Parsing of Polish in graph database environment

Jan Posiadała, Hubert Czaja, Eliza Szczechla, Paweł Susicki

Scott Tiger S.A.
15 Kolektorska Street, Warsaw, Poland
{janek,czajah,eliza,trzeci}@tiger.com.pl

Abstract

This paper describes the basic concepts and features of the Langusta system. Langusta is a natural language processing environment embedded in a graph database. The paper presents a rule-based syntactic parsing system for the Polish language using various linguistic resources, including those containing semantic information. The advantages of this approach are directly related to the deployment of the graph paradigm, in particular to the assumption, that rules describing the syntax of the Polish language are valid queries in a graph database query language (Cypher).

Keywords: NLP, graph databases, Cypher, deep parsing, corpus analysis, written corpora, stand-off annotation

1. Introduction

A number of papers have been published discussing various aspect of the synergy that exists between graph theory methods and the problems in natural language processing (NLP). Many of them (Ide and Suderman, 2007; Pęzik, 2013) focus on the issue of the multi-source and multi-layer annotation of natural language corpora. Application of the graph model to the problem of structuring linguistic informationresults in the abandon of inline annotations for more clear and flexible standoff annotations (Zeldes et al., 2009) without impact on annotation semantic.

All the publications mentioned above, along with this paper, emphasise the high level of generality of the graph model as well as the multiplicity and maturity of tools and algorithms used in the graph theory (Mihalcea and Radev 2011). Recent years have brought, along with the development of the NoSQL movement (Strauch, 2011), a significant growth in the field of the database systems implementing the graph paradigm such as Neo4j[1], OrientDB[2] or Apache TinkerPop[3].

This area of research has evolved from theoretical models, providing simple and elegant solutions for the basic NLP problems such as morphosyntactic annotation and modelling of the word-sense ambiguity and semantic role labelling, towards those solutions that lie on the borderline with the artificial intelligence domain, such as information extraction and QA (question & answer) systems support.

A persistent corpus representation featuring an underlying graph model and a high structural openness has resulted in a change in the language processing paradigm. The classic, pipeline based approach is usually implemented as a set of programs performing subsequent stages of linguistic processing (Graliński ea al., 2012, Shi et al., 2014) of a text represented in one of the standard markup formats (Przepiórkowski and Bański, 2009). The graph based representation sees this method replaced by a corpus-centric model of gradual enrichment of the graph representation by adding new layers of linguistic annotation. The solution has all the characteristics of the standoff annotation and emphasizes its advantages (Dipper, 2005). Also, with this data representation, the deployment of a rule-based system in the architecture of the text processing components often proves advantageous (Negnevitsky, 2001). We will show how this potential has been exploited in the Langusta system.

The approximate location of the used parsing method in the theoretical background will be facilitated when we notice that the main mechanism to calculate the result of rules applying is a pattern-matching, a mechanism which is less general and powerful then unification. Consequently, the presented method should be regarded as a rule-based parsing technique performed in a propitious executive environment. Therefore, the presented solution is not an implementation of formalism for unification grammar, nor a proposal for a new kind of formalism, in the type of Tree Adjoining Grammar (Joshi and Schabes, 1997).

The authors of this paper are aware of the abundance of context discussed above. Therefore, in our work we concentrate on the application of the graph model to deep syntactic parsing – a key issue in NLP (Szpakowicz 1978; Świdziński 1992).

1.1. Langusta

1.1.1. Assumptions and inspirations

Taking up the challenge of building a language processing environment was inspired by the need to build a rule-driven text processing system for the Polish language. One important design goal was to enable syntactic parsing of the Polish language without restrictions on parsing depth.

