`
`developer`	`nsubj`	`nmod:poss, pp(nmod,["by"])`
`product`	`obj`	`pp(nmod,["of"])`
`company`	`pp(obl,["at","in"])`	`pp(nmod,["at","in"])`
`beneficiary`	`pp(obl,["for"])`	`pp(nmod,["for"])`
`location`	`pp(obl,["at","in"])`	`pp(nmod,["at","in"])`

example, roles related to the main verb or infinitive are searched for within the entire sentence; roles for attributive adjectives are searched for only within the corresponding local phrase; roles for standalone adjectives are searched for only in their local context, and so on. A simple example of an alternative realisation of the PRODUCT-DEVELOPMENT frame is illustrated in Table 5. The example describes a realisation triggered by either a main verb or a noun (nominal).

For each role, a set of syntactic structures is defined and addressed via links in the dependency graph. These structures may be of various types. For simplicity, we presented a direct relation to a participant through a dependency edge `nsubj`, and an example of a prepositional phrase attachment `pp(obl,["at","in"])` via the `obl` edge and accepted prepositions *"at"*, *"in"*. It is necessary to distinguish between different morphological triggers. For instance, the role `developer` is determined via the `nsubj` edge when triggered by a main verb, but in the case of a nominal trigger, the relevant edges may be `nmod:poss` (*peter's **development***) or `pp(nmod,["by"])` (***development** by peter*).

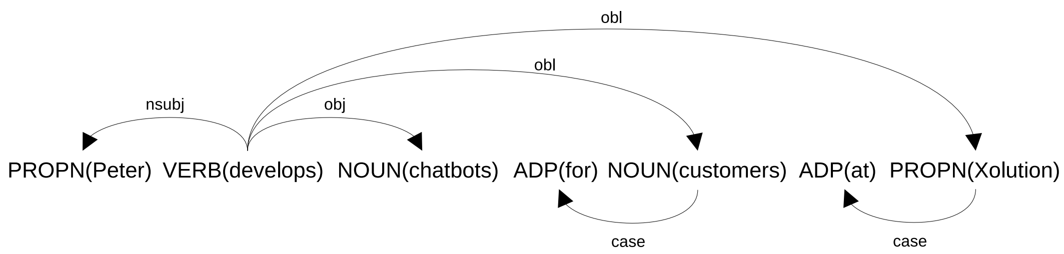

Figure 4: Example of dependency graph.

Let us consider the following example sentence: *Peter develops chatbots for customers at Xolution.* The corresponding dependency graph is shown in Figure 4 and the frame construction is illustrated in Table 6. The phrase *"at Xolution"* was mapped to the role `beneficiary`, but it could also have been mapped to the role `location`. Assuming that the taxonomic class `company` is known for *Xolution*, the resulting mapping is more accurate due to a better alignment with the expected taxonomic classes specified in the frame definition (Table 3).

Table 6
Frame construction example.

PRODUCT-DEVELOPMENT		
`role`	`assignment`	`used mapping`
`developer`	peter	`nsubj`
`product`	chatbots	`obj`
`company`	Xolution	`pp(obl,"at")`
`beneficiary`	customers	`pp(obl,"for")`

5.2.4. Structured Entities

Natural language frequently contains more complex constructions that express rich semantic descriptions of individual participants. Everyday language includes description of entities - physical objects - with an internal semantic structure that may be composed of properties and attributes, represented through nested frames and other entities. For example, we want the system to recognise the phrase *"system for analysis of natural language"* as a single entity and assign it the taxonomic class `nlp-system`. If the entity taxonomy includes a relation stating that `nlp-system` is a subclass of `product`, then the entity *"system for analysis of natural language"* can be correctly identified in the PRODUCT-DEVELOPMENT frame in the role of `product`.

In the described prototype, a mechanism was developed for identifying entities based on their internal structure. An entity is recognised as a specific subgraph within the syntactic structure. The entity recognition system supports rules for transforming graphs by adding auxiliary edges, converting graphs into other graphs, and mapping subgraphs to taxonomic classes.

