View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Repository of the Academy's Library

Building Definition Graphs using Monolingual Dictionaries of Hungarian

Gábor Recski[1], Attila Bolevácz[1], Gábor Borbély[2]

[1] Research Institute for Linguistics
Hungarian Academy of Sciences
recski@mokk.bme.hu, attila.bolevacz@protonmail.hu

[2] Department of Algebra
Budapest University of Technology
borbely@math.bme.hu

1 Introduction

We adapt to Hungarian core functionalitites of the `4lang` library [12], which builds `4lang` -style semantic representations [7] from raw text using an external dependency parser as proxy, and processes definitions of monolingual dictionaries to build definition graphs for concepts not defined in the hand-written `4lang` dictionary [8]. In Section 2 we provide a short overview of the `4lang` formalism, Section 3 describes the architecture of the `text_to_4lang` and `dict_to_4lang` systems. We describe in detail the steps taken to adapt our system to Hungarian in Section 4. The new tool is evaluated in Section 5. The new components presented in this paper are part of the latest version of the `4lang` library, which is available under an MIT license from http://www.github.com/kornai/4lang.

2 The `4lang` representation

`4lang` is both a formalism for representing meaning via directed graphs of concepts and also the name of a manually built lexicon of such representations for ca. 2700 words[3]. A formal presentation of the system is given in [7], the theoretical principles underlying `4lang` are presented in [5], we shall provide a short overview only.

`4lang` meaning representations are directed graphs of concepts with three types of edges. Nodes of `4lang` graphs correspond to *concepts*. `4lang` concepts are not words, nor do they have any grammatical attributes such as part-of-speech (category), number, tense, mood, voice, etc. For example, `4lang` representations make no distinction between the meaning of *freeze* (N), *freeze* (V), *freezing*, or *frozen*. Therefore, the mapping between words of some language and the language-independent set of `4lang` concepts is a many-to-one relation. In particular, many concepts will be defined by a single link to another concept

[3] https://github.com/kornai/4lang/blob/master/4lang

that is its hypernym or synonym, e.g. `above` $\xrightarrow{0}$ `up` or `grasp` $\xrightarrow{0}$ `catch`. Encyclopaedic information is omitted, e.g. `Canada`, `Denmark`, and `Egypt` are all defined as `country`, their definitions also containing an indication that an external resource (we use Wikipedia for this) may contain more information. In general, definitions are limited to what can be considered the shared knowledge of competent speakers - e.g. the definition of `water` contains the information that it is a colourless, tasteless, odourless liquid, but not that it is made up of hydrogen and oxigen.

The most common connection in `4lang` graphs is the 0-edge, which represents attribution: `dog` $\xrightarrow{0}$ `friendly`, the `IS_A` relation (synonymy and hypernymy): `dog` $\xrightarrow{0}$ `animal`, and unary predication: `dog` $\xrightarrow{0}$ `bark`. Edge types 1 and 2 connect binary predicates to their arguments, e.g. `cat` $\xleftarrow{1}$ `catch` $\xrightarrow{2}$ `mouse`). There are no ternary or higher arity predicates, see [6]. The formalism used in the `4lang` dictionary explicitly marks binary (transitive) elements – by using UPPERCASE printnames. The tools presented in this paper make no use of this distinction, any concept can have outgoing 1- and 2-edges. However, we will retain the uppercase marking for those binary elements that do not correspond to any word in a given phrase or sentence. The `4lang` tools described here also enforce a slight modification to the formalism: the 0-relation shall hold between a subject and predicate regardless of whether the predicate has another argument, so that e.g. the `4lang` representations for *John eats* and *John eats a muffin* shall share the subgraph `John` $\xrightarrow{0}$ `eat`. The `4lang` dictionary contains manually specified definition graphs for ca. 2700 concepts, a typical definition in the dictionary can be seen in Figure 1. `4lang` contains words for each concept in four languages: English, Hungarian, Polish, and Latin.

