Rule-based extraction of English verb collocates from a dependency-parsed corpus

Silvie Cinková, Martin Holub, Ema Krejčová, Lenka Smejkalová
Charles University in Prague, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské nám. 25
118 00 Praha 1 Czech Republic
{cinkova,holub,krejcova}@ufal.mff.cuni.cz

Abstract

We report on a rule-based procedure of extracting and labeling English verb collocates from a dependency-parsed corpus. Instead of relying on the syntactic labels provided by the parser, we use a simple topological sequence that we fill with the extracted collocates in a prescribed order. A more accurate syntactic labeling will be obtained from the topological fields by comparison of corresponding collocate positions across the most common syntactic alternations. So far, we have extracted and labeled verb forms and predicate complements according to their morphosyntactic structure. In the next future, we will provide the syntactic labeling of the complements.

1 Introduction

We commonly perceive the verb as the center of the sentence. By using a verb to interconnect nouns that we want to refer to, we make them participants in an event. About one half of the entries in the Oxford English Dictionary are noun entries whereas only one seventh of the entries are verb entries 'OED Online. March 2013. Oxford University Press.' <http://www.oed.com/view/Entry/113184 (accessed April 11, 2013).>.. The percentage rates vary slightly in European languages, but there are always fewer verbs than nouns. Isn't it astonishing that we need relatively few verbs to describe all events we want to comment on? No wonder that verbs are used in a variety of different argument structure patterns and with very different collocates. Let us consider a common verb such as *give*:

(1) The wooden chair gave a frightened squeak.
(2) Mom gave me a cookie.
(3) The results gave them quite a shock.
(4) Joanna gave her a disgusted look.
(5) The audience gave him the raspberry.
(6) Eventually, they had to give up.

We intuitively perceive combinations of syntactic patterns and collocates as different word senses, but in fact there is no such thing as a universal set of senses for each verb: the number of word senses in a dictionary, as well as their definitions, is based on individual judgments of the lexicographer and regulated by the editorial policy of a particular dictionary. Thus, making a dictionary is by no means objective modeling of the meaning of a lexical item, and lexicographers have never even claimed that ambition: "I don't believe in word senses", goes a thought-provoking quote of Sue Atkins, a respectable practitioner and pioneer of modern lexicography. On the other hand, the concept of semantic grouping of some sort is deeply anchored in our linguistic intuition, and we can hardly think of a different starting point, should the lexical description be intelligible. The question remains what sort of grouping should be applied and what the perception of a meaning shift is based on, which is where the lexicographical policies differ with respect to the intended use of each particular lexicon.

Already in 1997, Adam Kilgarriff, a visionary of computational lexicography, used Atkins' remark for the title of his influential paper (Kilgarriff 1997) to argue that "the corpus citations, not the word senses, are the basic objects in the ontology. The corpus citations will be clustered into senses according to the purposes of whoever or

Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013), pages 60–67,
Prague, August 27–30, 2013.

whatever does the clustering. In the absence of such purposes, word senses do not exist." (p. 91).

2 Related Work

Many important authors have elaborated the intuitive observation that what is perceived as lexical meaning arises from an interplay of syntactic and semantic features in the context of each actual use of the lexical unit under examination (e.g. Firth 1957; Sinclair 1991; Hoey 2005; Hanks 2013, Levin 1993, Gross 1994; Fillmore 2006; Palmer et al., 2005; Hudson 2006). They all describe, with a varying degree of formalization, the behavior of individual words; sometimes by their own or a collective introspection (Firth, Levin, Gross), sometimes based on the manual findings in a text corpus (Sinclair, Hanks, Fillmore and Palmer). However, their approach results again in man-made lexicons. These are necessarily biased towards the data on which they are based, and the word senses are hard-wired.

On the other hand, a number of collocation analysis tools are available; e.g. Sketch Engine (Ki lgarriff et al. 2004) and DeepDict (Bick 2009). While the Sketch Engine uses linear search, DeepDict extracts collocates from dependency trees.