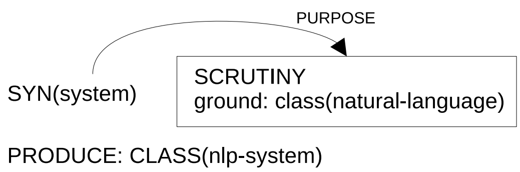

Figure 5: Example of a rule for entity identification.

For example, the graph for recognising the entity *"system for analysis of natural language"* could be defined as a statement like: *"a system whose purpose is to analyse an entity of class* `natural-language`*"*. A sample rule for identifying this entity is illustrated in Figure 5, where `SYN(system)` represents synonyms of the word *system*, and the frame SCRUTINY denotes the situation of analysis, with the role `ground` referring to what is being analysed. If the system detects such a subgraph, it identifies it as an entity and assigns it a corresponding taxonomic class. The dependency graph for this example is shown in Figure 6.

The example of the implementation procedure would proceed as follows. A rule is added to the knowledge base for identifying an entity of the class `natural-language` from the phrase *"natural language"*. A rule for producing a PURPOSE edge is added to the knowledge base. This rule covers all cases where an entity is extended by a prepositional phrase containing a frame introduced by the preposition *"for"*. The rule includes a graph in which the nodes are labeled with variables. Using these variables, an auxiliary edge is inserted. In addition, the rule specifies how to generate a standalone frame from the auxiliary edge during the final completion phase. This rule is fully general and covers, for example, *"system for analysis"*, *"ball for kicking"*, *"tool for writing"*, and similar expressions. Example rules are illustrated in Figure 7.

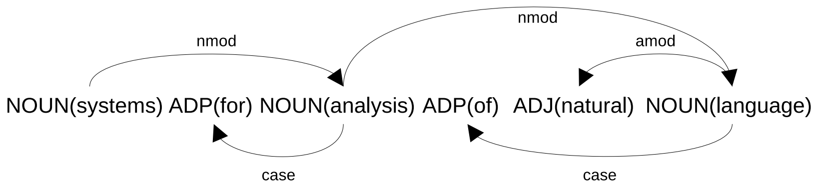

Figure 6: Example of an entity dependency graph.

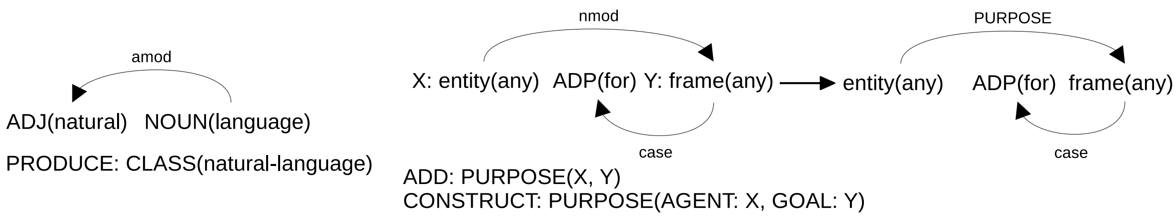

Figure 7: Examples of graph transformation rules.

Assuming that the knowledge base contains a description for the materialisation of the frame SCRUTINY triggered by a nominal *analysis* mapping role `ground` via prepositional phrase `pp(nmod,"of")`, the system will, through the successive application of rules, eventually achieve the final representation illustrated in Figure 8. Using the source rule (in Figure 5), this subgraph is identified as an entity with the taxonomic class `nlp-system`.