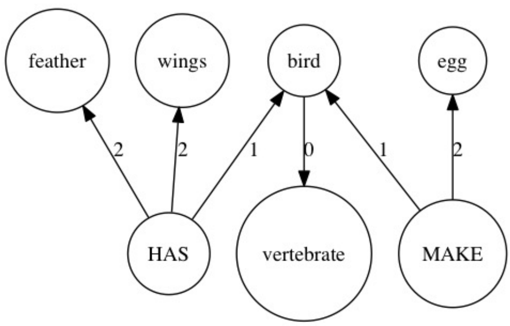

Fig. 1. 4lang definition of `bird`.

3 Architecture

The core tools in the 4lang library include the dep_to_4lang module for processing the output of a dependency parser and building 4lang representations by mapping dependencies to graph edges, the text_to_4lang module for using this functionality for mapping raw text to 4lang graphs, and the dict_to_4lang module for processing monolingual dictionaries to acquire definition graphs for words not manually defined in the 4lang dictionary. We now give a brief overview of these systems before presenting the modifications that enable us to run them on Hungarian data in Section 4.

The dep_to_4lang module implements a mapping from dependency triplets output by a syntactic parser to subgraphs over 4lang concepts corresponding to content words in the sentence. Words are lemmatized using the hunmorph morphological analyzer [13], concept nodes are created for lemmas of each content word that takes part in a dependency relation that dep_to_4lang processes. The output of the dependency parser is first postprocessed by a separate, language-specific module that recognizes some patterns of dependencies and adds new triplets based on them that can later be used to create the correct 4lang subgraphs. The mapping itself enforces two types of rules: some dependencies trigger an edge between two nodes, e.g. for a relation dobj(x, y) the edge $\mathtt{y} \xrightarrow{2} \mathtt{x}$ is added. Other relations will result in a binary node being added to the graph, e.g. the triplet tmod(x, y) will trigger $\mathtt{x} \xleftarrow{1} \mathtt{AT} \xrightarrow{2} \mathtt{y}$ (for a description of all Stanford dependency types see [2], for the full mapping for English see [12]). When processing raw English text using the text_to_4lang module, the Stanford Coreference Resolution system is run in addition to the Stanford Dependency parser and pairs of nodes in the resulting 4lang graph are unified accordingly. The dict_to_4lang module for processing dictionary definitions contains parsers for various monolingual dictionaries of English, and also runs a preprocessor for each datasource that transforms the definitions in order to make them easier to parse and more informative; e.g. the pattern someone who will be removed from the beginning of Longman definitions, reducing parser errors considerably, but without losing any relevant information: the pattern also triggers the addition of the edge $\xrightarrow{0} \mathtt{person}$ to the definition graph. Finally, the root node of each definition, which nearly always corresponds to a hypernym of the headword, is unified with the headword's node.

4 Modifications for Hungarian

In order to adapt the text_to_4lang and dict_to_4lang pipelines to Hungarian, we used the NLP library magyarlanc for dependency parsing and implemented a mapping to 4lang graphs that is sensitive to the output of morphological analysis – to account for the rich morphology of Hungarian encoding many relations that a dependency parse cannot capture. We describe the output of magyarlanc and the straightforward components of our mapping in Section 4.1. In Section 4.2 we discuss the use of morphological analysis in our pipeline, and in Section 4.3 we present some arbitrary postprocessing steps similar to those already implemented for English.

We shall also use our modifications to run the `dict_to_4lang` pipeline on two explanatory dictionaries of Hungarian: volumes 3 and 4 of the *Magyar Nyelv Nagyszótára* (`NSzt`), containing nearly 5000 headwords starting with the letter *b* [4][4], and over 120 000 entries of the complete *Magyar Értelmező Kéziszótár* (`EKsz`) [10], which has previously been used for NLP research [9]. Preprocessing of definitions involved replacing abbreviations in definitions, e.g. replacing *vmi* with *valami* 'something' or *Mo.* with *Magyarország* 'Hungary', performed by the `eksz_parser` and `nszt_parser` modules.