3 Five-slot forms and Canonical Sequence

We have a working hypothesis regarding verbs that their selectional preferences can be modeled by a statistical analysis of the surface-syntax structures of their uses and the distributional similarity of the nouns that occur as their complements and perhaps in some other positions. Since we do not want to confine ourselves to verb arguments and adjuncts, we refer to them as *verb collocates.* Under *collocations* we do not only understand idiosyncratic combinations of content words, but, more broadly, significantly co-occuring combinations of a given lexical verb with other content words as well as with the grammatical patterns surrounding it. Our conception of collocation overlaps e.g. with Hoey's term *textual colligation* (Hoey 2005), p. 52).

Unlike Sketch Engine and DeepDict, we want to regard the significance of each collocate noun with respect to its syntactic function in a given *clause template (CLT).* We have therefore selected the most common clause structures and are recording them as conditions in dependency trees. We formulated the templates with the Prague Markup Query Language (Pajas and Štěpánek 2009). The clause templates are in fact corpus queries. They highlight the verb under examination (*target verb*) and the present collocates. The collocates are numbered according to the position in the linear order of a statement clause in active verb voice with a neutral word order (no topicalizations, no verb-subject inversions), to which we refer as *Canonical Sequence* (Fig 1). Besides, they are marked with a letter. For instance, number 3 in the label `a3` encodes the information that the particular node would occur in the third position in a regular statement clause. The letter `a` indicates that it occurred in the first position in the given template. That would be the case of the subject of a passive clause.

1	2	3	4	5
Agent	TV	OBJ1/SC	OBJ2/OC	Prep+NP/ADV /RP/NPquant

Figure 1: Canonical Sequence of sentence elements

We have introduced the character-digit pair to consider the similarity of the lexical population of positions across different clause templates. We hope to be able to e.g. densify our data by neglecting passivization or the *to*-alternation.

The Canonical Sequence is one of the Five-slot forms that cover the most common clause structures. A target verb that matches a clause template will obtain the label of that template and the matching collocates will obtain the collocate labels.

The first position in the Canonical Sequence belongs to the Agent. The second position occupies the target verb (finite and in the active voice). The third position belongs to the first noun phrase in the row or to a verb clause. This is also the right position for an adjective phrase that is not preceded by a noun phrase right of the target verb position. The fourth position hosts the second of two noun phrases, whenever they occur in the clause, or an adjective phrase preceded by a noun phrase right of the target verb, or a verb clause. The fifth position is meant for prepositional noun phrases, adverbs, verb particles (tagged as `RP`), and non-prepositional noun phrases identified as adverbials of time or quantity. We use a

heuristic rule to identify these noun phrases. Our rule lists the most common lemmas of time/quantity information, such as names of weekdays, months and seasons, and quantity measures. Of course we do our best to avoid heuristic rules; this is an exception in our system.

As we are not able to differentiate between a prepositional object, obligatory modifier and a free adjunct, the position of the prepositional phrase is always optional in the query. Fig. 2 shows the visualizations of several clause templates.

4 Template sets

To find the clause templates in the corpus, we need the following information on the target verb:

- verb finiteness
- verb voice
- type of clause that it governs

We also need to find all words that can act as nouns and thus fill a position in the five-slot form, including their possible prepositions. Eventually, we want to identify adjectives and adverbs.

With these requirements we hit just between the part-of-speech tagging (Santorini 1990) and the syntactic labels provided by the parsers available to us. For instance a verb form marked as VBN (past participle) can represent a finite active verb form `(has/had read)`, a finite passive verb `form (is/was/is being/was being/has been/had been read)`, or an infinite passive verbform `to be read, to have been read`, etc., or the future tense `will be read` or `will have read`. The syntactic labels functions in their turn describe mostly the verb's syntactic relations to their governing syntactic elements. Therefore we have been creating our own labels. We have introduced a set of labels for the verbs, which we call Verb

Form Templates. All verb forms are captured in approximately 40 templates. We keep a separate class for verbs combined with modal and a few auxiliary verbs, such as `be going to, used to and have (got) to`.

We have also sketched about 30 collocate templates *(Verb Argument Templates, VAT)*.

CLT_008.

1	2	3	4	?5+ (optional)
a	b	c	d	e
NP not WH_coref	TV act, not governed by NP	NP	ADJP	PP PP ING
Agent		Obj_1	Object Complement	
John	calls	Mary	dull	with pleasure

CLT_009.

