ENIAM: Categorial Syntactic-Semantic Parser for Polish

Wojciech Jaworski **Jakub Kozakoszczak**
Institute of Computer Science, Polish Academy of Sciences
University of Warsaw
wjaworski@mimuw.edu.pl jkozakoszczak@gmail.com

Abstract

This paper presents ENIAM, the first syntactic and semantic parser that generates *semantic representations* for sentences in Polish. The parser processes non-annotated data and performs tokenization, lemmatization, dependency recognition, word sense annotation, thematic role annotation, partial disambiguation and computes the semantic representation.

1 Introduction

ENIAM is a syntactic and semantic parser that generates *semantic representations* for sentences in Polish. It is publicly available under the address http://eniam.nlp.ipipan.waw.pl and licensed under GPL 3. The parser processes non-annotated data and performs all the necessary steps: tokenization, lemmatization, dependency recognition, word sense annotation, thematic role annotation, partial disambiguation and computes the semantic representation. It is the first semantic parser for Polish.

The system was developed within the CLARIN-PL project (http://clarin-pl.eu) which aims at creating a research infrastructure intended for the humanities and social sciences dealing with large collections of Polish texts. The range of applications of the system is wide and includes all the language processing tasks that involve the semantic level, in particular Information Retrieval, Question Answering, Recognizing Textual Entailment. An example of a QA task is the processing of biograms for knowledge extraction and for answering questions of the type "Who publishes in a journal edited by themselves?".

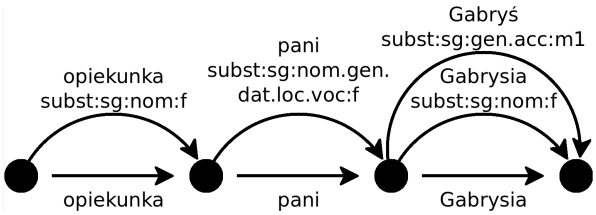
2 System Description

The text processing steps are executed in a fuzzy pipeline. The system doesn't disambiguate the output of every step immediately but makes a compact representation of the ambiguous outcome and passes it to next stages. Structure sharing assures that the growth of the representation is polynomial with respect to the length of the sentence, although the number of interpretations growths exponentially. The disambiguation is perform near the and of the pipeline, when all syntactic and semantic constraints are applied to data.

The reason for the fuzzy pipeline approach is that disambiguation is never 100% accurate and a single error during disambiguation after one stage of processing often makes it impossible to perform the next stage of processing. The other reason is that we consider ambiguity as a property of natural language that should be modeled under equal terms with other linguistic phenomena.

During the preprocessing (all stages before finding the dependency structure) the text is represented as a graph whose edges correspond to running words (tokens). For each interpretation of a token another alternative edge is being added, e.g.:

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations, pages 243–247, Osaka, Japan, December 11-17 2016.



Preprocessing includes character-level analysis (tokenization, haplology of punctuation), word-level analysis (lematization, morphosyntactic tagging, word sense annotation, subcategorization, semantic valence annotation) and phrase-level analysis (abbreviation extension, MWE recognition, Named Entity Recognition).

Lemmatization is based on the dictionaries SGJP (Zygmunt et al., 2015) and Polimorf (Wolinski et al., 2012). All possible lemmata and morphosyntactic tags are deduced from the endings. Then, if there is at least one known lemma among deduced lemmata, the unknown lemmata are being discarded. If none of the lemmata is known, all of them are returned.

Named entity annotation is also done with the help of SGJP and Polimorf which associate proper names with a general type of their referent, e.g. toponym, surname and such. Capitalized nouns that are not classified as proper names in the dictionaries are treated as proper names of unknown type.

Word senses taken from Słowosieć (the Polish WordNet) (Maziarz et al., 2014) are ascribed to lexemes. Each sense is represented in the WordNet style as a lemma with a number . The senses of proper names are their types.

The valency of the lexemes in the sentence is determined with the valency dictionary Walenty (Przepiórkowski et al., 2014). Walenty covers most verbs and many nouns, adjectives and adverbs. Each entry comprises syntactic schemata that include detailed syntactic description of the obligatory dependents, and semantic frames that give their semantic characteristic, namely thematic roles and selectional preferences expressed as synsets from Słowosieć. The schemata and the frames are mapped many-to-many.