4.1 Dependencies

The `magyarlanc` library[5] [15] contains a suite of standard NLP tools for Hungarian, which allows us, just like in the case of the Stanford Parser, to perform tokenization, morphological analysis, and dependency parsing using a single tool. The dependency parser component of `magyarlanc` is a modified version of the Bohnet parser [1] trained on the Szeged Dependency Treebank [14]. The output of `magyarlanc` contains a much smaller set of dependencies than that of the Stanford Parser. Parses of the ca. 4700 entries of the NSzT data contain nearly 60,000 individual dependencies, 97% of which are covered by the 10 most frequent dependency types. The dependencies `att, mode,` and `pred`, all of which express some form of unary predication, can be mapped to the 0-edge. `subj` and `obj` are treated in the same fashion as the Stanford dependencies `nsubj` and `dobj`. The dependencies `from, tfrom, locy, tlocy, to,` and `tto` encode the relationship to the predicate of adverbs and postpositional phrases answering the questions 'from where?', 'from when?', 'where?', 'when?', 'where to?', and 'until when?', respectively, hence they are mapped to the binary relations `FROM, since, AT, TO,` and `until` (see Table 1).

4.2 Morphology

In Hungarian the relationship between a verb and its NP argument is often encoded by marking the noun phrase for one of 21 distinct cases – in English, these relations would typically be expressed by prepositional phrases. While the Stanford Parser maps prepositions to dependencies and the sentence *John climbed under the table* yields the dependency `prep_under(table, climb)`, the Hungarian parser does not transfer the morphological information to the dependencies, all arguments other than subjects and direct objects will be in the `OBL` relation with the verb. Therefore we updated the `dep_to_4lang` architecture to allow our mappings from dependencies to `4lang` subgraphs to be sensitive to the morphological analysis of the two words between which the dependency holds. The

[4] The author gratefully acknowledges editor-in-chief Nóra Ittzés for making an electronic copy available.

[5] http://www.inf.u-szeged.hu/rgai/magyarlanc

Table 1. Mapping from `magyarlanc` dependency relations to `4lang` subgraphs

Dependency	Edge
att mode pred	$w_1 \xrightarrow{0} w_2$
subj	$w_1 \xrightarrow{1} w_2$
obj	$w_1 \xrightarrow{2} w_2$
from	$w_1 \xleftarrow{1} \texttt{FROM} \xrightarrow{2} w_2$
tfrom	$w_1 \xleftarrow{1} \texttt{since} \xrightarrow{2} w_2$
locy tlocy	$w_1 \xleftarrow{1} \texttt{AT} \xrightarrow{2} w_2$
to	$w_1 \xleftarrow{1} \texttt{TO} \xrightarrow{2} w_2$
tto	$w_1 \xleftarrow{1} \texttt{until} \xrightarrow{2} w_2$

resulting system maps the phrase *a késemért jöttem* the knife-POSS-PERS1-CAU come-PAST-PERS1 'I came for my knife' to `FOR(come, knife)` based on the morphological analysis of *késem* performed by `magyarlanc` based on the `morphdb.hu` database [13].

While this method yields many useful subgraphs, it also often leaves uncovered the true semantic relationship between verb and argument, since nominal cases can have various interpretations that are connected to their 'primary' function only remotely, or not at all. The semantics of Hungarian suffixes *-nak/-nek* (dative case) or *-ban/-ben* (inessive case) exhibit great variation – not unlike that of the English prepositions *for* and *in*, and the 'default' semantic relations `FOR` and `IN` are merely one of several factors that must be considered when interpreting a particular phrase. Nevertheless, our mapping from nominal cases to binary relations can serve as a strong baseline, just like interpreting English *for* and *in* as `FOR` and `IN` via the Stanford dependencies `prep_for` and `prep_in`. The full mapping from nominal cases of `OBL` arguments to `4lang` binaries is shown in Table 2.

