# Semantic Parser for Sentiment Analysis and the Emotional Computer Agents

Artemy Kotov, Anna Zinina
Kurchatov Institute
Moscow, Russia
{kotov_aa, zinina_aa}@nrcki.ru

Alexander Filatov
Samsung R&D Institute Rus
Moscow, Russia
filatov.alex@gmail.com

***Abstract*—We design a semantic parser for natural language texts, which constructs syntactic trees and a semantic representation – "frame": a set of valencies with semantic markers in each. This representation is used to recognize the description of emotional situations in texts by an emotional computer agent.**

## I. Introduction

Linguistic theories usually distinguish syntactic and semantic levels of representation [1]. It is theoretically considered that semantic representation is language independent and should be used for machine translation and other applied tasks, requiring "text comprehension". At the same time, most of the existing "semantic parsers" rely on syntactic structure as the deepest representation level. There are numerous attempts to transfer parts of semantic knowledge to the syntactic trees. In [2] syntactic analysis is combined with semantic role labeling. In [3] syntactic representation is enriched with lexical functions. In [4] tree nodes are associated with entities from an extended ontology. The extended usage of lexical functions with the annotation of word classes is used to construct semantic representation in [5]. We suggest a rule-based parser, which, in addition to syntactic dependency tree, constructs a standalone semantic representation, suitable for further processing. In particular, this representation is used in architecture of an emotional computer agent to construct emotional reactions to an incoming text, as described further in III.

## II. Parser Architecture

Parser is designed to implement the main levels of language representation: it includes morphology processor (stemmer), syntax dependency parser and the constructor of semantic representations (frames). Parser is written on C#, uses dictionary data, stored in SQL database, and grammar rules in XML format for syntax processing. It has a rule-based architecture, and in the present state does not rely on any machine learning approach.

### *A. Stemmer and dictionary*

Stemmer is based on the OpenCorpora resource dictionary [6]. For our dictionary we have selected only 20,000 most frequent lemmas (frequencies obtained from [7]). To extend the dictionary it is possible to attach a list of additional lemmas with the indication of inflectional class (according to [8]). The parser database stores inflections for Russian inflectional classes, allowing stemmer to recognize wordforms for lemmas from the additional dictionary.

The dictionary contains basic semantic information for the most frequent words. To annotate lexical semantics we use a set of 596 semantic markers, initially based on [9], but greatly extended to describe different lexical fields, distinguish lemmas inside the same lexical field and describe polysemy for each lemma. 14,000 of lemmas are annotated with semantic markers, there are 27,000 markers total in lexical semantics, 1.5 markers on average per lemma meaning. Unlike [4], where a word is considered as a representation of a specific semantic class from a universal ontology, we suggest, that a word carries semantic markers from different classes: *bank* has the markers for 'organization', 'building' and 'abstract container'. This description of polysemy allows us to simulate an important semantic observation, that a representation of a concept and the set of its focal markers varies from situation to situation [10], so different reference frames may address different focal markers in the semantics of a word.

### *B. Parser and grammar*

*1) General parser architecture.* We develop a dependency parser implementing left-to-right approach. On each shift the parser adds the next token (wordform) to stack(s) and then applies to each stack all the feasible reductions, as defined by the grammar rules. A rule is defined as a possible reduction, where the right-hand side of the rule can be reduced to the left-hand side head $\boldsymbol{h}$:

$$\boldsymbol{h} \rightarrow \langle a, b, \ldots\ n \rangle$$

In a grammar rule $\boldsymbol{h}$ can be defined as a non-terminal symbol, which does not appear as a standalone token in the text. This approach allows a developer to describe immediate constituent grammars with non-terminals. In this case $\boldsymbol{h}$ is considered as a "virtual head" of the rule. At the same time, it is possible to mark one of the right-hand segments as the rule head, so that the rule reduces the top of the stack to its terminal head:

$$\boldsymbol{h} \rightarrow \langle a, b, \boldsymbol{h}, \ldots\ n \rangle$$

ISBN 978-952-68397-0-7 (paperback), ISBN 978-952-68397-1-4 (PDF)

or, in a more compact form:

$$<a, b, \boldsymbol{h}^{\text{head}}, \dots n>$$

This approach allows a developer to design pure dependency grammar, where terminal segments (like ***h***) substitute other terminal segments (*a, b,… n*) within a syntactic tree.

