RASLAN 2013
Recent Advances in Slavonic Natural Language Processing

A. Horák, P. Rychlý (Eds.)

RASLAN 2013

Recent Advances in Slavonic Natural Language Processing

Seventh Workshop on Recent Advances in Slavonic Natural Language Processing, RASLAN 2013
Karlova Studánka, Czech Republic, December 6–8, 2013
Proceedings

Tribun EU
2013

Proceedings Editors

Aleš Horák
Faculty of Informatics, Masaryk University
Department of Information Technologies
Botanická 68a
CZ-602 00 Brno, Czech Republic
Email: hales@fi.muni.cz

Pavel Rychlý
Faculty of Informatics, Masaryk University
Department of Information Technologies
Botanická 68a
CZ-602 00 Brno, Czech Republic
Email: pary@fi.muni.cz

ISBN 978-80-263-0520-0

Preface

This volume contains the Proceedings of the Seventh Workshop on Recent Advances in Slavonic Natural Language Processing (RASLAN 2013) held on December 6–8, 2013 in Karlova Studánka, Czech Republic.

The RASLAN Workshop is an event dedicated to the exchange of information between research teams working on the projects of computer processing of Slavonic languages and related areas going on in the NLP Centre at the Faculty of Informatics, Masaryk University, Brno. RASLAN is focused on theoretical as well as technical aspects of the project work, on presentations of verified methods together with descriptions of development trends. The workshop also serves as a place for discussions about new ideas. The intention is to have it as a forum for presentation and discussion of the latest developments in the field of language engineering, especially for undergraduates and postgraduates affiliated to the NLP Centre at FI MU. We also have to mention the cooperation with the Dept. of Computer Science FEI, VŠB Technical University Ostrava.

Topics of the Workshop cover a wide range of subfields from the area of artificial intelligence and natural language processing including (but not limited to):

* text corpora and tagging
* syntactic analysis
* sense disambiguation
* machine translation, computer lexicography
* semantic networks and ontologies
* semantic web
* knowledge representation
* logical analysis of natural language
* applied systems and software for NLP

RASLAN 2013 offers a rich program of presentations, short talks, technical papers and mainly discussions. A total of 12 papers were accepted, contributed altogether by 19 authors. Our thanks go to the Program Committee members and we would also like to express our appreciation to all the members of the Organizing Committee for their tireless efforts in organizing the Workshop and ensuring its smooth running. In particular, we would like to mention the work of Aleš Horák, Pavel Rychlý and Lucia Kocincová. The TEXpertise of Adam Rambousek (based on LATEX macros prepared by Petr Sojka) resulted in the extremely speedy and efficient production of the volume which you are now holding in your hands. Last but not least, the cooperation of Tribun EU as both publisher and printer of these proceedings is gratefully acknowledged.

Brno, December 2013 Karel Pala

Table of Contents

Part I

Syntax, Morphology and Lexicon

Preparing VerbaLex Printed Edition

Dana Hlaváčková, Aleš Horák, and Karel Pala

NLP Centre
Faculty of Informatics
Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
{hlavack, hales, pala}@fi.muni.cz

Abstract. In the paper, we present the current state of the development of the Czech valency dictionary called VerbaLex. It contains a list of the most frequented Czech verbs and their valency frames in the form of the complex valency frames. VerbaLex includes information about verb case and adverbial links (morphosyntactic properties) and senses captured by an inventory of two-layer semantic roles that characterize the semantics of the verb arguments.
We also present the motivation and history of the design of the complex valency frames and the VerbaLex lexicon. One of the main aims here is the support of computer analysis of Czech, thus machine-readable features of the lexicon are emphasized since the beginning. Presently, we can refer to VerbaLex electronic version with more than 10 thousand verb lemmata, as well as to its printed form with a selected subset of the most frequent verbs. The full electronic form is available on-line after registration for academic and non-commercial purposes.

Key words: verb, verb frame, verb valency, VerbaLex, WordNet, VerbNet

1 Introduction

We present to readers a new Czech valency dictionary, which is being developed in the electronic form since 2006 with the title *VerbaLex*. The dictionary contains a list of the most frequented Czech verbs and their valency frames including information about their case and adverbial links (morphosyntactic properties) and senses captured by an inventory of the complex semantic roles that characterize the semantics of the verb arguments.

The dictionary is intended for an expert public, linguists, translators, researchers in the NLP area and anybody who is interested in a deeper understanding of Czech as a mother tongue. It can also serve (as a resource) in computer applications directed to information search, summarization and possibly Machine Translation. In this context we would like to mention another valency dictionary of Czech verbs, named *Vallex*, which was prepared by the group of authors from the Institute of Formal and Applied Linguistics, Charles University [1, 6460 lex. units].

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2013, pp. 3–11, 2013.

A question could be raised rightly why we consider it useful to develop another valency dictionary of Czech? A following metaphor offers an answer – a language, in our case Czech, can be considered by researchers as a fortress they are trying to conquer from various angles. Thus, it is natural to approach verb valencies in Czech from different standpoints with the purpose to reach a deeper understanding of their nature. *VerbaLex* offers a different view of the Czech verb valencies than *Vallex* – the main difference consists in the conception of the semantic roles characterizing the meaning of the verb arguments (in *Vallex* named actants).

In other words, the difference lies in the approach to semantic properties of the Czech verbs – we are convinced that different solutions can be chosen and all can be reasonably justified – in practice we usually prefer the one which proves to be more fitting for the particular application. We say more about the differences between actants in *Vallex* and semantic roles in *VerbaLex* below, see especially 3.1.

