Semantic parsing using Interpreted Regular Tree Grammars

Evelin Ács[1] Gábor Recski[2]
[1] *Department of Theoretical Linguistics*
Eötvös Loránd University
[2] *Department of Automation and Applied Informatics*
Budapest University of Technology and Economics
acs.evelin@gmail.com
recski@aut.bme.hu

Abstract. Syntactic parsers for natural language are key components for most processing pipelines within human language technologies. The most common approach taken by modern HLT systems is dependency parsing, which maps raw text to directed acyclic graphs (DAGs) over words of each input sentence. One class of systems, which includes the de facto standard Stanford Parser, creates dependency graphs via rule-based transformation of constituency structures output by a PCFG parser. Semantic parsers map raw text to some representation of meaning. The dep_to_4lang component of the 4lang library builds graphs of syntax-independent concepts from the output of any dependency parser that conforms to the Universal Dependencies format, also using simple template-matching. Since all components of this end-to-end semantic parser pipeline implements a form of graph transformation, its functionality can be unified in a single graph grammar. In this paper we present a reimplementation of the dep_to_4lang functionality using an interpreted regular tree grammar (IRTG), which maps rules of a regular tree grammar (RTG) to pairs of operations over UD and 4lang graphs, thereby allowing for efficient transformation between the two representations. The system supports the UD v2.1 format and automatic generation of terminal rules, it is therefore capable of basic semantic parsing on UD data for 80+ languages.

Keywords: Semantic parsing; Natural language processing; IRTG; 4lang; Graph transformation

1 Introduction

This paper reimplements and extends the functionality of `dep_to_4lang`, a core component of the semantic parsing system `text_to_4lang` [1], using an Interpreted Regular Tree Grammar [2] with multiple s-graph [3] interpretations. Using the open-source IRTG parser library `alto`, our grammar enables efficient two-way transformation between 4lang-style semantic representations and the output of any dependency parser that supports the Universal Dependencies [4] format. Sections 2.1 and 2.2 provide a brief overview of the fields of semantic parsing and dependency parsing, respectively. In Section 2.3 we give a summary of IRTGs and s-graph grammars, the formal tools used to build our parser, and

Frameset **accept.01** "take willingly"
Arg0: Acceptor
Arg1: Thing accepted
Arg2: Accepted-from
Arg3: Attribute

Figure 1: The argument structure of the verb *accept* [6, p. 75.].

we also briefly describe the `alto` library. Section 3 discusses our contribution: the IRTG that realizes the mapping between UD and 4lang and extends the functionality of the `text_to_4lang` system. Both our grammar and scripts used for data processing are available on GitHub[1] under an MIT license.

2 Background

2.1 Semantic parsing

Semantic parsing is the process of mapping raw natural language text to a representation of its meaning. Most current NLP applications rely on continuous vector space models of meaning that are constructed from large text corpora and are based solely on the *distribution* of words, i.e. statistics over the contexts in which they appear. The models discussed here, on the other hand, represent meaning as directed graphs of words and/or concepts, abstracting away from syntax to various degrees. We give an overview of two formalisms: Abstract Meaning Representations (AMRs) have gained popularity over the 5 years since they have been introduced, several parser systems have already been published. `4lang` is a highly language- and syntax-independent formalism based on a more general theory of semantics and inference.

2.1.1 AMR

AMR [5] is a system for representing the meaning of English sentences. AMRs are directed graphs of words based on *framesets* from `PropBank`, an annotated corpus of semantic roles that focuses on the argument structure of verbs ([6]). An example of a `PropBank` frameset is shown in Figure 1, an AMR representation of a short English sentence in Figure 2.

AMR nodes represent entities, properties, events, or states. Leaves are labeled with words, PropBank framesets, or keywords. Keywords are special entity types, quantities, or logical conjunctions. AMR uses approximately 100 relations to link entities ([5]), these represent general semantic relations, coreference, questions, modals, negation, etc. AMR's limitations include lack of treatment of inflectional morphology, determiners, and the universal quantifier.