Our motivation has been strengthened by the analysis of the advantages and limitations of the SPEJD (Przepiórkowski, 2008; Buczyński and Przepiórkowski, 2008) system. SPEJD is a shallow parsing system for the

[1] http://neo4j.com/

[2] http://orientdb.com/orientdb/

[3] Apache TinkerPop Project is most known for providing a set of interfaces that graph databases that database vendors can implement (Blueprints) to get all the features of the rest of the TinkerPop stack (Pipes, Gremlin, Frames, Rexster, Furnace) where each part of the stack provides a specific function in supporting graph-based application development; http://tinkerpop.apache.org/

Polish language, created by Linguistic Engineering Group in PAS and distributed under GNU General Public License.

Considering the identified needs, the inspiring capabilities of the SPEJD system include:

- clear and conceptually sophisticated formalism available for defining parsing rules (syntactic and semantic head, left and right surrounding)
- parser to tokenizer integration
- parser to morphosyntactic analyser integration and an extensible morphosyntactic tagset

Later, in the context of the identified needs, the inspiring limitations to the SPEJD system include:

- the inability to model the ambiguity of the results of the parsing process
- lack of an open, non-volatile representation of the final and intermediate results
- the inability to easily integrate with external language resources

Because of the profound inspiration taken from the SPEJD program, some of the examples of the syntactic parser rules quoted in this paper will be compared to the corresponding SPEJD formalism rules.

2. Data model

The graph data model is derived directly from the concept of a graph used in graph theory (Wilson, 1996). Langusta uses a directed property graph as its data model, adopting one of the most popular approaches to graph-based data modelling.

In this model, the basic concepts are:

- node - with labels and attributes,
- directed edge - with a label and attributes,
- path - a finite sequence of edges which connect a sequence of nodes.

The implementation supports simple Java compliant[4] types of attributes: boolean, byte, short, int, long, float, double, char, String and also heterogeneous arrays of values of those types. Among the numerous advantages of this model, structural capacity seems to be the most important in the foreseen applications. This strength offers a prospect of facilitating the incorporation and analysis of new resources within the designed system. This expectation can be further justified by the argument that the graph model is native to many linguistic resources e.g. plWordNet (Maziarz et al., 2012).

2.1. Query language

The data model described above enables the design and implementation of a query language. One example of such a language is Cypher (Robinson et al., 2014), originally implemented as query language for the Neo4j graph database. Cypher is a declarative, pattern-matching, graph model compliant query language for a graph database.

The choice of a query language as a declarative way to access the data was an important decision, due to the fact, that in a sense, Langusta is intended as a data analysis system. Taking this into account and also the fact that the graph model is a paradigmatic model, there were an inclination to choose the native language to the property graph model - both in semantic and syntactic sense. The assumption that the author of the rules - but also the data analyst - will be a linguistic expert created intention to preserve intuitive character of graph model in the processing of (and in the access to) data. That meant the rejection of query languages which syntax or computation model is derived from SQL, RDF or logic programming (Wood, 2009). An additional advantage of Cypher was the syntactical simplicity of manipulating of the paths and node's properties. The rule-based clause order in Cypher queries is not without significance, and increases resemblance to the SPEJD formalism.

A cypher language query is composed of the three basic clauses:

MATCH clause: The MATCH clause is the core element of a Cypher query. In this clause we describe the matching criteria for the sought subgraph. The primary way of setting out criteria for the subgraph is describing the structure of nodes connected by edges and tagged by labels.

```
MATCH
(p:PERSON)-[r:KNOWS]->(pp:PERSON)
RETURN
p, r, pp
```

Identifiers used in the MATCH clause to name nodes, edges and paths are bound with the corresponding matched objects in the database (nodes, edges, paths).

WHERE clause: The WHERE clause contains a boolean expression that filters objects sought in MATCH clause:

```
MATCH
(p:PERSON)-[r:KNOWS]->(pp:PERSON)
WHERE
p.age > pp.age
RETURN
p, r, pp
```

RETURN clause: The RETURN clause contains expressions returned as the result of a query for the subgraph meeting the selection (match and filter) criteria.