2 The VerbaLex Valency Lexicon

VerbaLex is an electronic lexical database comprising verb valency frames – it has been developed in the NLP Centre, Faculty of Informatics, Masaryk University in Brno (*FI MU*) during 2006–2013. It is a result of the work, which partly belongs to the area of linguistics and partly to the field of Natural Language Processing (NLP). During the development of *VerbaLex*, we have been using various corpus resources and electronic tools, which made it possible to observe the behaviour of the verbs in their natural contexts. The main part of the database has been compiled by the annotators who relied on their linguistic competence, followed given instructions and using the accessible software tools created what is called the *basic* and *complex valency frames*. In this way the issue of the Czech verb valencies could have been captured in their complexity as much as possible.

The verb valency is understood in the database as a semantically given ability of the verb allowing it to combine with other words – the verbs are described from this point of view together with their complements both on the left and right side. Thus valency frames include two kinds of information: the *morphosyntactic* and *semantic* one. Our effort was to capture as many Czech verbs as possible, presently the *VerbaLex* comprises approximately 10 500 verb lemmata.

When compiling *VerbaLex* we have used some existing resources, in the first place the *Valenční slovník českých sloves* (*Valency Dictionary of Czech Verbs*) with working name *BRIEF* [2].

Our motivation has been an effort for a deeper understanding of the semantics of Czech verbs and their arguments and creation of the new data resources, which for Czech exist only in part. In comparison with the traditional approaches, for instance [3], we have used methods and techniques, particularly semantic networks and ontologies, which do not appear in existing Czech

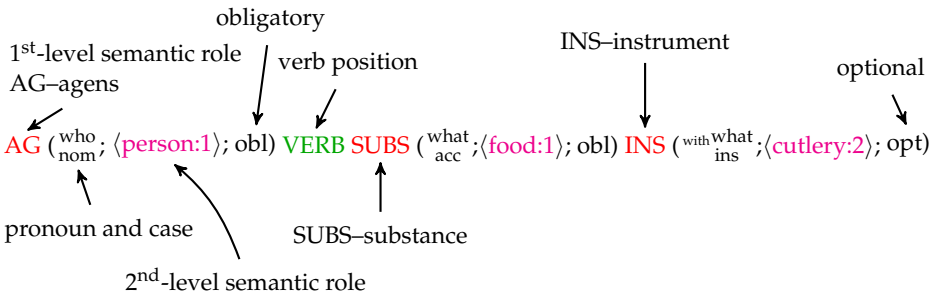


Fig. 1: Basic valency frame

grammars at all. The obtained results can be then exploited in the field of NLP, since *VerbaLex* captures semantic relations. Thus it can be used in various applications such as the intelligent search on the Web, word sense disambiguation, information extraction or text understanding [4,5].

2.1 VerbaLex Structure

After starting exploration of the suitable format for verb valencies in 2006, the structure of *VerbaLex* has consolidated in the form presented below. The database displays some basic features, in which it differs from similar dictionaries. The form of the *complex valency frames* allow us to capture the relevant information about a verb and its complements. The valency frames are assigned to individual verb senses (grouped in synonymical sets or *synsets*, see Section 2.4) and not only to individual lemmata (many synonyms share the same valency). To label the meanings of the verb complements the system of the two-level semantic roles has been used.

The basic valency frames (see Figure 1), which represent the core of *VerbaLex*, constitute the notation of the verb valency on the morphosyntactic and semantic level. The center of the frame is a marked verb position, its valency complements on the morphological level are represented by the pronominal expressions together with the respective case numbers. The notation follows the canonical word-order: "the complement on the left side – verb – the complements on the right side." On the semantic level the verb arguments are labeled by the two-level semantic roles, which specify the semantic environment of the verb precisely. The frame contains additional information about obligatoriness and optionality of the valency complements. The basic valency frame is always related to a *subsynset*, which is a subset of the defined synonymical set.

The basic frame is a part of the complex valency frame (see Figure 2), which is always related to the one synonymical set only. Apart from the frame there is additional information which includes verb sense, aspect (see Section 2.3) and verb semantic class (see Section 3.2). For each verb its ability to form passive voice is recorded, thus it is possible to generate lists of the transitive and intransitive verbs from the database. The important feature of the Czech verbs

jíst:1 (impf), požít:2(pf), požívat:2(impf) (eat:1)
definition: *přijímat potravu* (take in solid food)
class: *eat-39.1*
passive: *yes*

jíst:1 (eat:1) ≈
-frame: **AG**($^{\text{who}}_{\text{nom}}$;⟨person:1⟩;obl) **VERB** **SUBS**($^{\text{what}}_{\text{acc}}$;⟨food:1⟩;obl)
INS($^{\text{with what}}_{\text{ins}}$;⟨cutlery:2⟩;opt)
-example: *synovec jedl zmrzlinu (impf)* (the nephew ate an ice cream)
-example: *dcera jí polévku lžící (impf)* (the daughter eats a soop with a spoon)
-use: prim
-reflexivity: no

Fig. 2: Complex valency frame

is their obligatory or optional reflexivity – we provide information about three basic types of reflexivity: proper reflexives (reflexiva tantum), i.e. verbs with obligatory reflexive particle *se*, e.g. *bát se* (*to be afraid*). Further, object reflexivity is marked when the pronoun *se* or *si* replaces object of the action (*mýt se*, *čistit si* (*wash yourself*, *clean yourself*)) as well as reciprocity, mutual activity of the two subjects (*znát se*, *milovat se* (*to know each other*, *to love each other*)). Here the verb lemma is not given with the reflexive pronoun. For the rest of the verb lemmata, we mark the fact that they have or do not have the reflexive form in the respective sense and characterize it as another (not specified) type of the reflexivity. The next relevant information is related to the behaviour of the verb in a particular context: we mark their primary (basic) usage in contrast to the figurative (metaphorical) meaning. In some cases displaying a higher frequency in corpora we also indicate the idiomatic usage.