Right-hand side of a rule is flexible and may have from 1 to any number of segments. It is also possible to define optional segments within a rule. For example, the following rule $\boldsymbol{R}_1$ joins adjective pronoun *APro*, adjective *Adj* (where *APro* and *Adj* are optional) and a noun *N* within a noun phrase, where *N* is the head:

$$[APro], [Adj], N^{\text{head}}$$

This rule may apply to the following sequences:

- *N*
- *Adj, N*
- *APro, Adj, N*
- *APro, N*

Within a specific rule a developer may define the following procedures for each segment:

- check for a specific marker (grammeme or a syntactic feature);
- check for agreement for a specific grammatical category with other segments of the rule;
- set a specific grammeme (for the head only);
- copy grammeme of a specific type to rule head;
- check for a specific lemma.

The following rule binds subject to a finite verb in a reversed word order (*walked man*). It checks for a finite verb (`VFIN`), for the possibility of the verb to link substantive in nominative case (`0-Snom-ag`), checks for agreement in number, animation, person and gender, and accounts the binding: changes `0-Snom-ag` to `1-Snom-ag`. For a subordinate substantive it transfers its semantics to **ag** (**agens**) semantic valency, and – if the subject is a question word (having type `Ques`) – copies the value of this type to the head verb.

```
<rule name="VFIN-subj-r" fork="true">
  <seg head="true">
    <check marker="VFIN"/>
    <check marker="0-Snom-ag"/>
    <agr type="NMbr"/>
    <agr type="Anim"/>
    <agr type="PErs"/>
    <agr type="GNdr"/>
    <set type="Snom-ag" marker="1-Snom-ag"/>
  </seg>
  <seg semval="ag">
    <check marker="S"/>
    <check marker="nomn"/>
    <agr type="NMbr"/>
    <agr type="Anim"/>
    <agr type="PErs"/>
    <agr type="GNdr"/>
    <copyup type="Ques"/>
  </seg>
</rule>
```

The ability to rewrite certain markers (grammemes or syntactic features) of the rule head ***h*** may be used to indicate the subordinate segments of ***h*** or to limit repeated applications of the same rule. For example, rule `VFIN-subj-r` uses `0/1-Snom-ag` marker to provide one single binding with the subject noun. This approach allows a developer to concentrate on the list of specific grammemes and syntactic markers for a given segment *N*, rather than to manipulate with abstract non-terminal symbols.

The grammar developer can also manually control, if syntactic analysis forks (duplicates the stack) upon an application of a specific rule. For example, rule `VFIN-subj-r` (`VFIN + Substantive/Subject`) allows forks, cause Substantive may be the subject of the next verb. At the same time, rule `Substantive/Subject + VFIN` does not allow forks: substantive in nominative is attached to the verb with no option. In such cases the developer may mark rule as “no fork”, reducing the memory required for processing.

In order to reduce memory usage the parser also checks the structure of the stack before each shift according to special optimization rules. Stacks that contain non-reduced minor segments are excluded from further analysis. For example, if we shift from a finite verb, we may consider, that all nouns on the left should be reduced to this or other verbs, so we may eliminate stacks with remained nouns.

*2) Russian grammar.* This parser architecture allows us to design a flexible grammar for the Russian language. We follow the principle, that the head of a syntactic group should reflect the grammatical (and semantic) features of the whole group. So, for example, we consider prepositions as a noun category and reduce prepositions to nouns – this also forms direct link between a verb and its actant (without intermediary preposition), which is favorable for future semantic analysis. Although we generally use terminal heads (as usual for the dependency grammar), we use virtual heads (non-terminals) for conjunction noun phrases – as we consider, that none of the segments (neither conjunction itself, nor any of the nouns) represent grammatical or semantic characteristics of the whole phrase.