3. Parsing of Polish

3.1. Tokenization

Text tokenization implementation in Langusta does not go beyond the basic definition, i.e., its result is splitting text into tokens (words) and sentences (Mazur, 2005). In particular, the method developed for the PWN Corpus of Polish has been implemented (Rudolf and Swidzinski, 2004). In implementation there was no distinction between the layer of tokens and multi-token words, no less, the data model used in Langusta has a sufficient capacity to make the separation between tokens layer and words layer, in order to model the ambiguity of multi-token words.

[4] Java types description: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html

3.2. Morphosyntactic analysis

The morphosyntactic annotation is based on the morphosyntactic dictionary PoliMorf (Woliński et al., 2012). As a result of the process, a text structure is formed in the graph database. In this structure for each token there is a corresponding set of nodes representing a collection of interpretations from the morphosyntactic dictionary for which the inflected form is equal to the token (ignoring case). These nodes are labelled with the label `Word`. Each `Word` node has appropriate values of its grammatical class and its grammatical categories stored in its attributes.

The order of tokens in a sentence is represented by the relationship `:follows`. The relationship occurs between any two nodes `Word` representing consecutive tokens from the processed sentence. The process yields the following graph structure for the sentence: *"Młode dziewczyny biegły."*[5]. The sentence is tokenized to: "Młode", "dziewczyny", "biegły", ".".

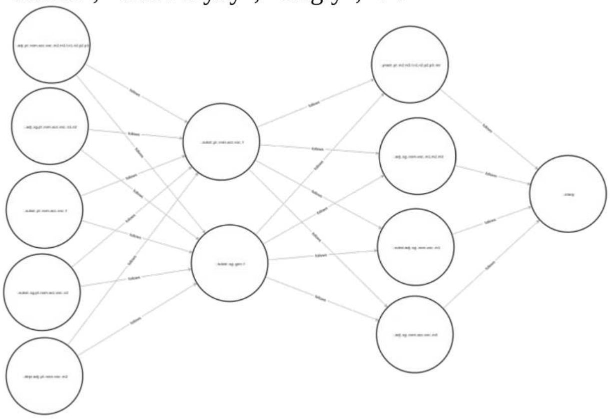

Fig. 1: Structure of sentence: "*Młode dziewczyny biegły.*" with compressed morphosyntactic interpretation in nodes caption.

3.2.1. Morphosyntactic dictionary compression

Langusta works with the content of the morphosyntactic dictionary in a compressed format. For this purpose the system uses a shortened representation of the grammatical interpretation. The shortened notation (Woliński and Przepiórkowski, 2001) is a widely used method, because of the Polish language system syncretism. In Langusta the shortened notation mentioned above is used as the compression method.

This has resulted in a significant reduction of the number of nodes representing tokens and their grammatical interpretations which will undergo further rule-based processing.

For this purpose, the content of the dictionary PoliMorf was transformed to an atomized form, i.e. each entry containing the alternative values for a grammatical category has been split into atomic entries containing unambiguous values. Next, for all the positions sharing a common inflected form and a common basic form, the set of atomized grammatical interpretations has been compressed using the shortened notation. The sum of Cartesian products of compressed entries equals the original set containing atomized entries.

Let us consider selected dictionary entries corresponding the inflected form: "*młode*".

młode młoda subst:pl:voc:f
młode młoda subst:pl:nom:f
młode młoda subst:pl:acc:f
młode młode subst:pl:voc:n2
młode młode subst:pl:nom:n2
młode młode subst:pl:acc:n2
młode młode subst:sg:acc:n2
młode młode subst:sg:voc:n2
młode młode subst:sg:nom:n2
młode młode depr:pl:voc:m2
młode młody depr:pl:nom:m2
młode młody adj:sg:nom.voc:n1.n2:pos
młode młody adj:sg:acc:n1.n2:pos
młode młody adj:pl:acc:m2.m3.f.n1.n2.p2.p3:pos
młode młody adj:pl:nom.voc:m2.m3.f.n1.n2.p2.p3:pos