2.2 Verb List Selection

The choice of the verb lemmata contained in the database *VerbaLex* has been based mainly on *Slovník spisovné češtiny* (Dictionary of Literary Czech, SSČ [6]) and *Slovník spisovného jazyka českého* (Dictionary of Literary Czech Language, SSJČ [7]). Moreover, the lemma selection stylistic features and frequencies of the particular verbs in the corpora *SYN2000* [8] and *ALL* (*NLP Centre FI MU*) have been considered. As a basic data resource, the *Valenční slovník českých sloves* [2] has served, which describes right side valency complements (without information about their meaning), for 15 079 Czech verbs. In *VerbaLex*, we have stored only verbs belonging to the literary Czech, where some of them can eventually have the emotional colouring. *VerbaLex* does not include verbs from colloquial Czech and dialects. We also have left aside verbs that are strongly bookish, archaic or rarely used. However, we have taken into consideration the cases when a verb is marked in a dictionary as bookish or rarely used (e.g. *pravit* (*say*, bookish form)), but they show high frequency in the corpora (verb forms

like *pravit* – shows 29 397 occurrences in the corpus *ALL*), in such cases we have respected the frequencies found in corpora.

The verb lemma is always one-word, in the case of the proper reflexives (reflexiva tantum) with the reflexive particles *se*, *si*. The database does not contain negated forms of the verb lemmata, we work with the assumption that the valency frames of the negated verbs remain unchanged (except for cases with the negated genitive) and the negated forms of lemmata is possible to derive automatically. In *VerbaLex*, there is a formal way how to handle to what we call variant lemmata. In such cases the verb forms differ only in the vowel alternation, otherwise all their characteristics remain the same (e.g. *muset / musit* (*must*), *bydlit / bydlet* (*live*), *červavět / červivět* (*become wormy*)).

2.3 Verb Aspect

In the complex valency frame notation also other important information about verbs has been captured. In the first place, it is a formal notation of the verb aspect as related to the respective verb sense. The aspect identification is primarily based on the information in the dictionaries *SSČ* a *SSJČ*. If a given verb sense is valid for both its aspect forms, the verb in the first position is marked as perfective (*pf*) together with the number of the sense followed in the brackets by its imperfective form (*impf*), which automatically takes over the sense number from the perfective and it is not necessary to indicate it again. The verbs with two aspects are denoted as biaspectual *biasp*. Iterative verb forms are not stored in the database. Their forms are derived very regularly and can be automatically added from the existing morphological database to the *VerbaLex* at any time.

If a verb sense is valid for just one aspect form or a verb is one-aspectual, we mark the perfective or imperfective verb form independently with its own sense number. In the case of the perfective verbs derived by prefixation from the imperfective verb, we consider as aspect pairs only the cases, which are explicitly marked in the dictionaries *SSČ* a *SSJČ* (prefixation with the aspect prefix only, e.g. *uvařit* (*to cook*), *učesat* (*to comb*)). Other prefixed verbs are marked either independently or in the pair with the respective secondary imperfective.

2.4 Synonymy and Verb Senses

As we have mentioned above, verbs in *VerbaLex* are organized in synonymical sets (*synsets*). In synsets, each verb lemma (and its variants) are marked with ordinal number denoting their sense.[1] This denotation directly corresponds to the numbering in the Czech WordNet, see Section 3. The appropriate synonyms have been chosen and verified in *Slovník českých synonym* (Dictionary of Czech Synonyms, SČS [9]). Each synset is accompanied with a short definition of its meaning. The definitions (with necessary modifications) are formulated on the basis of the lexicographic definitions in *SSČ* and *SSJČ*. In the specification

[1] The tuple "lemma:sense" is often called a *literal*.

of particular verb senses, we often cannot always use SSČ and SSJČ directly. Their approach differs in many cases and they state different number of verb senses. These dictionaries are also not sufficiently up-to-date source of current language. In case of a verb sense, which was not found in the dictionaries, the verb occurrences in new contexts were verified on corpus data (*SYN2000*, *ALL*). If the number of the verb sense occurrences reached adequate frequency, the sense was added to *VerbaLex* with new sense number. Obsolete and rare cases from the dictionaries are not used in *VerbaLex* at all. In accordance with the *WordNet* structure,[2] the verb sense determination is often more fine-grained than in usual Czech dictionaries.

3 Semantic Description Layer – VerbaLex and WordNet

One of the main differences of *VerbaLex* when compared to *Vallex* is the narrow connection of *VerbaLex* to the *WordNet* semantic network (*Princeton WordNet*, *PWN* [10]) since the very beginning. During the *BalkaNet* EU project in 2002, the Czech WordNet (*CzWn*) structure was supplemented with basic valency frames including semantic roles. According to the *PWN* structure, the frames were linked to whole synsets instead of individual verb lemmata. For the same reason, the frames were divided according to particular verb senses. This approach was then adopted in the *VerbaLex* preparation procedures.