[1] `https://github.com/kornai/4lang/tree/master/exp/alto`

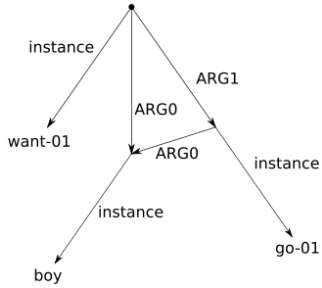

Figure 2: Representing the meaning of "The boy wants to go"([5], page 179).

Also, AMR graphs do not distinguish between real and hypothetical events or between events that happen in the past, present, or future.

2.1.2 4lang

The 4lang theory [7] represents meaning as directed graphs of language- and grammar-independent concepts. Edges may receive one of three labels: 0-edges represent attribution ($apple \xrightarrow{0} delicious$), the IS_A relation ($emu \xrightarrow{0} bird$) and unary predication ($cat \xrightarrow{0} meow$). 1- and 2-edges connect binary predicates to their arguments ($John \xleftarrow{1} buy \xrightarrow{2} book$).

The 4lang library[2] contains tools for building 4lang graphs from raw text (`text_to_4lang`) and dictionary definitions (`dict_to_4lang`). The system obtains dependency relations from text by invoking the Stanford Dependency Parser ([8]). A subsequent mapping from Stanford dependencies ([9]) to subgraphs of nine possible graph configurations (Table 1) was built manually. We present the reimplementation of this mapping with several modifications and enhancements in Section 3. 4lang is also the name of a manually built concept dictionary [10] containing more than 2000 definition graphs of language-independent concepts. The 4lang dictionary also maps each concept to words in four languages (Hungarian, English, Latin, Polish), hence its name.

2.2 Dependency parsing

Although constituent-based grammars have been the dominant approach to syntax in both contemporary linguistic theory and also in natural language processing of the past decades, recent NLP systems prefer *grammatical dependencies* as the primary source of syntactic information in downstream applications. All variants of dependency grammar (see [11] for a survey) represent sentences as directed, labeled edges between its words, each of which stands for a head-dependent relation, such that each word is dependent on exactly one other word in the sentence. Figure 3 shows an example analysis.

[2] `https://github.com/kornai/4lang`

Dependency	Edge
amod advmod npadvmod acomp dep num prt	$w_1 \overset{0}{\rightarrow} w_2$
nsubj csubj xsubj agent	$w_1 \underset{0}{\overset{1}{\rightleftharpoons}} w_2$
dobj pobj nsubjpass csubjpass pcomp xcomp	$w_1 \overset{2}{\rightarrow} w_2$
appos	$w_1 \underset{0}{\overset{0}{\rightleftharpoons}} w_2$
poss prep_of	$w_2 \overset{1}{\leftarrow} \texttt{HAS} \overset{2}{\rightarrow} w_1$
tmod	$w_1 \overset{1}{\leftarrow} \texttt{AT} \overset{2}{\rightarrow} w_2$
prep_with	$w_1 \overset{1}{\leftarrow} \texttt{INSTRUMENT} \overset{2}{\rightarrow} w_2$
prep_without	$w_1 \overset{1}{\leftarrow} \texttt{LACK} \overset{2}{\rightarrow} w_2$
prep_P	$w_1 \overset{1}{\leftarrow} \texttt{P} \overset{2}{\rightarrow} w_2$

Table 1: Mapping from Stanford dependency relations to 4lang subgraphs [1, p. 12.].

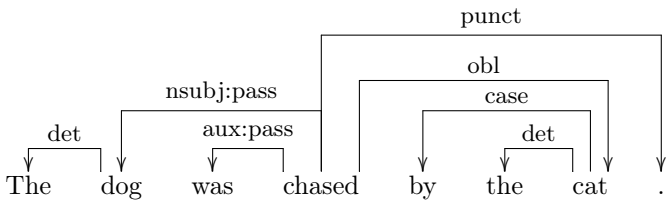

Figure 3: Dependency analysis of the sentence 'The dog was chased by the cat.' Source: http://universaldependencies.org/introduction.html