1	2	3	4	?5+ (optional)
a	b	c		d
NP not WH_coref	TV act, not governed by NP	Quote, that-clause, subclause without subordinator, wh-clause	-	PP PP ING
Agent				
John	says	that it is raining	-	with pleasure
John	says	it is raining	-	
John	says:	"It is raining"	-	
John	says	what we want to hear		

Figure 2: Visualization of two clause templates

We distinguish the following VAT:

- noun phrase (including numbers, determiners and personal pronouns)
- possessive noun phrase (Saxon genitive and possessive forms of personal pronouns)
- relative expression in a wh-clause or a relative clause, antecedent identification added
- relative expression in a pseudocleft clause, antecedent identification added
- expletive `it`.

Fig. 3 shows the VFT of a lexical verb governed by a modal verb.

5 Overcoming parser errors with Five-slot forms

We have introduced the Five-Slot forms to overcome the weaknesses of the automatic parsing.
In traditional grammars, such as (Quirk et al. 2004), verbs are neatly grouped according to the number of objects. The problem is that the parsers available, unlike human experts, are not able to deal with structural ambiguities. Therefore they often give random results in syntactic labeling. For instance, a head noun following an active verb is mostly classified as an object, even when it is a subject complement or a temporal adverbial (e.g. `He arrived Sunday`).

Making use of the fixed word order in English, we use the structured parser output to provide relevant collocates with positional

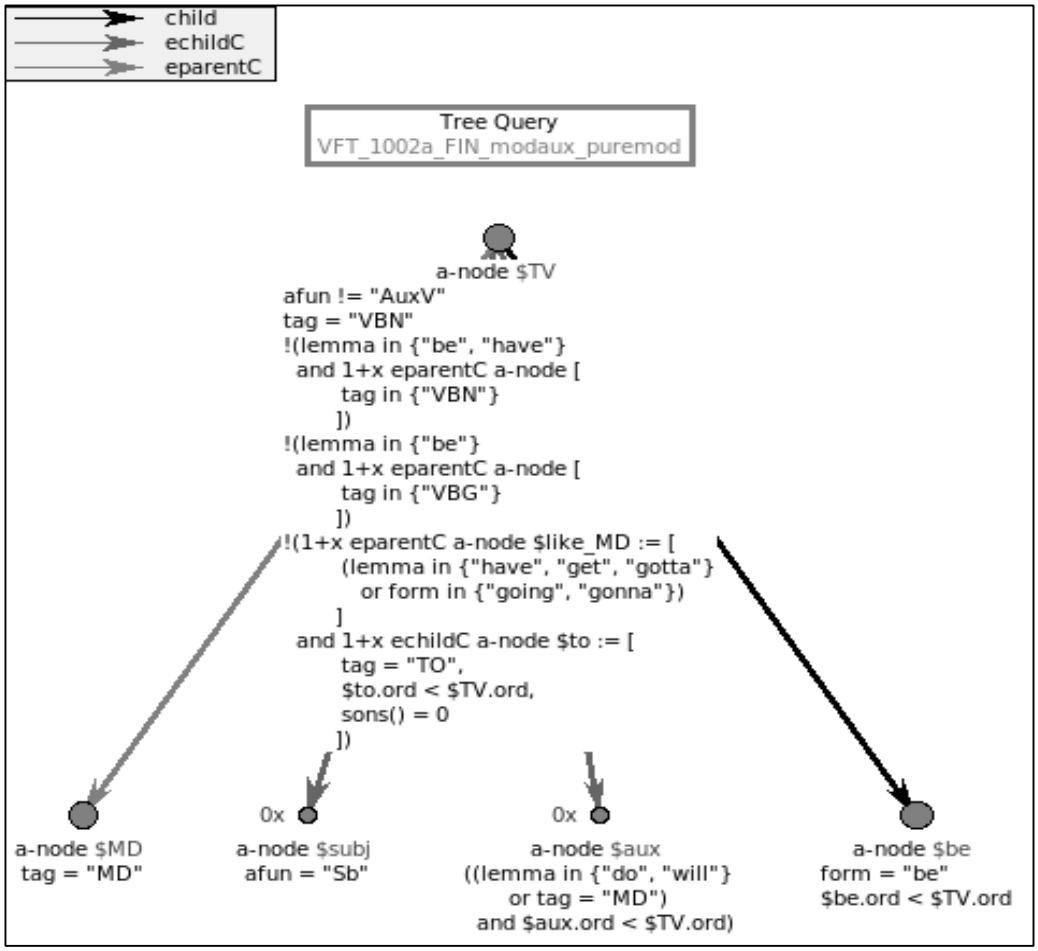

Figure 3: Verb form template of a lexical verb governed by a modal verb (regarded as a finite form of the lexical verb

labels. So we can bypass the errors in the syntactic analysis and are still able to compute occurrences of nouns.