The verb synonymy is here understood in a broader sense than usual. The synset participants are often *near synonyms*, which cannot be freely interchanged in the same contexts. Synsets in *PWN* are interlinked by several kinds of relations, the most important being the hypernym/hyponym hierarchy. The hypernym/hyponym relations are most significant for nouns, in case of verbs this relation corresponds to *troponymy*, the relation of doing something in a specific manner.[3]

In the late 90s, the *PWN* approach was applied within the EU projects *EuroWordNet-1* and *2* (*EWN*), in which new national *WordNets* were created for Dutch, Italian, Spanish, French, German, Czech and Estonian. The synsets in the national *WordNets* were interlinked by means of *Interlingual Index* (*ILI*) describing the translational equivalents. In each language, for which a *WordNet* was created, we can find at least 15 000 synsets with equivalents in *PWN*.

In *VerbaLex*, all verb senses are directly linked to their English equivalents in *PWN*. The newly added synsets were linked to the *PWN* English synsets using the WordNet Assistant tool [11]. Appropriate equivalents could be found for 85 % out of 3 686 new synsets. In 15 % there is usually not direct lexicalized equivalent – for perfective, reflexive or prefixed verbs, or verbs with expressive or metaphoric meaning. For example, the Czech verb *povyskočit* ("jump up a little") is not found in any standard bilingual dictionary. The same holds for *povyskakovat* ("jump out of [something] one after another") and many other

[2] see Section 3

[3] E.g. "pohybovat se" (to move) → "chodit" (to walk) → "klopýtat" (to stumble).

Substance – in VerbaLex a semantic role:
1st-level – **SUBS**
2nd-level, PWN hypernym – **substance:1**
Two-layer semantic role – **SUBS(substance:1)**
Hyponymic lexical units as specifiers:
SUBS⟨solid:1⟩, *SUBS⟨liquid:3⟩*, *SUBS⟨gas:2⟩*, *SUBS⟨food:1⟩*, *SUBS⟨beverage:1⟩*, ...
Hyponymic subclass of particular examples:
SUBS⟨beverage:1⟩ = milk:1, alcohol:1, chocolate:1, fruit juice:1, soft drink:1, coffee:1, tea:1, drinking water:1, ...

Fig. 3: Example of a two-layer semantic role

verbs with two prefixes, e.g. *povyřizovat* ("finish doing things successively"), *dovyplnit* ("fill in an extra information").

3.1 Semantic Roles

Within the *EWN* projects, the core of the shared interlingual lexicon was defined as formed by the *Top Ontology* and a larger set of *Base Concepts*.[4] The top ontology also inspired the *VerbaLex* system of two-level semantic roles. Above all, we have selected the concepts covering large classes of lexical meanings. The classes correspond to the top hypernyms in the *PWN* hierarchy. We have chosen the hypernyms that best reflect the relevant meanings of the semantic roles and that are branching to expected hyponyms. *VerbaLex* 1st-level semantic roles use literals with sense number 1 or 2, i.e. basic meanings, which belong to the set of base concepts. The whole set of 1st-level roles is currently formed by 32 semantic roles, which describe very general meanings reflecting the reality. Each role covers one well recognized and specified meaning area, e.g. *ARTifact*, *ACTivity*, *INStrument*, *COMmunication*, *EVENt*, *LOCation*, or *TIME*.

The 2nd-level roles use direct hyponyms from *PWN* serving as a specification of the "most expected" meaning of this verb argument. The hyponyms of such literals can then serve as instances of the appropriate class. An example can be two-level roles denoting all substances (solid, liquid or gas), see Figure 3. The usage of 2nd-level roles can also be understood as subcategorization features, or selectional restrictions. They form an open system of labels, which can be continuously extended with regard to current applications.[5] The motivation for such approach lies in the aspiration to obtain a detailed description of particular verb senses. In this respect, *VerbaLex* fundamentally differs from the *Vallex* lexicon.

[4] At the beginning the set included about 1 000 base concepts, which was later extended to 8 000 concepts.
[5] *VerbaLex* currently contains 811 2nd-level semantic roles.

3.2 Semantic Classes of Verbs

The *VerbaLex* database describes not only meanings of the verb arguments, but also the meaning of the verb itself, which are one of the principal factors of its valency frames at both syntactic and semantic levels. The verb meaning is, besides the human readable definition, captured by detailed classification using verb semantic classes. Experimentally, we have chosen the classification system of English verbs by B. Levin [12], which builds upon the syntactic and semantic features of English verbs. The system divides verbs according to alternations of their participants. Within the *VerbNet* project of M. Palmer 48 basic semantic classes of Levin were extended to 83 classes (numbered 9.–91. [13]). Ambiguous verbs, originally instantiated in multiple classes, were detached to individual classes with their own meaning.

In *VerbaLex*, we have adapted the original set of semantic classes from *VerbNet* (numbered from 9 according to Levin and Palmer) and divided some of them to meaning subclasses resulting in 109 current verb class repository. The classes are originally based on the description of changes in the argument structure of English verbs, but after the adaptation they serve the purpose very well also for Czech.

4 Conclusions

In this paper, we have presented in detail the current state of the development of the *VerbaLex* valency lexicon of Czech verbs, including the electronic version with 10 449 verb lemmata, as well as its printed form with selected subset of the most frequent verbs. The full electronic form is available on-line after registration for academic and non-commercial purposes. The electronic database definitely offers more information by direct connection to the Czech and English WordNet and the possibility of intelligent browsing and searching. Even though the printed form of *VerbaLex* is a limited volume, we believe that a handy book also has its advantages.