The above shortened notation expands to 38 atomized dictionary entries with non-empty values for grammatical class (subst, depr, adj) and non-empty values for grammatical case, number and gender. As the result of compression we get 5 entries (see Fig. 1):

młode młoda subst:.pl:nom.acc.voc:f
młode młode subst:sg.pl:nom.acc.voc:n2
młode młody adj:pl:nom.acc.voc:m2.m3.f.n1.n2.p2.p3
młode młody depr.adj:pl:nom.voc:m2
młode młody adj:sg.pl:nom.acc.voc:n1.n2

The direct consequence of compression of the morphosyntactic dictionary is a change to the types of attribute values in nodes representing data derived from that dictionary, i.e. string attributes become string array attributes.

3.3. Parsing rules

The Langusta rules performing the syntactic analysis are valid Cypher queries. Let us consider the parsing rule used for the phrase "młode dziewczyny". The core of the rule looks as follows:

```
MATCH[6]
(adj)--(subst)
WHERE
subst.pos  *=  ['ger','pact',  'ppas',
'subst']
and 'adj' in adj.pos
and adj.gender *= subst.gender
and adj.number *= subst.number
and adj.case *= subst.case
```

The corresponding SPEJD rule (Przepiórkowski and Buczyński, 2007) accurate to a set of tags and set of syntactic groups used in the National Corpus of Polish (Przepiórkowski et al., 2012) is as follows[7],[8]:

[5] Translation to English: "Young girls run."

[6] The list intersection operator `*=` is not supported by the implementation of Cypher in the Neo4j database. The interpretation is: false if and only if the list is empty.

[7] Correspondence between WHERE expression in Langusta rule and unify operator in SPEJD rule is limited to condition component of unify operator. Application of Langusta rule rejects no interpretation.

[8] Correspondence between semantic of group action in SPEJD rule and consequence of Langusta rule application seems to be very strong, obviously excluding capability

```
Match:([pos~"Adj|Pact|Ppas"]|
[type="AdjG|AdjGk"])
([pos~"Noun"] | [type="NGg|NGs|NGb"]);
Eval: unify(case number gender,1,2);
group(NGa,2,2);
```

The full version of the rule in Langusta is as follows:

```
WITH
'R.B.Subst.01' as code,
100 as rate ,
90 as InversionRate ,
['znajomy      kolega',      'znajomy
krzywdzący',  'znajomy  pokrzywdzony',
'znane zagranie'] as examples
MATCH
(adj)--(subst)
WHERE
subst.pos  *=  ['ger',  'pact',  'ppas',
'subst'] and
'adj' in adj.pos and
adj.gender *= subst.gender and
adj.number *= subst.number and
adj.case *= subst.case
RETURN subst as synh
```

In the WITH clause, in which the computational environment for the query is predefined, the following values are passed in:

- Code – a mnemotechnic rule ID
- Rate – the weight of the rule
- InversionRate - the weight of the inverted rule[9]
- Examples - examples of expressions parsed by the rule.

In the RETURN clause the node matched under the criteria for the subst variable is aliased synh for further processing. The node will pass its syntactic features on to a new node representing the parsed phrase "*młode dziewczyny*".

The mechanism described above is insufficient to ensure accuracy of the values of attributes storing grammatical categories of the newly created node.

In the sample rule RBSubst.01 one could observe that values of the attributes storing grammatical categories should be consistent with the conditions of equalisation for grammatical attributes of the nodes adj and subst. And so, the values of attributes in the newly created node should compile with the listing beneath:

Attribute	Expression	Value
pos	subst.pos *= ['ger', 'pact', 'ppas', 'subst']	['subst']
number	adj.number *= subst.number	['pl']
case	adj.case *= subst.case	['nom','acc','voc']
gender	adj.gender *= subst.gender	['f']

Extending of a rule with expressions for attribute values for each newly created node is carried out automatically on the basis of the analysis of the conditions of equalization.