Preliminary disambiguation tries to crudely match selectional preferences of all the possible heads with hypernyms of all the possible dependents. If a selectional preference is not a hypernym of any lexeme in the sentence, it is discarded, and if a hypernym is not a preference of any of the lexemes, it is discarded as well. The slots in the syntactic schemata in Walenty correspond to traditional argument positions (such as subject or object). The system adds slots for remaining dependents if needed (e.g. locative modifiers)

Syntactic Analysis Preprocessing is followed by the construction of the syntactic tree done in Type Logical Categorial Grammar (TLCG) (Morrill, 2010). To parse a sentence is to give a proof in a non-commutative intuitionistic linear logic, therefore fragments of the proof system can be implemented to obtain quick parsers with guaranteed correctness. The lexicon is generated dynamically for each query from SGJP and Walenty entries.

The categorial grammar doesn't play further role in building the semantic representation. The full lexicalization of categorial grammars facilitates integration with the resources: morphological and valence dictionaries. The choice of TLCG instead of the standard CCG (Bos et al., 2004) is motivated by the wider choice of connectives that express important information, like ambiguity or polymorphism.

Parser The parser is based on the CKY algorithm and is a direct implementation of a fragment of the proof system for linear logic. It has the expressiveness of context-free grammar. Since the categorial framework allows for inflectional ambiguity representation, the size of the generated lexicon is exponentially smaller than a context-free grammar that models the same language. The dependency structure between tokens is generated in a lazy way. The ambiguity is expressed in the form of a compressed tree. Fragments of the parse tree are compressed immediately after obtaining.

Semantic valence In order to better handle the ambiguities caused by the variety of senses and thematic roles that are assigned to lexemes, no semantic information is introduced to the categorial grammar and thus no semantic representation is immediately obtained together with the dependency structure. The

alternating meanings and valency frames are added in the maximally local way. Thus this approach leads towards Universal Dependencies (De Marneffe and Manning, 2008), where non-semantic prepositions, numerals, auxiliary verbs etc. becomes dependents of their traditional arguments.

Disambiguation Disambiguation is done in stages. First, we check the satisfiability of selectional restrictions, then we select the most likely lemmas on the basis of a list of NKJP1M (Adam Przepiórkowski and Lewandowska-Tomaszczyk, 2012) lemma frequency list, and at the end, we choose word senses. Other types of ambiguity, such as eg. PP attachment ambiguity remains currently ambiguous. For the purpose of the presentation in the demo, 10 unambiguous dependency structures are drawn.

Semantics Semantic representation is built upon dependency trees augmented with concepts from Słowosieć (which is an ontology for our meaning representation) and relations between concepts that extend the set of thematic roles defined in Walenty.

The semantic analysis is shallow in that it describes the world in accordance to its view imprinted in the language:

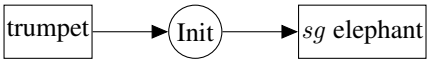
- The intensions of lexemes are concepts.
- The concepts provide truth conditions for the referents of the words.
- Relations between the referents are provided by valency dictionary or given in the syntactic relations.

Its main merit is that it requires little resources for a complete semantic representation.

However, the presented shallow semantic analysis is a starting point for further development, mainly through enrichment with domain-specific theories.

Metalanguage We assume an extended version of FOL with one meta-predicate DSCR that binds a formula with its identifier. The logical formulae are presented on the form of semantic graphs.

(1) *Słoń* *trąbi.*
Elephant trumpets.
'An elephant trumpets.'



The boxes represent entities mentioned in the text. One is the *elephant* and another is the action of *trumpeting*. [1] The symbol sg is a quantifier that define the count of the entities exactly one.

The circles represent relations between the entities. The Init relation says that the *elephant* is the initiator of *trumpeting*. The ingoing arrow leads from the first argument and the outgoing one leads to the second.