Acknowledgements This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin project LM2010013 and by the Ministry of the Interior of CR within the project VF20102014003.

References

1. Lopatková, M., Žabokrtský, Z.: Vallex (2008) `http://ucnk.ff.cuni.cz`, 6460 lex.units.
2. Pala, K., Ševeček, P.: Valence českých sloves (Valencies of Czech Verbs). In: Sborník prací Filosofické fakulty Masarykovy university, A45, Brno (1997) 41–54
3. Svozilová, N., Prouzová, H., Jirsová, A.: Slovesa pro praxi: valenční slovník nejčastějších českých sloves (Verbs for Practice: Valency Dictionary of the Most Frequent Czech Verbs). Academia, Prague (1997)

4. Jakubíček, M., Horák, A.: Punctuation Detection with Full Syntactic Parsing. Research in Computing Science, Special issue: Natural Language Processing and its Applications **46** (2010) 335–343
5. Hlaváčková, D., Horák, A., Kadlec, V.: Exploitation of the VerbaLex Verb Valency Lexicon in the Syntactic Analysis of Czech. In: Proceedings of Text, Speech and Dialogue 2006, Brno, Czech Republic, Springer-Verlag (2006) 79–85
6. Filipec, J., Daneš, F., Mejstřík, V., eds.: Slovník spisovné češtiny pro školu a veřejnost (Dictionary of Literary Czech for School and Public). Academia, Prague (2000)
7. Havránek, B., ed.: Slovník spisovného jazyka českého (Dictionary of Literary Czech Language). Academia, Prague (1989)
8. Intitute of Czech National Corpus, FA CU: Czech National Corpus – SYN2000 (2000) `http://ucnk.ff.cuni.cz`.
9. Pala, K., Všianský, J.: Slovník českých synonym (Dictionary of Czech Synonyms). Lidové noviny, Prague (1996)
10. Fellbaum, C., ed.: WordNet: An Electronic Lexical Database. Language, Speech, and Communication. MIT Press, Cambrige (1998)
11. Němčík, V., Pala, K., Hlaváčková, D.: Semi-automatic linking of new czech synsets using princeton wordnet. In: Proceedings of the Intelligent Information Systems XVI Conference (IIS'08), Warszawa, Academic Publishing House EXIT (2008) 369–374
12. Levin, B.: English Verb Classes and Alternations: A Preliminary Investigation. University of Chicago Press, Chicago (1993)
13. Palmer, M., Rosenzweig, J., Dang, H.T., Kipper, K.: Investigating regular sense extensions based on intersective levin classes. In: Proceedings of the 17th international conference on Computational linguistics, Association for Computational Linguistics (1998) 293–299

Web Application for Semantic Network Editing

Adam Rambousek and Tomáš Hrušo

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
xrambous@fi.muni.cz, 396240@mail.muni.cz

Abstract. Semantic network editor, DEBVisDic [1], has been used to create more than 20 national wordnets. The editor was developed in Mozilla Development Platform, as the extension for Mozilla-based web browsers.
However, the development of the web-related technologies took a step from the browser-based extensions to rich web applications, usable in any browser. We decided to rewrite the editor from scratch and create multi-platform web-based application for general semantic networks editing. In the first phase, the editor will be used to build Open Dutch Wordnet, udner the Cornetto project. In case of successful deployment and evaluation, the editor will be enhanced to build any wordnet-like semantic network.

Key words: semantic network, ontology, editor, web application, DEB-VisDic

1 Introduction

The original wordnet, Princeton WordNet, is one of the most popular lexical resources in the NLP field [2]. It was followed by multilingual EuroWordNet 1, 2 projects (1998-99) [3] and Balkanet project (2001-4) [4] in which the wordnets for 13 languages have been developed (English, Dutch, Italian, Spanish, French, German, Czech, Estonian, Bulgarian, Greek, Romanian, Serbian and Turkish). In the course of this work the software tools for browsing and editing wordnets have been designed and implemented, without whose the job could hardly have been performed. Within the EuroWordNet project the Polaris (and Periscope) tools have been implemented and used [5].

For Balkanet project the browser and editor VisDic has been prepared at the NLP Laboratory at the Faculty of Informatics Masaryk University [6] since the development of the Polaris tool has been closed by 1999.

In comparison with the previous tools VisDic exploits XML data format thus making the wordnet-like databases more standard and exchangeable. Not only that, thanks to the XML data format used and to its dictionary specific configurability VisDic can serve for developing various types of dictionaries, i.e. monolingual, translational, thesauri and multilingually linked wordnet-like databases. The experience with the VisDic tool during Balkanet project has been positive [7] and it was used as the main tool with which all Balkanet wordnets were developed.

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2013, pp. 13–19, 2013.

2 DEB platform and DEBVisDic editor

VisDic, however, has its disadvantages, particularly it is designed for off-line use by single user, and team coordination is really difficult.

Based on the experience with VisDic, we designed and implemented more universal dictionary writing system that could be exploited in various lexicographic applications to build large lexical databases. The system has been called Dictionary Editor and Browser (further DEB) [8] and has been used in many lexicographic projects, i.e. for development of the Czech Lexical Database [9], or currently running Pattern Dictionary of English Verbs [10], and Family names in UK [11].