4.3 Postprocessing

In the Szeged Dependency Treebank, and consequently, in the output of `magyarlanc`, copular sentences will contain the dependency relation `pred`. Hungarian only requires a copular verb in these constructions when a tense other than the present or a mood other than the indicative needs to be marked (cf. Figure 3). While the first example is analyzed as `subj(Ervin, álmos)`, all remaining sentences will be assigned the dependencies `subj(Ervin, volt)` and `pred(volt, álmos)`. The same copular structures allow the predicate to be a noun phrase

Table 2. Mapping nominal cases of `OBL` dependants to `4lang` subgraphs

Case	Suffix	Subgraph
sublative	*-ra/-re*	$w_1 \xleftarrow{1} \texttt{ON} \xrightarrow{2} w_2$
superessive	*-on/-en/-ön*	
inessive	*-ban/-ben*	$w_1 \xleftarrow{1} \texttt{IN} \xrightarrow{2} w_2$
illative	*-ba/-be*	
temporal	*-kor*	$w_1 \xleftarrow{1} \texttt{AT} \xrightarrow{2} w_2$
adessivel	*-nál/nél*	
elative	*-ból/-ből*	$w_1 \xleftarrow{1} \texttt{FROM} \xrightarrow{2} w_2$
ablative	*-tól/-től*	
delative	*-ról/-ről*	
allative	*-hoz/-hez/-höz*	$w_1 \xleftarrow{1} \texttt{TO} \xrightarrow{2} w_2$
terminative	*-ig*	
causative	*-ért*	$w_1 \xleftarrow{1} \texttt{FOR} \xrightarrow{2} w_2$
instrumental	*-val/-vel*	$w_1 \xleftarrow{1} \texttt{INSTRUMENT} \xrightarrow{2} w_2$

(e.g. *Ervin tűzoltó* 'Ervin is a firefighter'). In each of these cases we'd like to eventually obtain the `4lang` edge `Ervin` $\xrightarrow{0}$ `sleepy` (`Ervin` $\xrightarrow{0}$ `firefighter`), which could be achieved in several ways: we might want to detect whether the nominal predicate is a noun or an adjective and add the `att` and `subj` dependencies accordingly. Both of these solutions would result in a considerable increase in the complexity of the `dep_to_4lang` system and neither would simplify its input: the simplest examples (such as (1) in Figure 3) would still be treated differently from all others. With these considerations in mind we took the simpler approach of mapping all pairs of the form `nsubj(x, c)` and `pred(c, y)` (such that `c` is a copular verb) to the relation `subj(x, y)`, which can then be processed by the same rule that handles the simplest copulars (as well as verbal predicates and their subjects.)

Unlike the Stanford Parser, `magyarlanc` does not propagate dependencies across coordinated elements. Therefore we introduced a simple postprocessing step where we collect words of the sentence governing a `coord` dependency, then find for each the words accessible via `coord` or `conj` dependencies (the latter connects coordinating conjunctions such as *és* 'and' to the coordinated elements). Finally, we unify the dependency relations of all coordinated elements[6].

[6] This step introduces erroneous edges in a small fraction of cases: when a sentence contains two or more clauses that are not connected by any conjunction – i.e. no connection is indicated between them – a `coord` relation is added by `magyarlanc` to connect the two dependency trees at their root nodes.

Table 3. Hungarian copular sentences

(1)	*Ervin*	*álmos*		
	Ervin	sleepy		
	'Ervin is sleepy'			
(2)	*Ervin*	*nem*	*álmos*	
	Ervin	not	sleepy	
	'Ervin is not sleepy'			
(3)	*Ervin*	*álmos*	*volt*	
	Ervin	sleepy	was	
	'Ervin was sleepy'			
(4)	*Ervin*	*nem*	*volt*	*álmos*
	Ervin	not	was	sleepy
	'Ervin was not sleepy'			