We also have to deal with the fact that the parser is forced to overdo the semantic interpretation of a sentence to achieve a fully structured tree (and it often does it in a way a human never would). One interesting example of this sort is the to-infinitive following a noun phrase that follows the target verb.

There are three structure options for a number of structurally ambiguous cases. Cf.:

```
(7)   ... persuaded the
visitor to leave/... shut the
door to hide
(8)   ... hated the woman to
go
(9)   ... became the first
player to score
```

In the first case, `persuade` and `shut` have two complements each: `the visitor` and `leave` (`persuaded the visitor that the visitor should leave`, `shut the door in order to hide`). Note that we ignore the labeling. In the first case, the infinitive clause with `leave` is a regular argument, while in the second clause `hide` is a free adjunct, that is, a purpose clause. In the second example, `hate` has only one argument – `go`, whereas `the woman` is the subject of `go` (`hated that the woman went`). In the third case, the only argument of `become` is `player`, which is modified by the attributive infinitive `to score` (`became the first player who scored`).

The statistical parser has learned about these three structures. It even produces correct results in verbs that occurred in the training data frequently enough, such as `expect` and `hate`. Nevertheless, the resulting structure is completely unpredictable in most verbs, and, even worse, the resulting structures are inconsistent in different occurrences of the same verb. This inevitably causes a strong bias in the collocation statistics.

We had to bypass this problem by querying all three structures in each verb occurrence and merging the results (Figures 4,5 and 6).

The parser provides labeling of syntax elements, but is often grossly wrong. For instance, prepositional phrases are typically labeled as adverbials: the prepositional objects of the verbs `rely` and `indulge` would be labeled as adverbials. On the other hand, non-prepositional adverbials, such as `last year` or `two miles` would be labeled as objects. Nor is the (i.e. any) parser particularly good at making a difference between the direct object of a bitransitive verb and the object complement expressed by a noun, and therefore we cannot retrieve them as two separate categories. Cf.:

```
John bought me a book.
John called me an idiot.
```

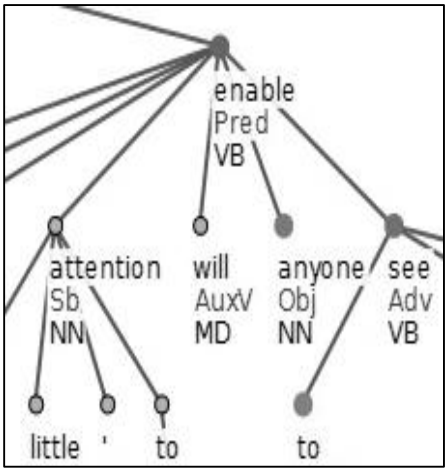

Figure 4: Control

All these structures are simple to recognize for a human, since the human uses their lexical knowledge to resolve the structural ambiguity, and they also make the human conceptualize the events in very different ways. As the parser is not able to tell them apart correctly, we have to merge these categories and try to separate them later.

Also, the parser very often picks the first verb form in the sentence to be the main predicate, even when it is a participial phrase and/or is introduced by a subordinator. The misindentification of the main predicate affects the argument recognition not just in the first predicate, but also in the second and further in an unpredictable way.

The issues mentioned above are both homogeneous and frequent enough to be detected by manual inspection. Their frequency slightly varies with respect to different verb lemmas, whose context we examine.