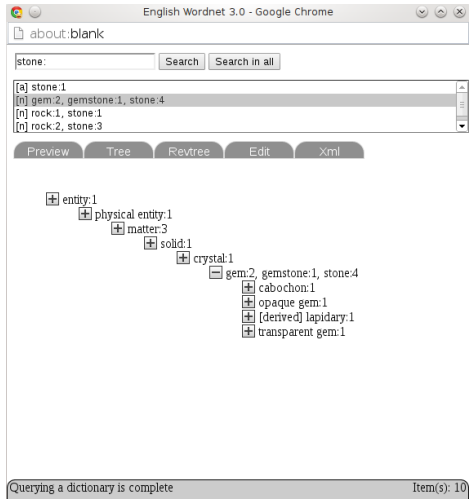


Fig. 1: Example of hypero-hyponymic tree.

The DEB platform is based on client-server architecture, which brings along a lot of benefits. All the data are stored on the server and considerable part of functionality is also implemented on the server, while the client application can be very lightweight.

This approach provides very good tools for team cooperation; data modifications are immediately seen by all the users. Server also provides authentication and authorization tools.

The design of the DEB allows us to modify it also for building wordnet-like databases. For this purpose, VisDic tool was re-implemented on top of the DEB platform, as the DEBVisDic editor[1].

DEBVisDic editor was designed as a client application for the DEB server, and created using the Mozilla Developement Platform[12], which was at the time the best option to design and build cross-platform GUI applications, utilizing open standard.

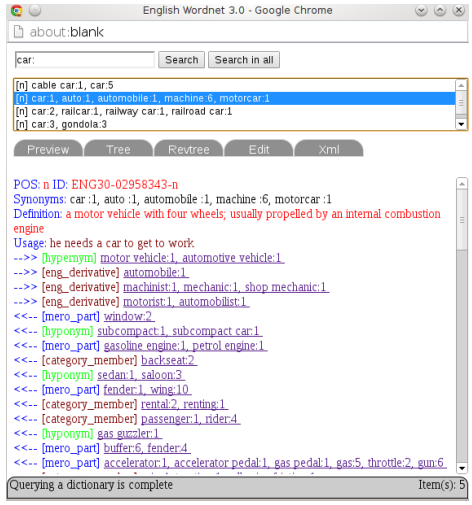


Fig. 2: Example of synset preview.

However, the applications based on Mozilla Developement Platform are limited only to Mozilla-based browsers (mainly Firefox), while users prefer many different web-browsers. Since the developement of DEBVisDic, Firefox browser introduced several major changes to application interface, limiting DEBVisDic to be used only in specified versions of Firefox browser. As a result, the editor would need major changes to work with recent Firefox versions.

Fortunately, the standards for web-based application supports much more features and are implemented by all the major web browsers. Considering all the options, we decided to re-implement DEBVisDic editor as a general web application, not limited to single web browser and without need to install specific extensions.

3 DEBVisDic 2

Thanks to the client-server architecture of DEB platform, no changes were needed on the server side. Only the client side application needed to be reimplemented, reusing the existing DEB interface. Main feature requests when designing the new version, were reimplementing all DEBVisDic features, and to provide application working in all major web browsers.

Similar to previous version, *DEBVisDic 2* aims primarily on wordnet-type semantic network browsing and editing, but supports different types of dictionaries. The application consists of main window with settings and separate windows for each dictionary that user want to edit. Single dictionary window includes the list of entries (synsets) and a set of tabs with several views on selected entry: basic preview, XML representation, hypero-hyponymic tree, and editing form. Context (right-click) menu provides functions for displaying and

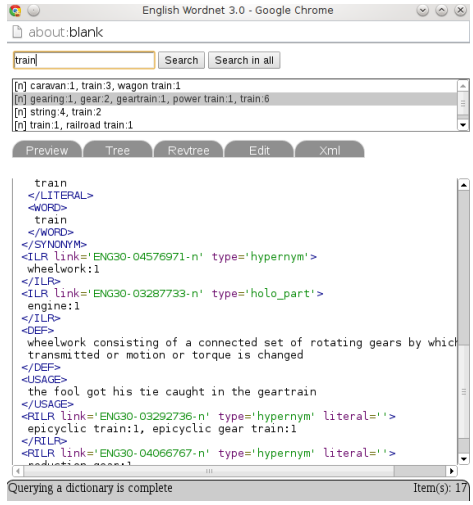


Fig. 3: Example of synset XML representation.

creating inter-dictionary links (i.e. display all synsets using selected ontology term).

DEBVisDic 2 utilizes Model-view-controller architecture and design follows this principle. Current open standards are used in the application: HTML and CSS for data presentation (view), and Javascript for application logic scripting (model, controller). The application is modular, with separate core shared by all the dictionaries, and a plugin with specific functionality for each type of dictionary.

Because the implementation of web-related standards (mainly Javascript) may vary in different browsers, several frameworks and libraries provide unified environment on top of the browser interface. After reviewing several frameworks, we decided to use jQuery library[13], that is versatile Javascript library for simpler document and data manipulation, but doesn't add unnecessary features, thus staying lightweight and not slowing down the application.

One of the most challenging features, was the implementation of the context menu functions, because of the huge differences in different browsers. In the end, we were able to implement the context menu to behave the same as in DEBVisDic, with the help of jQuery contextMenu plugin[1]. Pretty printing of entry in XML format is provided by Prettify plugin[2].