3.3.1. Algorithm

The parsing algorithm applies parsing rules to the text corpus represented as a graph. The graph structure is the output from the process of tokenization and morphosyntactic analysis. When a rule is satisfied a new `Word` node is created to represent the correspondent piece of text. The new node inherits the syntactic features from the node designated as synh, i.e. from the syntactic head. The new node inherits all the incoming `:follows` relationships from the first node in the matched path. Likewise, the new node inherits all the outgoing `:follows` relationships from the last node in the matched path. Lastly, an `:is element of` relationship is being created between the new node and all the `Word` nodes of the path matched by the use of the rule.

The algorithm applies the rule set until no rule produces a new node. The algorithm guarantees that no rule will be successfully applied more than once to the same sequence of `Word` nodes, thus ensuring the uniqueness of their representation.

3.4. Additional linguistic resources

The structural capacity of the graph data model allows for a straightforward incorporation of additional linguistic resources that can be used to increase precision of parsing rules.

3.4.1. plWordNet

One such example is the lexical database for the Polish language, plWordNet. With this solution, the semantic dependencies can be applied at the stage of performing syntactic analysis. Let us consider the beneath core of rule designed to match phrases like "*butelka z benzyną*" or "*worki na liście*".[10]

```
MATCH[11] (cont:Word)--(prep)--(subst),
cont<-[:occurs]-(f:Form),
f-[:formof]->(b:Base),
b-[:means]->(lu:LexicalUnit),
lu<-[r:DNRS_hiponimia]-h
WHERE    subst.pos    *=    ['ppron12',
'ppron3', 'subst']
```

of ambiguity representation.

[9] Langusta supports the handling of word order inversion which is common in the Polish language which is a synthetic language. Through this mechanism the number of rules for parsing the corresponding expressions in normal and inverted order is not doubled. The use of mechanism is limited to rules which match 2 Word nodes. That means that in Langusta system, the expression "*dziewczyny młode*" will be parsed by the same rule (although certainly not by the same query). To apply the a given rule to the inverted word order it suffices to pass in the appropriate InversionRate value in the environment, i.e. the value of the weight for the rule which tries to perform matching using inverted order of of matching nodes.

[10] Phrases "*bottle of gasoline*", "*sacks for leaves*" as instances of prepositional phrases: "*container of/for something*". "*Bottle*" and "*sack*" are hyponyms of "*container*" and inherit its valency features.

[11] When the MATCH clause contains more than one path, Langusta selects the first one as the matching path by default. The unnamed and undirected relationships between the nodes on this path are labelled `:follows` and directed from left to right.

```
and 'subst' in container.pos
and prep.base in ['na', 'po', 'z']
and 'prep' in prep.pos and
and prep.case *= subst.case
and h.name in ['pojemnik','zbiornik']
```

In the above query we require that the noun `cont` be a form of a word that is a hyponyme[12] of one of the nouns: “pojemnik” or “zbiornik”.

A query of this type can increase the precision of parsing of phrases like: “*Worek na ziemię został rzucony.*”.

3.4.2. Walenty

It is only natural for a Langusta user to extend the basic set of rules by adding new rules automatically generated from the existing language resources. One such example is reaching out for the set of rules automatically generated from the valence dictionary Walenty (Przepiórkowski et al., 2014) and including them in the system.