5 Evaluation

5.1 `text_to_4lang`

To evaluate the `text_to_4lang` pipeline we chose 20 random sentences and checked the output manually. The source of our sample is the Hungarian Webcorpus [3], to obtain a random sample we ran the GNU utility `shuf` on a sequence of files containing one sentence on each line. We shall start by providing some rough numbers regarding the average quality of the 20 `4lang` graphs, then proceed to discuss some of the most typical issues, citing examples from our sample. 10 of the 20 graphs were correct `4lang` representations, or had only minor errors. An example of a correct transformation can be seen in Figure 3. Of the remaining graphs, 4 were mostly correct but had major errors, e.g. 1-2 content words in the sentence had no corresponding node, or several erroneous edges were present in the graph. The remaining 6 graphs had many major issues and can be considered mostly useless.

When investigating the processes that created the more problematic graphs, nearly all errors seem to be caused by sentences with multiple clauses. When a clause is introduced by a conjunction such as *hogy* 'that' or *ha* 'if', the dependency trees of each graph are connected via these conjunctions only, i.e. the parser does not assign dependencies that hold between words from different clauses. While we are able to build good quality subgraphs from each clause, further steps are required to establish the semantic relationship between them based on the type of conjunction involved – a process that requires case-by-case treatment. An example from our sample is the sentence in Figure 2; here a conditional clause is introduced by a phrase that roughly translates to 'We'd be glad if...'. Even if we disregard the fact that a full analysis of how this phrase affects the semantics of the sentence would require some model of the speaker's desires – clearly beyond our systems current capabilities – we could still interpret the sentence literally by imposing some rule for conditional sentences, e.g. that given a structure of the form A if B, the `CAUSE` relation is to hold between the root nodes of B and A. Such arbitrary rules could be introduced for several types of conjunctions in the future. A further, smaller issue is caused by the general lack of personal pronouns in sentences: Hungarian is a *pro-drop* language: if a verb is inflected for person, pronouns need not be present to indicate the subject of the verb, e.g. *Eszem.* 'eat-1SG' is the standard way of saying 'I'm eating' as opposed to *?Én eszem* 'I eat-1G' which is only used in special contexts where emphasis is necessary. Currently this means that `4lang` graphs built from these sentences will have no information about who is doing the `eat`ing, but in the future these cases can be handled by a mechanism that adds a pronoun subject to the graph based on the morphological analysis of the verb. Finally, the lowest quality graphs are caused by very long sentences containing several clauses and causing the parser to make multiple errors.

Örülnénk,	*ha*	*a*	*konzultációs*	*központok*
rejoice-COND-1PL	if	the	consultation-ATT	center-PL
közötti	*kilométerek*	*nem*	*jelentenének*	
between-ATT	kilometer-PL	not	mean-COND-3PL	
az	*emberek*	*közötti*	*távolságot.*	
the	person-PL	between-ATT	distance-ACC	
'We'd be glad if the kilometers between consultation centers did not mean distance between people'				

Fig. 2. Subordinating conjunction

5.2 `dict_to_4lang`

We also conducted manual error analysis on the output of the `dict_to_4lang` pipeline, in this case choosing 20 random words from the EKsz dictionary[7]. The graphs built by `dict_to_4lang` were of very good quality, with only 3 out of 20 containing major errors. This is partly due to the fact that `NSzt` contains many very simple definitions, e.g. 4 of the 20 headwords in our random sample contained a (more common) synonym as its definition. All 3 significant errors are caused by the same pattern: the analysis of possessive constructions by `magyarlanc` involve assigning the `att` dependency to hold between the possessor and the possessed, e.g. the definition of `piff-puff` (see Figure 4) will receive the dependencies `att(hang, kifejezés)` and `att(lövöldözés, hang)`, resulting in the incorrect `4lang` graph in Figure 5

[7] the 20 words, selected once again using `shuf`, are the following: *állomásparancsnok, beköt, biplán, bugás, egyidejűleg, font, főmufti, hajkötő, indikál, lejön, munkásőr, nagyanyó, nemtelen, összehajtogat, piff-puff, szét, tipográfus, túlkiabálás, vakolat, zajszint*