Our present Russian grammar contains 289 rules. In our studies we mainly concentrated on the recognition of explicitly expressed emotional patterns (d-scripts), rather than on recall of syntactic analysis: parser runs on text collections with the average speed of 475 sentences per minute with 20% of successfully parsed sentences (in 47% of cases failures appear due to the incomplete dictionary, and in 33% – do to the incomplete or inaccurate grammar).

### *C. Semantic representations constructor*

The structure of semantic representation is one of the major problems of theoretical linguistics. We consider semantic representation of a simple sentence (clause) as a set of semantic roles (valencies), similar to the roles by Fillmore [11] (and extended in [12], [13]) with the predicate ***p*** at the head and with actants as subordinate tree nodes. We use 21 valency types, in particular: **p, agens, patient, content, instrument, source-point, target-point, cause, effect** etc.

Semantic representations are constructed with the help of syntactic rules. Each syntactic rule may indicate, that one of its segments should be mapped to a specific valency, e. g. when a noun in nominative case is bound with a verb, the binding rule maps the noun to **agens**. Each entity of the sentence (verb and actants) is represented by a set of semantic markers for the corresponding lemma and all its subordinate lemmas: meanings of adjectives and subordinate noun phrases are combined with the head noun, meanings of adverbs are combined with the head verb. As the semantic representation for a simple sentence is a one level tree, it can be drawn as a table of valencies ("frame") with a set of semantic markers for each valency (**p** is always the head, although it is not indicated at the table). Phrase *a brown cat drinks milk* is described by the following semantic representation:

TABLE I. SEMANTIC REPRESENTATION FOR 'A BROWN CAT DRINKS MILK'

| **p** | **agens** | **patient** |
|---|---|---|
| ingest<br>drink | object<br>living-being<br>has-color | object<br>food |

This representation can be further used for analysis and classification: for facts extraction ('living beings ingest food') or emotions modelling by the affective computer agents ('someone eats my food!').

## III. ARCHITECTURE OF THE COMPUTER AGENT

### *A. Emotional reactions*

As the parser extracts semantic frames from text, it can be used in a variety of applications, covering data mining and fact extraction. In particular, the parser is used inside an emotional computer agent to make it sensitive to the emotional information, contained in incoming texts. The agent is designed as a simple cartoon character (Fig. 1), which receives semantic representations, generates simple gestures and utterances and may address different counterparts, including itself [14], [15].

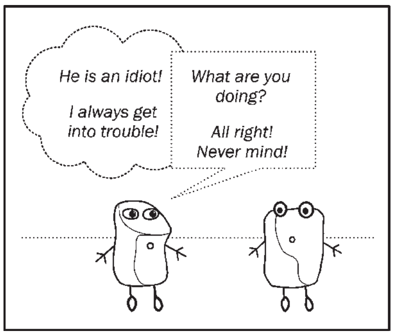

Fig. 1. Cartoon situation for a computer agent

We rely on the theoretical architecture of emotional computer agent, suggested by Sloman [16]. In this architecture an incoming event (or incoming text semantics) can be handled on three levels of processing: reactive (responsible for basic emotional processing), deliberative (responsible for rational reasoning) and meta-cognitive (responsible for higher forms of reasoning, like introspection and reasoning about own plans). We implement the opposition between reactive and deliberative processing with the help of two groups of scripts: d-scripts (dominant scripts) and r-scripts (rational scripts). These scripts compete for the processing of each incoming semantic representation and form different output – gestures and output utterances.

The list of d-scripts, responsible for the emotional processing, is based on [17]. It includes 34 units, 13 of which are responsible for the recognition of negative situations: DANGER, APPROPR ("Appropriation"), SUBJV ("Subjectivity" – e. g. 'all he thinks about is himself!') etc., while 21 d-scripts are responsible for positive situations: CONTROL, CARE, COMFORT, ATTENTION (e. g. 'they all adore you!'), APPROVAL (e. g. 'you did it like a real man!') etc.