The core of a parser rule in Langusta corresponding to a valence rule in the Walenty Dictionary:

buntować:pewny:_:imperf: prepnp(przeciw,dat)

has the form of:

```
WITH 'R.Wal.Prep.Za.Inst' as code, 40
as rate,['buntować'] as verbBases
MATCH (Verb)--(przeciw)--(subst)
WHERE 'inst' in subst.case
and subst.pos *= ['ger', 'ppron12',
'ppron3', 'siebie', 'subst']
and 'przeciw' = przeciw.base
and 'inst' in przeciw.`case`
and 'prep' in przeciw.pos
and Verb.base in verbBases
and Verb.pos *= ['fin', 'ger', 'imps',
'impt', 'inf', 'pact', 'pant', 'pcon',
'ppas', 'praet']
```

4. Conclusions and discussions

The distinctive features of the presented approach include:

- An open-structure, persistent and queryable corpora representation
- A transparent way of dealing with ambiguity the grammatical interpretation of inflected forms and with multiplicity of syntactic trees constructed; this has been achieved by unleashing the potential of the large structural capacity of the graph data model
- Choosing an open, declarative query language as the formalism used for the description of parsing rules and the resulting ease of the automated generation of grammatical rules based on the primary linguistic resources (see 3.4.2)
- The ease of representation and use of the basic language resources for the Polish and the ease of incorporation of additional linguistic resources (see 3.4.1)

Synergy of the above features builds up an advantage over other solutions, especially in the area of deep syntactic parsing of Polish. The advantage manifests itself in the ease of parse rule set management as well as quality of received results.

Most important works considered for the future include:

- Comprehensive analysis of the performance aspects of the proposed solution
- A comprehensive comparison of analytical capabilities for the presented solution with search engine Poliqarp[13][14]
- A more extensive use of the linguistic resources for the Polish language, currently used and those under development (i.e. semantic frames in Walenty)
- Application of the Langusta environment to more advanced topics from the field of natural language processing and understanding such as relation extraction or multi-text summarization
- Enhancing the solution by the introduction of a statistical component. The authors believe that the graph paradigm based approach present in the current solution may well be adopted in the future system featuring support for statistical methods.
- One of the identified areas of application of the statistical component is system tuning by choosing weights for rules based on statistical data.

As described above, the deployment of graph database environment has met our expectations sufficiently to allow for planning further development of the solution.

References

Buczyński, A. and Przepiórkowski, A. (2008). *Demo: An Open Source Tool for Partial Parsing and Morphosyntactic Disambiguation*. In: *Proceedings of LREC 2008*.

Dipper, S. (2005). *Stand-off representation and exploitation of multi-level linguistic annotation.* In Proceedings of Berliner XML Tage 2005 (BXML 2005), pages 39–50, Berlin.

Graliński, F. Jassem, K. and Junczys-Dowmunt, M. (2012). *PSI-Toolkit: Natural language processing pipeline. Computational Linguistics - Applications*, Heidelberg: Springer.

Ide, N. and Suderman, K. (2007). *GrAF: A Graph-based Format for Linguistic Annotations. Proceedings of the Linguistic Annotation Workshop*, 1–8, Prague: Czech Republic.

Joshi A.K., Schabes Y. (1997). *Tree-adjoining grammars In: Handbook of formal languages*, vol. 3 Pages 69-123 Springer-Verlag New York, Inc. New York, NY, USA ISBN:3-540-60649-1

[12] To increase ease of use of the plWordNet dictionary, the rules work with the transitive closure of the WordNet graph, traversing the hyponymy relation edges, taking into account transition through synset groups, i.e. if a lexical unit: lu1 is a hyponyme of a lexical unit lu2, then all the lexical units sharing the same synset group with lu1 are hyponymes of all lexical units sharing a synset group with lu2.

[13] Poliqarp, similary to SPEJD, based its syntax on the formalism CQP derived from the project CWB - The IMS Open Corpus Workbench (http: //cwb.sourceforge.net/)

[14] Poliqarp, similary to SPEJD, is was used as a part of NKJP project.

Negnevitsky, M. (2001). *Artificial Intelligence: A Guide to Intelligent Systems*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 2001

Maziarz, M., Piasecki, M. and Szpakowicz, S. (2012). *Approaching plWordNet 2.0. Proceedings of the 6th Global Wordnet Conference*. Matsue: Japan.