Apart from complete reimplementation of DEBVisDic tool, new version comes with several new features. For example, saving user settings (opened dictionaries and window positions, with the possibility to store more information) on the server, thus allowing user to switch browsers and computers, and continue in work.

[1] http://medialize.github.io/jQuery-contextMenu/

[2] http://google-code-prettify.googlecode.com/svn/trunk/README.html

Another major new feature are more generalized links and relations between dictionaries. It is possible to use any part of XML entry to build inter-dictionary search queries. For example, selecting all lexical units in a synset, automatically view details of ontology term for selected synset, all synonym or near synonym synsets between two wordnet languages.

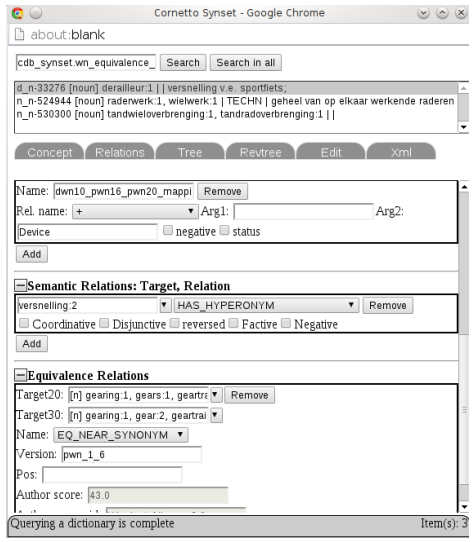


Fig. 4: Example of editing form.

4 Testing

DEBVisDic 2 editor was developed for the creation of Open Dutch Wordnet, based on the Cornetto project [14]. The data from the Cornetto project are not just a simple wordnet-like semantic network. It contains separate lexical database with detailed information about lexical units, and semantic network with synsets linked to each other, and also with links to corresponding lexical units and to several versions of English Wordnet.

Because of the database design, specific dictionary modules were needed for Cornetto Synsets, Cornetto Lexical Units, Open Dutch Wordnet, and English Wordnet, all inter-connected together.

Lexicographers' feedback after a few weeks of intensive editing is highly positive, and we are gathering comments to incorporate in future updates.

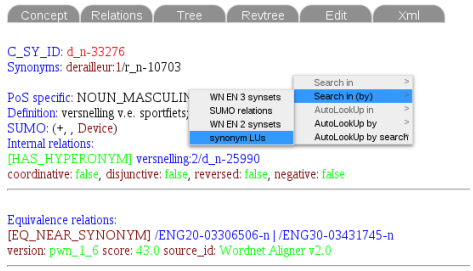


Fig. 5: Example of context menu with inter-dictionary links.

5 Future work

After the end of initial editing phase, *DEBVisDic 2* will be updated based on user feedback. We plan to add support for the editing of general wordnet-type semantic networks, thus encouraging the creation of more national wordnets.

Acknowledgements This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin project LM2010013.

References

1. Horák, A., Pala, K., Rambousek, A., Povolný, M.: First version of new client-server wordnet browsing and editing tool. In: Proceedings of the Third International WordNet Conference - GWC 2006, Jeju, South Korea, Masaryk University, Brno (2006) 325–328
2. Fellbaum, C., ed.: WordNet: An Electronic Lexical Database. MIT Press (1998)
3. Vossen, P., ed.: EuroWordNet: a multilingual database with lexical semantic networks for European Languages. Kluwer (1998)
4. Christodoulakis, D.: Balkanet Final Report, University of Patras, DBLAB (2004) No. IST-2000-29388.
5. M., L.: Polaris user's guide. Technical report, Belgium (1998)
6. Horák, A., Smrž, P.: VisDic – wordnet browsing and editing tool. In: Proceedings of the Second International WordNet Conference – GWC 2004, Brno, Czech Republic (2003) 136–141 `http://nlp.fi.muni.cz/projekty/visdic/`.
7. Horák, A., Smrž, P.: New features of wordnet editor VisDic. In: Romanian Journal of Information Science and Technology. Volume 7. (2004) 1–13
8. Horák, A., Rambousek, A.: DEB Platform Deployment – Current Applications. In: RASLAN 2007: Recent Advances in Slavonic Natural Language Processing, Brno, Czech Republic, Masaryk University (2007) 3–11
9. Rangelova, A., Králík, J.: Wider Framework of the Research Plan Creation of a Lexical Database of the Czech Language of the Beginning of the 21st Century. In: Proceedings of the Computer Treatment of Slavic and East European Languages 2007, Bratislava, Slovakia (2007) 209–217
10. Hanks, P.: Corpus pattern analysis. In: Proceedings of the Eleventh EURALEX International Congress, Lorient, France, Universite de Bretagne-Sud (2004)

11. Hanks, P., Cullen, P., Draper, S., Coates, R.: Family names of the United Kingdom. (2014)
12. Oeschger, I., et al.: Creating Applications with Mozilla. O'Reilly and Associates, Inc., Sebastopol, California (2002)
13. jQuery Foundation: jQuery. `http://jquery.org` (2013)
14. Horák, A., Vossen, P., Rambousek, A.: The Development of a Complex-Structured Lexicon based on WordNet. In: Proceedings of the Fourth Global WordNet Conference, Szegéd, Hungary, University of Szegéd (2008) 200–208

Portable Lexical Analysis for Parsing of Morphologically-Rich Languages

Marek Medveď and Miloš Jakubíček

Natural Language Processing Centre
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
{xmedved1,jak}@fi.muni.cz

Abstract. In this paper we present new approach to lexical analysis in the `Synt` parser. We describe three fast lexical analyzers we have exploited for lexical analysis and advantages of the `re2c` fast lexical analyzer in comparison to others. This paper shows a new lexical analysis workflow which is both easy to maintain and portable to new languages. Finally we provide an evaluation of the new lexical analysis against the original lexical analysis.