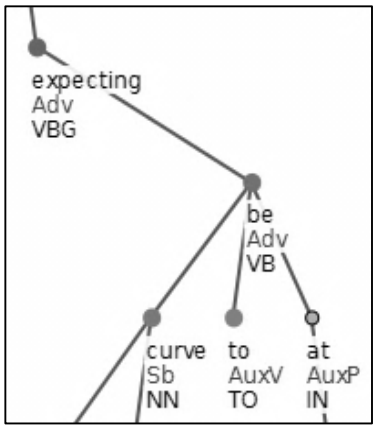

Figure 5: Raising

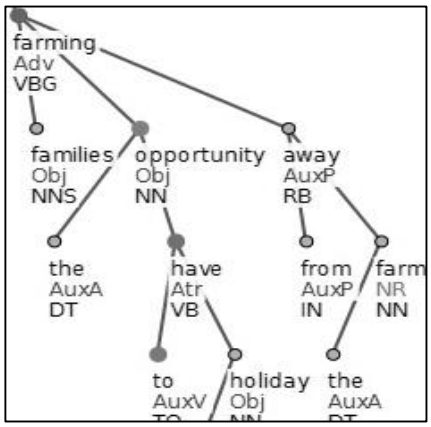

Figure 6: Attributive infinitive

6 The corpus

We perform the labeling on BNC50, a 50-million-token subset of BNC used in the Pattern Dictionary of English Verbs (Hanks and Pustejovsky 2005).

We have chosen to work with a syntactically parsed corpus, since our earlier (unpublished) preliminary study showed that collocation extraction has better recall on a parsed corpus than on a plain text corpus, with all tested parsers (McDonald, Lerman, and Pereira 2006); (de Marneffe, MacCartney, and Manning 2006) giving similar results. We use our in-house NLP infrastructure tool Treex (Žabokrtský 2011) to transform the outputs of different parsers into a uniform dependency representation in the annotation style of the Prague Dependency Treebank family (Hajič, 2004). The rules are tailored to the PDT-scheme and thus not specific to one of the original parsers, which means that they are applicable to any parser output processed by Treex. We have been using the MST parser.

At the beginning, we had to decide which annotation layer we should write our rules for, since Treex offers two different PDT-style linguistic representations. The PDT-like corpora have two different but interlinked syntax annotation layers: the analytical (i.e. surface syntax) and the tectogrammatical layer (i.e. underlying syntax with semantic labeling and coreference).

At first, the tectogrammatical layer was the apparent favorite, since it offers a straightforward extraction of verb arguments:

- It abstracts from regular syntactic alternations, such as passivization and reciprocity. Both active and passive clauses have the same tree representations and the same semantic labels (functors) of the arguments.
- Semantic labels (functors) distinguish arguments from adjuncts and give a semantic classification of the adjuncts.
- Missing verb arguments are substituted with labeled substitute nodes according to a valency lexicon. This feature compensates not only for textual ellipsis, but, more importantly, also for grammatical ellipsis; e.g. artificial subjects of controlled infinitives are inserted.
- Anaphora are resolved even in the grammatical coreference. For instance, the artificial subject node of a controlled infinitive contains a reference to a real node that would be the subject of the infinitive, if it were not controlled by another verb.

For all these enhancements, the tectogrammatical representation would have been our first choice, at least considering the manually annotated data (Hajič et al. 2012).

However, the automatic English tectogrammatical annotation is very unreliable, compared to the manual standard: the semantic labels are often wrong, and hardly any missing nodes are reconstructed. Besides, the tectogrammatical representation takes away word forms and auxiliary words. To retrieve the auxiliary words or word forms, one has to refer to the lower (analytical) layer, increasing

the complexity of the corpus queries. Hence we preferred the analytical layer.

We had to reflect and compensate for systematic errors in the analytical parse. Most of them had propagated already from the constituency parser. By our manual estimation, all parsers known to us have similar problems, so we could not just switch the constituency parser at the beginning of the process pipeline to avoid these problems.

7 Future work

At the moment we are implementing the templates to be able to evaluate them. We have been testing the corpus queries continuously and revising the templates accordingly. However, we still have to do a quantitative evaluation. In the future we want to use the template labels as features in a model of selectional preferences of verbs.

8 Discussion

There are at least two well-working tools for collocation sorting for English: the Sketch Engine (Kilgarriff et al. 2004) and DeepDict (Bick 2009). However, the sorting we intend to do goes slightly beyond what they provide. To the best of our knowledge, neither the Sketch Engine nor DeepDict consider the collocates across different syntactic alternations. We believe that the quality of our syntactic labeling will rapidly increase with the growing data, so that collocate lists based e.g. on the English GigaWord (Graff and Cieri 2003) or CzEng (Bojar et al. 2012) will reflect the mapping of collocate positions between clause types well. A preliminary manual evaluation of the VFT and VAT annotation of several hundred sentences revealed minor inconsistencies, which are being fixed at the moment, but the labels proved generally appropriate. Inaproppriately labeled instances almost always occurred in trees with substantial parsing errors.