The graph is equivalent to the logical formula

$$\exists(s, \text{TYPE}(s, \text{elephant}) \wedge |s| = 1, \exists(t, \text{TYPE}(t, \text{trumpet}) \wedge \text{INITIATOR}(t, s))) \quad (2)$$

where each entity is identified by a variable. The predicate $\text{TYPE}(x, \text{t})$ assigns a type t to a variable x, i.e. it states that set of objects denoted by x belongs to the ontological category t.

Variables are always assigned to sets of entities. Singular number is denoted by the statement that a set is a singleton, and plural number by the statement that it has quantity greater than 1.

Proper names are represented using predicate $\text{HASNAME}(x, '\text{name}')$ which connects a string 'name' with a set of objects denoted by x. In semantic graphs quotation marks indicate the property of being a proper name. Those names don't define an ontological type of the referent but identify it by assigning a label.

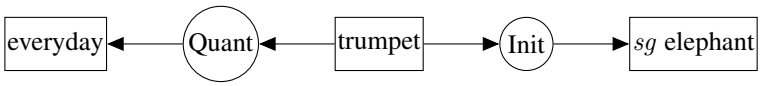
[1] For the convenience of non Polish speaking readers all presented logical formulae are translated into English and are therefore not identical to the parser output. In particular the parser presents concept names in Polish.

We consequently reify all concepts. Every lexeme (or MWE) which is not a quantifier, conjunct or non-semantic item is translated into a TYPE or HASNAME predicate. The reasons for this are: the possibility of modifiers for virtually every part of speech in Polish, uniformization of all parts of speech for the clarity of the Semantic Metalanguage and for further processing.

Apart from the above two predicates we have a fixed number of binary predicates that denote relations between concepts such as Init (Initiator), Thme (Theme), etc.

We also extend FOL with special quantifiers existing in the language, e.g. *co dziesiąty* ('every tenth') or *prawie każdy* ('almost every').

(3) *Słoń* *codziennie* *trąbi.*
Elephant everyday trumpets.
'An elephant trumpets everyday.'

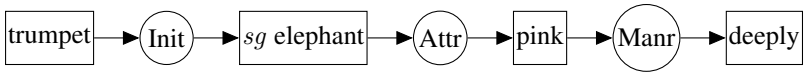


$$\exists(s, \text{TYPE}(s, \text{elephant}) \wedge |s| = 1, \text{EVERYDAY}(t, \text{TYPE}(t, \text{trumpet}) \wedge \text{INITIATOR}(t, s))) \quad (4)$$

Quantifiers are ordered according to words in the sentence. This solution to the problem of quantifier scope ambiguity is motivated by the fact that Polish is nearly free word order language.

Properties Properties are typically expressed by adjectives and adverbs.

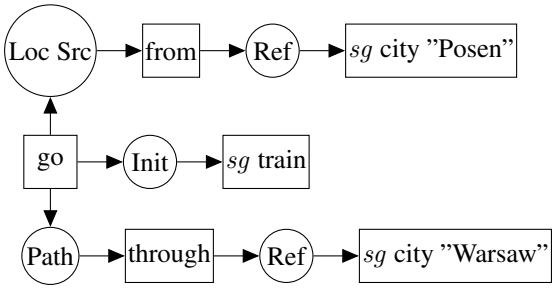
(5) *Intensywnie* *różowy* *słoń* *trąbi.*
Deeply pink elephant trumpets.
'An elephant in deep pink trumpets.'



The name of the property is bound with the name of the entity that has the property with the predicate Attribute for adjectives and Manner for adverbs. Individuals have properties and properties also have properties.