While early approaches to dependency parsing were based on manually built grammars, today's systems employ machine learning models that are trained on dependency treebanks and try to predict the most likely analysis for previously unseen sentences. For an overview of various ML-based approaches to dependency parsing, the reader is referred to [12]. Dependency parsing is used in a variety of NLP applications such as sentiment analysis [13, 14] or question answering [15]. The Universal Dependencies (UD) project[3] provides a cross-linguistically consistent grammatical annotation and more than 100 UD treebanks in 60 languages (as of version 2.1, released in November 2017) [4]. UDs also enables the cross-linguistic evaluation of dependency parsers: more than 30 teams participated in the 2017 CoNLL shared task on multilingual dependency parsing [16].

2.3 IRTGs and Alto

We reimplement the transformation between dependency graphs and 4lang graphs using interpreted regular tree grammars [17]. Intuitively, IRTGs are context-free grammars in which each rule is mapped to operations over an arbitrary number of algebras, so that any series of operations on any of these algebras that corresponds to at least one derivation in the IRTG will in turn correspond to a series of operations over each of the other algebras. For example, Figure 4 illustrates a derivation in an IRTG whose two *interpretations* are both mappings to algebras formed by strings and concatenation. This IRTG can be used to implement transformation between pairs of strings by establishing a correspondence between their derivations. Formally, RTGs are defined as quadruples $\mathcal{G} = (N, \Sigma, P, S)$ where N is a signature of nonterminal symbols, Σ is a signature of terminal symbols, $S \in N$ is a distinguished start symbol, and P is a finite set of productions of the form $B \to t$, where B is a nonterminal symbol, and $t \in T_{N \cup \Sigma}$ [17]. Σ-interpretations are defined as pairs $\mathcal{I} = (h, \mathcal{A})$ where $\mathcal{A}$ is a Δ-algebra and $h : T_\Sigma \to T_\Delta$ is a (tree) homomorphism. Then IRTGs can be defined as $(\mathcal{G}, \mathcal{I}_1, \mathcal{I}_2, \ldots)$, where $\mathcal{I}_i$ are Σ-interpretations [17].

S-graph algebras or HR algebras [18] have recently been used with IRTGs for semantic parsing [2]. Here we provide only a brief, informal overview of s-graph algebras, see [17] for a more formal explanation. The central operation of an

[3] `http://universaldependencies.org/`

RTG rule	homomorphisms
$S \to r_1(NP, VP)$	1: $x_1 \bullet x_2$ 2: $x_1 \bullet x_2$
$VP \to r_2(V, NP)$	1: $x_1 \bullet x_2$ 2: $x_2 \bullet x_1$
$NP \to r_3$	1: John 2: Hans
$NP \to r_4$	1: the box 2: die Kiste
$NP \to r_5$	1: opens 2: öffnet

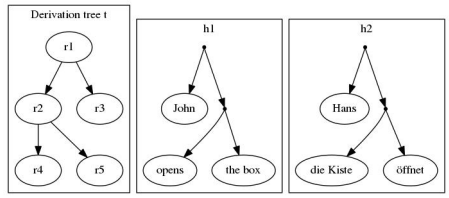

Figure 4: An IRTG with an example derivation ([2], page 4)

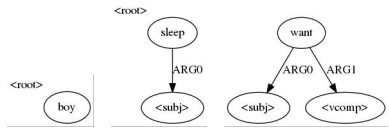

Figure 5: S-graph examples ([2], page 5)

s-graph algebra is the binary `merge`, which unifies pairs of s-graphs following a simple mechanism specified by labels on each s-graph: an s-graph's nodes may be marked with a set of *source names*, each taken from a fixed finite set. Source names identify nodes that should be merged when entire subgraphs are `merged`: when two s-graphs G_1 and G_2 are `merged`, the resulting s-graph G' will contain all nodes of G_1 and G_2, and when a pair of nodes $(v_1, v_2) \in E(G_1) \times E(G_2)$ have the same source name, they will be mapped to a single node v' in G' that has all adjacent edges of v_1 and v_2. In semantic applications, e.g. [2], source names correspond to semantic argument positions of a grammar. An example of `merge` over semantic graphs is shown in Figures 5 and 6, here the `root` source node indicates the head of the construction, `subj` and `vcomp` mark slots for subject and verbal complement, respectively.