9 Conclusion

In this still initial investigation we have been labeling verb occurrences in a dependency treebank with clause types and their complements with numbers of positions they would occupy in the linear scheme of a finite statement clause with neutral word order. We have been pursuing this exercise because we believe that clause types and complement position labels will represent a useful set of features for statistical modeling of the selectional preferences of English nouns and verbs.

10 Acknowledgements

This work was supported by the Czech Science Foundation (grant No GAP103/12/ G084).

References

Eckhard Bick. 2009. DeepDict - A Graphical Corpus-based Dictionary of Word Relations. In *Proceedings of NODALIDA* , 4:pp. 268–271. Tartu: Tartu University Library.

Ondřej Bojar , Zdeněk Žabokrtský, Ondřej Dušek, Petra Galuščáková, Martin Majliš, David Mareček, Jiří Maršík, Michal Novák, Martin Popel, and Aleš Tamchyna. 2012. The Joy of Parallelism with CzEng 1.0. In LREC Proceedings, 3921–3928. Istanbul, Turkey: European Language Resources Association.

Marie-Catherine De Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. 'Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of the IEEE / ACL 2006 Workshop on Spoken Language Technology* Stanford University.

Charles J Fillmore. 2006. Frame Semantics. In *Encyclopedia of Language & Linguistics*, 613–620. Oxford: Elsevier.

John Rupert Firth. 1957. *Papers in Linguistics 1934–1951*. London: Oxford University Press.

David Graff, and Christopher Cieri. 2003. English Gigaword. *Linguistic Data Consortium, Philadelphia*.

Maurice Gross. 1994. The Lexicon Grammar of a Language Application to French. In *Encyclopedia of Language and Linguistics*, R. E. Ashe, 2195–2205.

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Ondřej Bojar, Silvie Cinková, Eva Fučíková, et al. 2012. Announcing Prague Czech-English Dependency Treebank 2.0. In LREC Proceedings, 3153–3160. Istanbul, Turkey: European Language Resources Association.

Patrick W. Hanks. 2013. *Lexical Analysis: Norms and Exploitations*. University Press Group Limited.

Patrick Hanks, and James Pustejovsky. 2005. A Pattern Dictionary for Natural Language Processing. *Revue Francaise de Linguistique Appliquée* 10 (2).

Michael Hoey. 2005. *Lexical Priming: A New Theory of Words and Language*. Routledge.
Richard Hudson. 2006. Word Grammar. In *Encyclopedia of Language & Linguistics*, 633–642. Oxford: Elsevier.
Adam Kilgarriff. 1997. "I Don't Believe in Word Senses". *Computers and the Humanities* 31 (2): 91–113.
Adam Kilgarriff, Pavel Rychly, Pavel Smrz, and David Tugwell. 2004. The Sketch Engine. In .
Beth Levin. 1993. *English Verb Classes and Alternations*. University of Chicago Press.
Ryan McDonald , Kevin Lerman, and Fernando Pereira. 2006. Multilingual Dependency Analysis with a Two-stage Discriminative Parser. In , 216–220. Association for Computational Linguistics.
OED Online. March 2013. Oxford University Press.
Petr Pajas, and Jan Štěpánek. 2009. System for Querying Syntactically Annotated Corpora. In , 33–36.
Martha Palmer, Dan Gildea, and Paul Kingsbury. 2005. The Proposition Bank: A Corpus Annotated with Semantic Roles. *Computational Linguistics Journal* 31 (1).
Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 2004. *A Comprehensive Grammar of the English Language*. Longman.
Beatrice Santorini. 1990. Part-of-Speech Tagging Guidelines for the Penn Treebank Project. *University of Pennsylvania 3rd Revision 2nd Printing* (MS-CIS-90-47): 33.
John Sinclair. 1991. *Corpus, Concordance, Collocation*. Oxford University Press.
Zdeněk Žabokrtský. 2011. Treex – an Open-source Framework for Natural Language Processing. In Information Technologies – Applications and Theory, 7–14.