Mazur, P. (2005) *Text Segmentation in Polish*. In the: *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 43–48, 8th–10th September 2005, Wroclaw: Poland.

Mihalcea, R. and Radev, D. (2011). *Graph-Based Natural Language Processing and Information Retrieval*, Cambridge: UK: Cambridge University Press.

Pęzik, P. (2013*). Indexed graph databases for querying rich TEI annotation*. Retrieved from: http://digilab2.let.uniroma1.it/teiconf2013/wp-content/uploads/2013/09/Pezik.pdf.

Przepiórkowski, A. (2008). *Powierzchniowe przetwarzanie języka polskiego*. Warsaw: Akademicka Oficyna Wydawnicza EXIT.

Przepiórkowski, A., Bańko, M., Górski, R.L. and Lewandowska-Tomaszczyk, B., editors. (2012). *Narodowy Korpus Języka Polskiego.* Wydawnictwo Naukowe PWN, Warsaw.

Przepiórkowski, A. and Bański, P. (2009) *Which XML Standards for Multilevel Corpus Annotation?* Proceedings of the 4th Language & Technology Conference, Poznań, Poland.

Przepiórkowski, A. and Buczyński, A., (2007). *Shallow Parsing and Disambiguation Engine*. In: Zygmunt Vetulani, editor, *Proceedings of the 3rd Language & Technology Conference*, pages 340–344, Poznań: Poland.

Przepiórkowski, A., Hajnicz, E., Patejuk, A., Woliński, M., Skwarski, F. and Świdziński M.. (2014). *Walenty: Towards a comprehensive valence dictionary of Polish*. *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, LREC 2014, pages 2785–2792, Reykjavík: Iceland: ELRA.

Robinson, I., Webber, J. and Eifrem, E. (2013). *Graph Databases.* O'Reilly Media

Rudolf M., Świdziński M. (2004). *Automatic utterance boundaries recognition in large Polish text corpora.* In Mieczysław A. Kłopotek; Sławomir T. Wierzchoń, Krzysztof Trojanowski, editors. Intelligent Information Systems. Proceedings of the International IIS: IIPWM ´04 page 247-256. Conference Held in Zakopane, Poland, May 17-20, 2004, Springer

Shi, C., Verhagen, M. and Pustejovsky, M. (2014). *A Conceptual Framework of Online Natural Language Processing Pipeline Application, Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT, pages 53–59, Dublin, Ireland, August 23rd.*

Strauch, Ch. (2011). *NoSQL Databases* http://www.christof-strauch.de/nosqldbs.pdf

Szpakowicz, S., (1978). *Automatyczna analiza składniowa polskich zdań pisanych*. Praca doktorska (promotor Waligórski S.), Instytut Informatyki UW.

Świdziński, M., (1992), *Gramatyka formalna języka polskiego*, „*Rozprawy Uniwersytetu Warszawskiego*”, t. 349, Warsaw.

Wilson. J.R. (1996) *Introduction to Graph Theory (4th Edition)* Addison Wesley

Woliński, M., Miłkowski, M., Ogrodniczuk, M., Przepiórkowski A. and Szałkiewicz Ł. (2012) *PoliMorf: a (not so) new open morphological dictionary for Polish*. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012*, pages 860–864, Istanbul: Turkey: ELRA

Woliński, M. and Przepiórkowski, A. (2001) *Projekt anotacji morfosynktaktycznej korpusu języka polskiego. Prace IPI PAN 938, grudzień 2001*

Wood P.T. (2009), Query languages for graph databases, ACM SIGMOD Record, v.41 n.1, March 2012

Zeldes, A., Ritz, J., Lüdeling, A. and Chiarcos, C. (2009). *ANNIS: A Search Tool for Multi-Layer Annotated Corpora. In Proceedings of Corpus Linguistics 2009*, Liverpool: July 20-23, 2009.