The Algebraic Language Toolkit, or `alto`[4] [19] is an open-source parser for IRTGs. It implements a variety of algebras for use with IRTGs, including the s-graph algebra. When constructing a grammar file for `alto`, one must explicitly

[4] `https://bitbucket.org/tclup/alto`

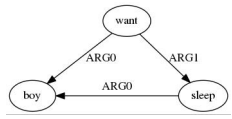

Figure 6: S-graphs of Figure 5 combined ([2], page 6)

```
X -> _nsubj(X, X)
[graph] r_gov_root(f_dep(merge(
    merge(r_gov(?1), "(g<gov> :nsubj (d<dep>))"), r_dep(?2))))
[fourlang] r_gov_root(f_dep(merge(
    merge(r_gov(?1), "(g<gov> :1 (d<dep> :0 g))"), r_dep(?2))))
```

Figure 7: An example from our grammar, illustrating the operations of the HR algebra. The grammar is discussed further in Section 3.

declare all interpretations with types, then provide for each RTG rule mappings to operations for each of these interpretations. The example in Figure 7 shows a rule that, when interpreted over dependency graphs, connects dedicated (`root`) nodes of two subgraphs with a single directed edge that represents the grammatical dependency `nsubj`. The same rule, when applied to 4lang graphs, will create the configuration $w_1 \underset{0}{\overset{1}{\rightleftharpoons}} w_2$. The RTG rule and its two interpretations together form an IRTG rule that implements a mapping between an operation on pairs of UD graphs and an operation on pairs of 4lang graphs. The `gov` and `dep` sources used in the example mark the start and end points of directed edges, the `f_dep` and `r_gov_root` operations are `alto`'s shorthands for `forget(dep)` and `rename(gov, root)`.

3 Semantic parsing with IRTGs

We present an IRTG which reimplements `dep_to_4lang` functionality, extends it to support UD (as opposed to classic Stanford Dependencies, or SD), and provides a treatment of some additional phenomena, such as various uses of the UD relation `case`. The grammar file is available on GitHub.[5] For the most part, our mapping is an upgrade of `dep_to_4lang` rules from SD to UD (see Table 2 for the UD-conform version). Correspondence between such simple configurations are easily implemented by rules similar to that in Figure 7. The `case` relation requires a slightly more complex set of rules.

In English, the UD relation `case` connects prepositions and their arguments in prepositional phrases. The governing phrases are in turn marked as arguments of either a predicate (e.g. *give the toys [to the children]*), a modified nominal phrase (*the office [of the chair]* or the subject of a copular sentence (*Sue is [in shape]*). In these cases the argument of the preposition may be the dependent of the relation `obl`, `nmod`, or `nsubj`, respectively (see Fig-

[5] `https://github.com/kornai/4lang/blob/master/exp/alto/ud/en_ud_bi.irtg`

Dependency	**Edge**
advcl	$w_1 \xrightarrow{0} w_2$
advmod	
amod	
nmod	
nummod	
appos	$w_1 \overset{0}{\underset{0}{\rightleftharpoons}} w_2$
dislocated	
csubj	$w_1 \overset{1}{\underset{0}{\rightleftharpoons}} w_2$
nsubj	
ccomp	$w_1 \xrightarrow{2} w_2$
obj	
xcomp	

Table 2: UD-conform version of the mapping of Table 1.

ure 8). In 4lang, each of these prepositions would correspond to binary relations: `give` $\xleftarrow{1}$ `to` $\xrightarrow{2}$ `children`, `office, chair` $\xleftarrow{1}$ `of` $\xrightarrow{2}$ `,` `Sue` $\xleftarrow{1}$ `in` $\xrightarrow{2}$ `shape`. We add three ternary rules to our grammar to map each use of `case` to the appropriate 4lang graph, the one treating `obl` arguments is shown in Figure 9.