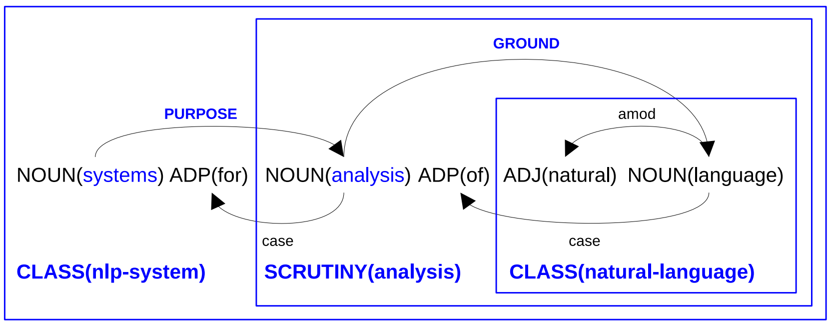

Figure 8: Example of transformation result.

The resolver implemented in the prototype operates as a production system combining frames and entities. The result is a recursive semantic structure where frames contain entities and entities contain frames. During production, the resolver combines all possible cases of identified frames and applied transformation rules. The final output is a puzzle assembled from individual frames and entities. Both complete and partial results are evaluated and pruned on the fly using a variety of scoring heuristics. The score takes into account factors such as the overall completeness of the representation (how many words were or were not used in the structure), taxonomic accuracy, valency and many more. As a result, the system can generate multiple different combinations with the same score.

When combining frames and entities, a duality can occur where the same text is interpreted both as a frame and as an entity. For example, the phrase *"otcova **cesta** z mesta do práce"* / *"father's **road/journey** from the city to work"* can be interpreted as the frame MOTION, where father travels from one place to another, but at the same time as an entity representing a physical object - the **road** owned by the father, with known start and end points. Both interpretations are correct. This duality can be clarified by adding further context. For instance, in the sentence *"Otcova **cesta** z mesta do práce **trvala dlho**"* / *"Father's **road/journey** from the city to work **took a long time**"*, the main frame TAKING-TIME is identified, which describes an event unfolding over time. In this case, *road/journey* is unambiguously identified as the frame MOTION, and the interpretation using the entity is discarded.

5.2.5. Inference

The prototype implements two types of explicit knowledge inference:

- Generalisation - inheritance within the frame taxonomy, where parent frames are materialised. For example, the frame ACTIVITY can be inferred from the subordinated frame MOTION.
- So-called entailments, where frames that logically follow from other frames are materialised. For instance, from the frame COMMERCIAL-BUY (someone bought something), it is possible to infer the frame COMMERCIAL-SELL (someone sold something), and if the purchased item and its price are mentioned, the frame EXPENSIVENESS (how much something costs) can also be inferred.

In both cases, inference is realised as a simple mapping of roles between frames. All known mappings for the materialised frame are applied through activity spreading. Thus, inference is performed recursively for the given frame as well as for each derived frame. This way, all generalisations and entailments are straightforwardly inferred.

For mapping, it is possible to specify the conditions under which it can be performed. For example, the mapping to the frame EXPENSIVENESS is carried out only if both the product and the price are known.

5.3. Evaluation

Since this is an experimental prototype working with declarative knowledge, and as we do not have any statistical data available, it is not possible to quantitatively evaluate the system's performance at this stage. Furthermore, we are still experimenting with multiple aspects of knowledge representation and modeling patterns, so the prototype is not yet considered complete.

For experimental evaluation, a question-answering system was developed. It allows inputting facts (textual inputs) that are converted into frame-based representations. Subsequently, questions can be posed that take into account all aspects of knowledge inference. We currently use this system for both tuning the prototype and for empirically assessing the expressive power of the chosen semantic representation.

6. Conclusion

FrameNet builds on the hypothesis that people interpret the conceptual meaning of words against semantic frames, which are understood as schematised situations, templates, scenarios, or generalised patterns arising from repeating similar types of events in the real world. Scenes-and-Frames semantics has a significant influence on Natural Language Processing and Artificial Intelligence, as it can help machines to generate more coherent and contextually appropriate responses and address complex language understanding problems. The ability to work with text meaning is a prerequisite for those machines to answer questions without the need for large datasets.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] M. Grepl, P. Karlík, Skladba současné češtiny, Votobia, Olomouc, 1998.