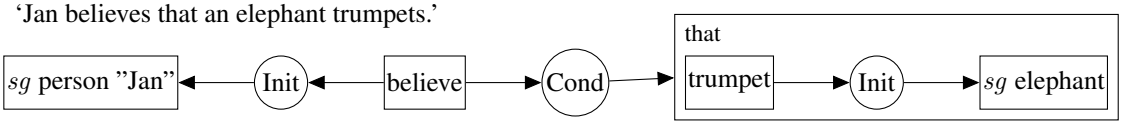
Time and space descriptions Time and space is usually given in adverbs and prepositional phrases. Prepositions of location and direction give the relations between locations and temporal prepositions give the relations between points and or intervals in time taken separately or in sets. Those relations undergo reification because they can be modified, e.g. *dość głęboko w szafie* 'quite deep in the closet'. The predicate Location indicates the location of a situation / an event. The predicates Location Source, Location Goal, Path indicate the presence of movement, its location and direction. The predicates Time and Duration give information about the temporal location of an entity (typically an event) and about its duration. The Ref predicate binds a proposition with its dependent.

(6) *Z* *Poznania* *jedzie* *pociąg* *przez* *Warszawę.*
From Posen-GEN goes train-NOM through Warsaw-ACC.
'A train goes from Posen through Warsaw.'



Inner models A proposition in a subordinate clause doesn't need to be implied by the whole sentence. We place it in a separate box in order to indicate that its truth value ought to be determined against the subjective point of view:

(7) *Jan wierzy, że słoń trąbi.*
Jan believes, that elephant trumpets.
'Jan believes that an elephant trumpets.'



A meta-predicate DSCR is added to the logical notation is a tool for representing relations between embedded models in the subject language .

$$\exists(w, \text{TYPE}(w, \text{believe}) \wedge \exists(j, \text{TYPE}(j, \text{person}) \wedge \text{HASNAME}(j, \text{'Jan'}) \wedge |j| = 1, \text{INITIATOR}(w, j)) \wedge \exists(x,$$
$$\text{TYPE}(x, \text{that}) \wedge \text{DSCR}(x, \exists(s, \text{TYPE}(s, \text{elephant}) \wedge |s| = 1,$$
$$\exists(t, \text{TYPE}(t, \text{trumpet}) \wedge \text{INITIATOR}(t, s)))), \text{CONDITION}(w, x)))$$

3 Conclusions

Presented system is novel not only as a tool for semantic processing of Polish. ENIAM introduces the fuzzy pipeline approach to language processing and implements a subset of LCG form large scale language processing. It also takes advantage of huge semantic resources (such as Słowosieć and Walenty) which were created as a part of CLARIN-PL project. The fact that ENIAM does not require any prior annotation of processed sentences make it universal and ready to use tool.

Acknowledgements

Work financed as part of the investment in the CLARIN-PL research infrastructure funded by the Polish Ministry of Science and Higher Education.

References

Rafał L. Górski Adam Przepiórkowski, Mirosław Bańko and Barbara Lewandowska-Tomaszczyk. 2012. National corpus of polish. *Wydawnictwo Naukowe PWN, Warsaw*, pages 51–58.

Johan Bos, Stephen Clark, Mark Steedman, James R Curran, and Julia Hockenmaier. 2004. Wide-coverage semantic representations from a ccg parser. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1240. Association for Computational Linguistics.

Marie-Catherine De Marneffe and Christopher D Manning. 2008. Stanford typed dependencies manual. Technical report, Technical report, Stanford University.

Marek Maziarz, Maciej Piasecki, Ewa Rudnicka, and Stan Szpakowicz. 2014. plwordnet as the cornerstone of a toolkit of lexico-semantic resources. In *Proceedings of the Seventh Global Wordnet Conference*, pages 304–312.

Glyn Morrill. 2010. *Categorial grammar: Logical syntax, semantics, and processing*. Oxford University Press.

Adam Przepiórkowski, Elżbieta Hajnicz, Agnieszka Patejuk, and Marcin Woliński. 2014. Extended phraseological information in a valence dictionary for NLP applications. In *Proceedings of the Workshop on Lexical and Grammatical Resources for Language Processing (LG-LP 2014)*, pages 83–91, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Marcin Wolinski, Marcin Milkowski, Maciej Ogrodniczuk, and Adam Przepiórkowski. 2012. Polimorf: a (not so) new open morphological dictionary for polish. In *LREC*, pages 860–864.

Saloni Zygmunt, Woliński Marcin, Wołosz Robert, Gruszczyński Włodzimierz, and Skowrońska Danuta. 2015. Grammatical dictionary of polish.