We validate our grammar by parsing the English section of the UD v2.1 treebank. For this purpose we use a simple scripts to convert CoNLL-formatted dependencies to the `ISI`-format supported by `alto` and another couple of one-liners to automatically create preterminal and terminal IRTG rules, e.g.

```
X -> _VERB(VERB)
[graph] ?1
[fourlang] ?1

VERB -> find
[graph] "(find<root> / find)"
[fourlang] "(find<root> / find)"
```

Our system supports the UD v2.1 format and automatic generation of terminal rules, it is therefore capable of basic semantic parsing on UD data for 80+ languages.

4 Conclusion

In its current form, the system presented in this paper is merely a reimplementation of an existing tool for semantic parsing that, when compared to its predecessor, is more efficient, easier to maintain, and truly language independent by virtue of the Universal Dependencies project. Several other core NLP

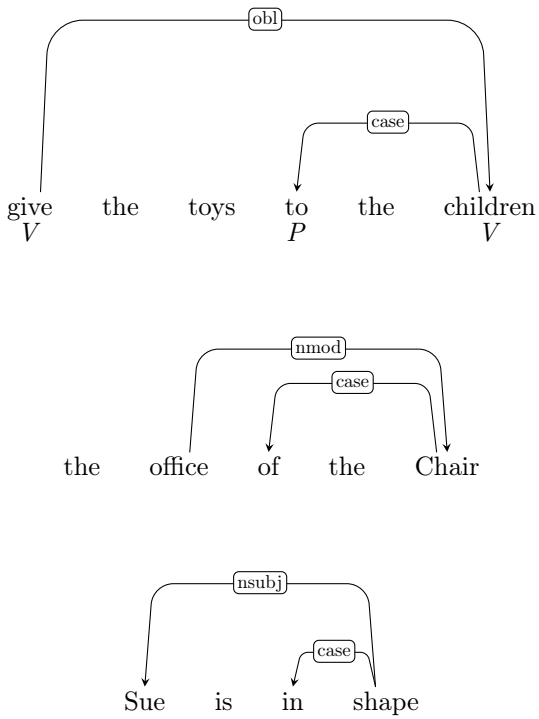

Figure 8: Uses of the `case` relation (source: http://universaldependencies.org/)

```
X -> _obl_case(X, X, X)
[graph] r_gov_root(f_dep2(f_dep1(merge(merge(
    merge(r_gov(?1), "(g<gov> :obl (d1<dep1> :case (d2<dep2>)))"),
    r_dep1(?2)), r_dep2(?3)))))
[fourlang] r_gov_root(f_dep2(f_dep1(merge(merge(
    merge(r_gov(?1), "(d2<dep2> :1 (g<gov>) :2 (d1<dep1>))"),
    r_dep1(?2)), r_dep2(?3)))))
```

Figure 9: One of the three rules for the `obl` relation

tasks, however, are instances of graph transformation. Most state-of-the-art dependency parsers, for example, build dependency graphs incrementally, in a bottom-up fashion. Semantic inference, when performed over concept networks, also involves manipulating graphs. Not only could one attempt to reimplement any such system as an IRTG, a grammar generic enough may implement multiple processing steps simultaneously. Our short term plans include an IRTG for mapping UD to AMR graphs and another for reimplementing the Stanford Parser's mapping between syntactic constituents and dependency graphs. Then, if a single probabilistic IRTG were to implement the parallel parsing of strings, syntactic constituency structures, dependency graphs, and semantic graphs like 4lang or AMR, it could be trained simultaneously on each of these types of gold-standard data as a single end-to-end system for semantic parsing.

References

[1] G. Recski, "Building concept definitions from explanatory dictionaries," *International Journal of Lexicography*, 2018.

[2] A. Koller, "Semantic construction with graph grammars," in *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, (London), 2015.

[3] J. Groschwitz, A. Koller, and C. Teichmann, "Graph parsing with s-graph grammars," in *Proceedings of the 53rd ACL and 7th IJCNLP*, (Beijing), 2015.