1995	*telén*	*vidrafelmérést*	*végeztünk*
1995	winter-POSS-SUP	otter-survey-ACC	conduct-PST-1PL
az	*országos*	*akció*	*keretében.*
the	country-ATT	action	frame-POSS-INE
‘In the winter of 1995 we conducted an otter-survey as part of our national campaign’			

⇓

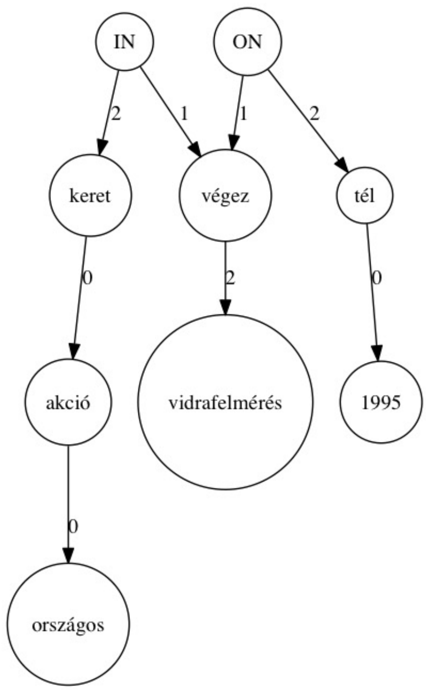

Fig. 3. Example of perfect `dep_to_4lang` transformation

instead of the expected one in Figure 6. $\texttt{kifejezés} \xrightarrow{0} \texttt{hang} \xrightarrow{0} \texttt{lövöldözés}$ instead of $\texttt{kifejezés} \xleftarrow{2} \texttt{HAS} \xrightarrow{1} \texttt{hang} \xleftarrow{2} \texttt{HAS} \xrightarrow{1} \texttt{lövöldözés}$. These constructions cannot be handled even by taking morphological analysis into account, since possessors are not usually marked (although in some structures they receive the dative suffix *-nak/-nek*, e.g. in embedded possessives like our current example (*hangjának* 'sound-POSS-DAT' is marked by the dative suffix as the possessor of *kifejezésére*). Unless possessive constructions can be identified by `magyarlanc`, we shall require an independent parsing mechanism in the future. The structure of Hungarian noun phrases can be efficiently parsed using the system described in [11], the grammar used there may in the future be incorporated into a `4lang` -internal parser, plans for which are outlined in [12].

Lövöldözés	*vagy*	*ütlegelés*	*hangjának*	*kifejezésére*
Shooting	or	thrashing	sound-POSS-DAT	expression-POSS-DAT

'Used to express the sound of shooting or thrashing'

⇓

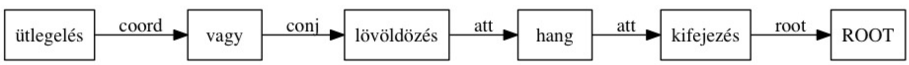

Fig. 4. Dependency parse of the `EKsz` definition of the (onomatopoeic) term `piff-puff`