### *B. Script activation*

Each script has a reference semantic representation. Each incoming frame is processed by a number of competing scripts. Scripts are activated to different degrees depending on the proximity to the frame. Proximity metric evaluates the number of frame's markers matched on a per-valency basis to script's reference frame. Scripts activation is also affected by agent's "temperament" (sensitivity of different scripts) and "mood" (current activation of scripts).

For a sample phrase *I'll hit you!* – the parser constructs the following representation:

TABLE II. SEMANTIC REPRESENTATION FOR 'I'LL HIT YOU!'

| **p** | **agens** | **patient** |
|---|---|---|
| touch<br>with-force | object<br>somebody<br>principal<br>other | object<br>somebody<br>self |

This representation is compared to all the scripts and causes the highest activation to the script DANGER with the following reference frame:

TABLE III. REFERENCE SEMANTIC FOR DANGER D-SCRIPT

| **p** | **agens** | **patient** |
|---|---|---|
| touch<br>with-force | somebody<br>other | somebody<br>self |

This forces the agent to prefer an aggressive or flee reaction in speech and gestures.

Here the marker 'principal' indicates the author of the incoming text (he refers to himself as "*I*") and 'self' corresponds to the agent (it is referred to as "*you*" in the incoming phrase). If no evident pronouns appear in the text, the agent may identify itself with a valency, corresponding to the prevailing emotion (d-script), e. g. if 'teachers beat students', the agent associates itself with a 'student' if it is afraid (DANGER d-script) and with 'teacher', if happy to scare others (WE●DANGER d-script). Both reactions may appear during processing, representing possible ambiguity in emotional situations, where a listener may partly associate himself with an aggressor and the victim.

In a more sophisticated case the parser should react on texts of the internet blogs. In particular, we observe the following picture for the utterance: *A real man is always interested in the life of the beloved girl*. Parser creates the following representation:

TABLE IV. SEMANTIC REPRESENTATION FOR 'A REAL MAN IS ALWAYS INTERESTED IN THE LIFE OF THE BELOVED GIRL'

| p | agens | patient |
|---|---|---|
| think<br>pay-attention<br>frequently | object<br>somebody<br>man<br>positive | abstract<br>time-period<br>existence<br>object<br>somebody<br>woman<br>of-minimal-age<br>positive |

Presently the semantic markers of the noun phrase join into the same valency. For *life of the beloved girl* – markers of *life* ('abstract', 'time-period', 'existence'), *beloved* ('positive') and girl ('object', 'somebody', 'woman', 'of-minimal-age') are combined into **patient**. This semantic representation activates the following d-scripts:

- PLAN: Somebody plans something frightening against me – 'man makes some evil plans against woman'.
- SUBJV ("Subjectivity"): Somebody is narrow-minded, thinks only about one thing – 'all men think about are women'.
- ATTENTION: Subject is pleased, because somebody pays an attention to him – 'woman is happy because of the men's attention'.
- APPROVAL: Somebody acts like a hero, does something right – 'real men do it right to pay attention'.

Although APPROVAL (with subordinate usage of ATTENTION) is the correct option suggested by the text, other – negative scripts, although their activation may be considered as false-positive, may give another emotional view to the situation: (a) they suggest a possible negative reaction in an ambiguous emotional situation, where attention may seem both pleasant and frightening, and (b) they offer a possible ironical answer for the agent, as designed in [18]. Option (a) can be chosen for a "depressive" agent and (b) by an ironical agent. These alternative emotional reactions may enrich the expressive behavior of a computer agent and bring its behavior closer to natural emotional dynamics.