[4] J. Nivre, Ž. Agić, L. Ahrenberg, L. Antonsen, M. J. Aranzabe, M. Asahara, L. Ateyah, M. Attia, A. Atutxa, L. Augustinus, E. Badmaeva, M. Ballesteros, E. Banerjee, S. Bank, V. Barbu Mititelu, J. Bauer, K. Bengoetxea, R. A. Bhat, E. Bick, V. Bobicev, C. Börstell, C. Bosco, G. Bouma, S. Bowman, A. Burchardt, M. Candito, G. Caron, G. Cebiroğlu Eryiğit, G. G. A. Celano, S. Cetin, F. Chalub, J. Choi, S. Cinková, Ç. Çöltekin, M. Connor, E. Davidson, M.-C. de Marneffe, V. de Paiva, A. Diaz de Ilarraza, P. Dirix, K. Dobrovoljc, T. Dozat, K. Droganova, P. Dwivedi, M. Eli, A. Elkahky, T. Erjavec, R. Farkas, H. Fernandez Alcalde, J. Foster, C. Freitas, K. Gajdošová, D. Galbraith, M. Garcia, M. Gärdenfors, K. Gerdes, F. Ginter, I. Goenaga, K. Gojenola, M. Gökırmak, Y. Goldberg, X. Gómez Guinovart, B. Gonzáles Saavedra, M. Grioni, N. Grūzītis, B. Guillaume, N. Habash, J. Hajič, J. Hajič jr., L. Hà My, K. Harris, D. Haug, B. Hladká, J. Hlaváčová, F. Hociung, P. Hohle, R. Ion, E. Irimia, T. Jelínek, A. Johannsen, F. Jørgensen, H. Kaşıkara, H. Kanayama, J. Kanerva, T. Kayadelen, V. Kettnerová, J. Kirchner, N. Kotsyba, S. Krek, V. Laippala, L. Lambertino, T. Lando, J. Lee, P. Lê Hong, A. Lenci, S. Lertpradit, H. Leung, C. Y. Li, J. Li, K. Li, N. Ljubešić, O. Loginova, O. Lyashevskaya, T. Lynn, V. Macketanz, A. Makazhanov, M. Mandl, C. Manning, C. Mărănduc, D. Mareček, K. Marheinecke, H. Martínez Alonso, A. Martins, J. Mašek,

Y. Matsumoto, R. McDonald, G. Mendonça, N. Miekka, A. Missilä, C. Mititelu, Y. Miyao, S. Montemagni, A. More, L. Moreno Romero, S. Mori, B. Moskalevskyi, K. Muischnek, K. Müürisep, P. Nainwani, A. Nedoluzhko, G. Nešpore-Bērzkalne, L. Nguyen Thi, H. Nguyen Thi Minh, V. Nikolaev, H. Nurmi, S. Ojala, P. Osenova, R. Östling, L. Øvrelid, E. Pascual, M. Passarotti, C.-A. Perez, G. Perrier, S. Petrov, J. Piitulainen, E. Pitler, B. Plank, M. Popel, L. Pretkalniņa, P. Prokopidis, T. Puolakainen, S. Pyysalo, A. Rademaker, L. Ramasamy, T. Rama, V. Ravishankar, L. Real, S. Reddy, G. Rehm, L. Rinaldi, L. Rituma, M. Romanenko, R. Rosa, D. Rovati, B. Sagot, S. Saleh, T. Samardžić, M. Sanguinetti, B. Saulīte, S. Schuster, D. Seddah, W. Seeker, M. Seraji, M. Shen, A. Shimada, D. Sichinava, N. Silveira, M. Simi, R. Simionescu, K. Simkó, M. Šimková, K. Simov, A. Smith, A. Stella, M. Straka, J. Strnadová, A. Suhr, U. Sulubacak, Z. Szántó, D. Taji, T. Tanaka, T. Trosterud, A. Trukhina, R. Tsarfaty, F. Tyers, S. Uematsu, Z. Urešová, L. Uria, H. Uszkoreit, S. Vajjala, D. van Niekerk, G. van Noord, V. Varga, E. Villemonte de la Clergerie, V. Vincze, L. Wallin, J. N. Washington, M. Wirén, T.-s. Wong, Z. Yu, Z. Žabokrtský, A. Zeldes, D. Zeman, and H. Zhu, "Universal dependencies 2.1," 2017. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

[5] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider, "Abstract meaning representation for sembanking," in *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, (Sofia, Bulgaria), pp. 178–186, Association for Computational Linguistics, 2013.