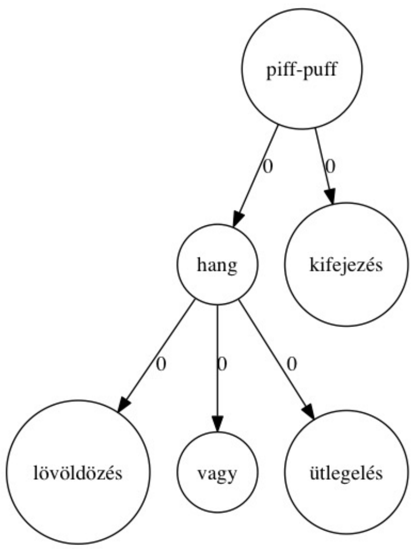

Fig. 5. Incorrect graph for `piff-puff`

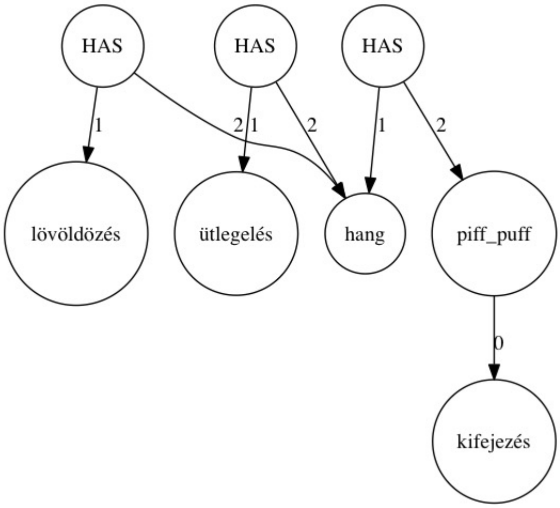

Fig. 6. Expected graph for `piff-puff`

References

1. Bernd Bohnet. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 89–97, Beijing, China, August 2010. Coling 2010 Organizing Committee.
2. Marie-Catherine DeMarneffe, William MacCartney, and Christopher Manning. Generating typed dependency parses from phrase structure parses. In *Proc. LREC*, volume 6, pages 449–454, Genoa, Italy, 2006.
3. Péter Halácsy, András Kornai, László Németh, András Rung, István Szakadát, and Viktor Trón. Creating open language resources for Hungarian. In *Proceedings of the 4th international conference on Language Resources and Evaluation (LREC2004)*, pages 203–210, 2004.
4. Nóra Ittzés, editor. *A magyar nyelv nagyszótára III-IV*. Akadémiai Kiadó, 2011.
5. András Kornai. The algebra of lexical semantics. In Christian Ebert, Gerhard Jäger, and Jens Michaelis, editors, *Proceedings of the 11th Mathematics of Language Workshop*, LNAI 6149, pages 174–199. Springer, 2010.
6. András Kornai. Eliminating ditransitives. In Ph. de Groote and M-J Nederhof, editors, *Revised and Selected Papers from the 15th and 16th Formal Grammar Conferences*, LNCS 7395, pages 243–261. Springer, 2012.
7. András Kornai, Judit Ács, Márton Makrai, Dávid Márk Nemeskey, Katalin Pajkossy, and Gábor Recski. Competence in lexical semantics. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 165–175, Denver, Colorado, June 2015. Association for Computational Linguistics.
8. András Kornai and Márton Makrai. A 4lang fogalmi szótár. In Attila Tanács and Veronika Vincze, editors, *IX. Magyar Számitógépes Nyelvészeti Konferencia*, pages 62–70, 2013.

9. Márton Miháltz. *Semantic resources and their applications in Hungarian natural language processing*. PhD thesis, Pázmány Péter Catholic University, 2010.
10. Ferenc Pusztai, editor. *Magyar értelmező kéziszótár*. Akadémiai Kiadó, 2003.
11. Gábor Recski. Hungarian noun phrase extraction using rule-based and hybrid methods. *Acta Cybernetica*, 21:461–479, 2014.
12. Gábor Recski. *Computational methods in semantics*. PhD thesis, Eötvös Loránd University, Budapest, 2016.
13. Viktor Trón, György Gyepesi, Péter Halácsy, András Kornai, László Németh, and Dániel Varga. Hunmorph: open source word analysis. In Martin Jansche, editor, *Proceedings of the ACL 2005 Software Workshop*, pages 77–85. ACL, Ann Arbor, 2005.
14. Veronika Vincze, Dóra Szauter, Attila Almási, György Móra, Zoltán Alexin, and János Csirik. Hungarian dependency treebank. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10)*, 2010.
15. János Zsibrita, Veronika Vincze, and Richárd Farkas. magyarlanc: A toolkit for morphological and dependency parsing of Hungarian. In *Proceedings of RANLP*, pages 763–771, 2013.