## IV. CONCLUSION

Semantic parser can be designed in a way to construct a standalone semantic representation, suitable for further processing. This approach engages theoretical linguistic concepts: (a) explicit usage of semantic markers and valencies within semantic representation, and (b) separation of syntactic and semantic representation levels of language (c) changes of semantic representation of a concept, depending on the situation (frame). Semantic representation can also serve further semantic procedures, like recognition of emotional situations in incoming texts.

## ACKNOWLEDGMENT

The research is partially supported by RFBR grant 15-29-01173.

## REFERENCES

[1] I.A. Mel'čhuk, *The experience of theory of linguistic models "Meaning ⇔ Text"*. Moscow: The school "Languages of Russian Culture", 1999 (in Russian).
[2] A. Björkelund, B. Bohnet, L. Hafdell, and P. Nugues, "A High-Performance Syntactic and Semantic Dependency Parser", *in. Proc. Coling 2010: Demonstration Volume*, Aug. 2010, pp. 33–36.
[3] L.L. Iomdin, "Representing Semantics in the Digital Combinatorial Dictionaries of the ETAP-3 System: New Developments", *in. Proc. MONDILEX Fourth Open Workshop*, June–July 2009, pp. 69-75.
[4] K.V. Anisimovich, K.Ju. Druzhkin, F.R. Minlos, M.A. Petrova, V.P. Selegey, and K.A. Zuev, "Syntactic and semantic parser based on ABBYY Compreno linguistic technologies", *Computational Linguistics and Intellectual Technologies*, vol.11(18), 2012, pp. 810-822.
[5] Tuzov V.A. *Computer semantics of Russian language*, St.-Petersburg, 2003. (in Russian)
[6] V.V. Bocharov, S.V. Alexeeva, D.V. Granovsky, E.V. Protopopova, M.E. Stepanova and A.V. Surikov, "Crowdsourcing morphological annotation", *Computational Linguistics and Intellectual Technologies*, vol.12(19), 2013, pp. 109-114.
[7] O.N. Lyashevskaya, and S.A. Sharov, *Frequency Dictionary of Modern Russian Language (on materials of the Russian National Corpus)*. Moscow: Azbukovnik, 2009 (in Russian).
[8] A.A. Zaliznyak, *Grammatical dictionary of Russian language*. Moscow: Russian language, 1980 (in Russian).
[9] A. Wierzbicka, *Lingua Mentalis: The semantics of natural language*. New York: Academic Press, 1980.
[10] W. Yeh, and L.W. Barsalou. The situated nature of concepts. American Journal of Psychology, vol. 119, 2006, pp. 349-384.
[11] C.J. Fillmore, "The Case for Case". *In Bach and Harms (Eds.): Universals in Linguistic Theory*, 1968, pp. 1-88.
[12] E.V. Kashkin, and O.N. Lyashevskaya, "Semantic roles and construction net in Russian FrameBank", *Computational Linguistics and Intellectual Technologies*, vol.12(19), 2013, pp. 325-343 (in Russian).
[13] O.N. Lyashevskaya, "Bank of Russian constructions and valencies", *in Proc. LREC'2010*, May 2010, pp. 1802-1805.
[14] A.A. Kotov, "Simulating Dynamic Speech Behaviour for Virtual Agents in Emotional Situations". *Affective Computing and Intelligent Interaction*, vol. 4738, 2007, pp. 714-715.
[15] A.A. Kotov, "Application of d-script model to emotional dialogue simulation". *Affective Dialogue Systems,* LNCS, vol. 3068, 2004, pp. 193-196.
[16] A. Sloman, and R. Chrisley, "Virtual Machines and Consciousness". *Journal of Consciousness Studies*, vol. 10(4-5), 2003, pp. 133-172.
[17] A.A. Kotov, *Mechanisms of speech influence in publicistic mass media texts*. Ph.D. thesis. Moscow: RSUH, 2003 (in Russian).
[18] A.A. Kotov, "Accounting for irony and emotional oscillation in computer architectures". *in. Proc. of International Conference on Affective Computing and Intelligent Interaction ACII 2009*, Sept. 2009, pp. 511.