[6] M. Palmer, D. Gildea, and P. Kingsbury, "The proposition bank: An annotated corpus of semantic roles," *Computational linguistics*, vol. 31, no. 1, pp. 71–106, 2005.

[7] A. Kornai, J. Ács, M. Makrai, D. M. Nemeskey, K. Pajkossy, and G. Recski, "Competence in lexical semantics," in *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics (*SEM 2015)*, (Denver, Colorado), pp. 165–175, Association for Computational Linguistics, 2015.

[8] M.-C. DeMarneffe, W. MacCartney, and C. Manning, "Generating typed dependency parses from phrase structure parses," in *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, vol. 6, (Genoa, Italy), pp. 449–454, 2006.

[9] M.-C. De Marneffe and C. D. Manning, "Stanford typed dependencies manual," tech. rep., Technical report, Stanford University, 2008.

[10] A. Kornai and M. Makrai, "A 4lang fogalmi szótár," in *IX. Magyar Számitógépes Nyelvészeti Konferencia* (A. Tanács and V. Vincze, eds.), pp. 62–70, 2013.

[11] J. Nivre, "Dependency grammar and dependency parsing," *MSI report*, vol. 5133, no. 1959, pp. 1–32, 2005.

[12] D. Jurafsky and J. H. Martin, "Computing with word senses." `https://web.stanford.edu/~jurafsky/slp3/`, 2018. 3rd ed. draft.

[13] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pp. 347–354, Association for Computational Linguistics, 2005.

[14] Y. Wu, Q. Zhang, X. Huang, and L. Wu, "Phrase dependency parsing for opinion mining," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pp. 1533–1541, Association for Computational Linguistics, 2009.

[15] H. Cui, R. Sun, K. Li, M.-Y. Kan, and T.-S. Chua, "Question answering passage retrieval using dependency relations," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 400–407, ACM, 2005.

[16] D. Zeman, M. Popel, M. Straka, J. Hajic, J. Nivre, F. Ginter, J. Luotolahti, S. Pyysalo, S. Petrov, M. Potthast, F. Tyers, E. Badmaeva, M. Gokirmak, A. Nedoluzhko, S. Cinkova, J. Hajic jr., J. Hlavacova, V. Kettnerová, Z. Uresova, J. Kanerva, S. Ojala, A. Missilä, C. D. Manning, S. Schuster, S. Reddy, D. Taji, N. Habash, H. Leung, M.-C. de Marneffe, M. Sanguinetti, M. Simi, H. Kanayama, V. dePaiva, K. Droganova, H. Martínez Alonso, c. Çöltekin, U. Sulubacak, H. Uszkoreit, V. Macketanz, A. Burchardt, K. Harris, K. Marheinecke, G. Rehm, T. Kayadelen, M. Attia, A. Elkahky, Z. Yu, E. Pitler, S. Lertpradit, M. Mandl, J. Kirchner, H. F. Alcalde, J. Strnadová, E. Banerjee, R. Manurung, A. Stella, A. Shimada, S. Kwak, G. Mendonca, T. Lando, R. Nitisaroj, and J. Li, "Conll 2017 shared task: Multilingual parsing from raw text to universal dependencies," in *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, (Vancouver, Canada), pp. 1–19, Association for Computational Linguistics, 2017.

[17] A. Koller and M. Kuhlmann, "A generalized view on parsing and translation," in *Proceedings of the 12th International Conference on Parsing Technologies*, pp. 2–13, Association for Computational Linguistics, 2011.

[18] B. Courcelle, "Graph grammars, monadic second-order logic and the theory of graph minors," *Contemporary Mathematics*, vol. 147, pp. 565–565, 1993.

[19] J. Gontrum, J. Groschwitz, A. Koller, and C. Teichmann, "Alto: Rapid prototyping for parsing and translation," in *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 29–32, 2017.