[2] J. Ruppenhofer, M. Ellsworth, M. R. L. Petruck, C. R. Johnson, J. Scheffczyk, FrameNet II: Extended Theory and Practice, Technical Report, International Computer Science Institute, Berkeley, California, 2006. URL: http://www.icsi.berkeley.edu/framenet.
[3] C. J. Fillmore, The case for case reopened, in: P. Cole (Ed.), Syntax and Semantics 8: Grammatical Relations, Academic Press, New York, 1977, pp. 59–81.
[4] C. J. Fillmore, C. F. Baker, A frames approach to semantic analysis, in: B. Heine, H. Narrog (Eds.), The Oxford Handbook of Linguistic Analysis, 2nd ed., Oxford University Press, Oxford, 2015, pp. 791–816.
[5] J. Ruppenhofer, H. C. Boas, C. F. Baker, FrameNet, in: P. A. Fuertes-Olivera (Ed.), Routledge Handbook of Lexicography, Routledge, 2017, pp. 383–392.
[6] D. Gildea, D. Jurafsky, Automatic labeling of semantic roles, Computational Linguistics 28 (2002) 245–288.
[7] L. Tesnière, Elements of Structural Syntax, John Benjamins, Amsterdam, Philadelphia, 2015.
[8] C. J. Fillmore, Valency issues in FrameNet, in: T. Herbst, K. Götz-Votteler (Eds.), Valency: Theoretical, Descriptive and Cognitive Issues, Mouton de Gruyter, Berlin/New York, 2007, pp. 129–160.
[9] M. Ivanová, M. Sokolová, M. Kyseľová, V. Perovská, Valenčný slovník slovenských slovies na korpusovom základe, https://www.ff.unipo.sk/sloval/Slovnik/slovnik.html, 2014. Published by Filozofická fakulta Prešovskej univerzity v Prešove.
[10] M. Lopatková, V. Kettnerová, E. Bejček, A. Vernerová, Z. Žabokrtský, Valenční slovník českých sloves – VALLEX, druhé, přepracované a rozšířené ed., Univerzita Karlova, nakladatelství Karolinum, Praha, 2016.
[11] Google, Dialogflow, https://cloud.google.com/dialogflow, 2024.
[12] Meta, Wit.ai, https://wit.ai, 2024.
[13] T. Bocklisch, J. Faulker, N. Pawlowski, A. Nichol, Rasa: Open source language understanding and dialogue management, arXiv preprint arXiv:1712.05181 (2017). URL: https://arxiv.org/abs/1712.05181.
[14] Industrial-strength natural language processing in Python, https://spacy.io/, 2026.
[15] K. Kipper-Schuler, VerbNet: A broad-coverage, comprehensive verb lexicon, Ph.D. thesis, University of Pennsylvania, 2005.
[16] M. Palmer, D. Gildea, P. Kingsbury, The proposition bank: An annotated corpus of semantic roles, Computational linguistics 31 (2005).
[17] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, N. Schneider, Abstract meaning representation for sembanking, in: Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, Association for Computational Linguistics, Sofia, Bulgaria, 2013.
[18] J. E. L. Van Gysel, M. Vigus, J. Chun, K. Lai, S. Moeller, J. Yao, T. O'Gorman, A. Cowell, W. Croft, C.-R. Huang, J. Hajič, J. H. Martin, S. Oepen, M. Palmer, J. Pustejovsky, R. Vallejos-Yopán, N. Xue, Designing a uniform meaning representation for natural language processing, KI - Künstliche Intelligenz 35 (2021). doi:10.1007/s13218-021-00722-w.
[19] E. Páleš, SAPFO. Parafrázovač slovenčiny. Počítačový nástroj na modelovanie v jazykovede, Veda, Bratislava, 1994.
[20] M. C. de Marneffe, C. D. Manning, J. Nivre, D. Zeman, Universal dependencies, Computational Linguistics 47 (2021) 255–308.