Key words: lexical analysis, synt parser

1 Introduction

Any parser that is not fully lexicalized (i.e. operates on some – fine or coarse grained – categories of words) must deal with the issue of performing the lexical analysis – mapping of words into the respective categories – before it can actually perform any parsing. For analytical languages the mapping procedure may be very simple and may boil down to the mapping to part-of-speech categories, however for wide-coverage parsers dealing with morphologically-rich languages, this is not the case. For those it holds that precise and complex (fine-grained) lexical analysis is the starting point for a successful syntactic analysis.

As with any hand-written rule system, one has to decide what formalism is optimal for designing and developing the lexical analysis. We hereby list the requirements for both of these based on our past experience – the formalism for development of the lexical analysis should be:

- **simple** to edit and maintain for language specialists
- easily **portable** to new languages
- **expressive** enough so as to be appropriate for the task

As for the actual execution of the lexical analysis (i.e. evaluation of the mappings on particular input), there is a single straightforward condition in place: it must be fast enough so as not to harm the speed of the parser.

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2013, pp. 21–26, 2013.

In this paper we present a new approach to lexical analysis in the Czech parser `Synt`[1,2] that satisfies the conditions listed above. At first we describe the former lexical analysis in the parser and its deficiencies, then we discuss possible alternatives that we exploited and finally we describe in detail the new approach based on a `re2c`-generated lexical scanner.

2 The `Synt` parser

Syntactic analyzer `Synt` has been developed over the past years in the Natural Language Processing Centre at Faculty of Informatics, Masaryk University. The `Synt` parser is based on a context-free backbone enhanced with contextual actions and performs a stochastic agenda-based head-driven chart analysis.

The input of `Synt` is a sentence morphologically annotated by the morphological analyzer `Ajka`[3] which uses an attributive tagset described in [4][1].

The lexical analysis follows immediately after loading an input sentence and its morphological annotation. In the context of a phrase-structure grammar, the lexical analysis provides mapping from words to the so called *pre-terminals* – non-terminal leaves which are directly rewritten to the surface word in the resulting syntactic tree. Therefore correct assignment of a word's pre-terminal is crucial for the analysis to succeed.

After the lexical analysis, the `Synt` parser proceeds with two-step parsing: first with a basic context-free analysis of the sentence and then with evaluation of complex contextual actions finally producing one of the possible outputs, which might be:

- **a phrase-structure tree**
  This is the main output of `Synt` and consists of a set of phrase-structure trees ordered according to the tree ranking. The system makes it possible to retrieve n-best trees effectively.
- **a dependency graph**
  A dependency graph represents a packed structure which can be utilized to extract all possible dependency trees. It is created by using the head and dependency markers that might be tied with each rule in the grammar.
- **set of syntactic structures**
  The input sentence is decomposed into a set of unambiguous syntactic structures chosen by the user.

3 Lexical analysis

In the current Czech grammar used in `Synt` the lexical analysis exploited the following token properties to be able to assign the pre-terminal correctly: the **word** itself, its **lemma** and its **morphological tag**, mainly the part-of-speech including its subclassification as provided by the Majka morphological analyser.

[1] Current version of the tagset is available online at `http:///nlp.fi.muni.cz/ma/`.

3.1 Tools

The former implementation was based on a hard-coded C module that simulated a decision tree operating on the three features (word, lemma, tag) used for the task. This approach, while being very efficient during the execution of the lexical analysis, suffered from several obvious drawbacks, the biggest one being that good knowledge of the system and programming skills were required in order to be able to develop the lexical analysis module, which made it practically impossible for a linguist specialist to do the task. Another, rather more engineering complications, included the necessity for special handling of wide-character strings and the fact that manual maintenance of a hand-written decision tree is quite error-prone.

For all these reasons we decided to replace the current lexical analysis module with a better formalism without the deficiencies listed above. We concluded that from the user perspective the mapping should be maintained "as is" in the most natural form consisting of lines mapping the word, lemma, tag triple to the pre-terminal. For this task, we tried to exploit three available tools for automatic generation of lexical scanners/analyzers – the lex[2], flex[3] and re2c[4].

A lexical scanner is a program which recognizes lexical patterns in text and after it matches a lexical pattern it executes an associated action. Input for all three tools is a description of the scanner in the form of pairs of regular expressions and C code actions, called rules. The output is then the generated scanner in the form of a C source file.

```
Lexical analysis:
"January" -> MONTH

Metagramar rule:
adv -> NUMBER '.' MONTH
```

Fig. 1: Principle of lexical analysis

From these three tools we first ruled out the `lex` because of its limited support for Unicode character set (wide/multibyte character strings). From the remaining two, `flex` and `re2c`, the latter one turned out to be much faster and hence our implementation is based on that one.

[2] Current version of lex is available online at `http://dinosaur.compilertools.net/lex/`.

[3] Current version of flex is available online at `http://flex.sourceforge.net/`.

[4] Current version of re2c is available online at `http://re2c.org/`.

```
def scan(string){
------ macro part ------
    IS_NUMBER = [0-9]+;
        ...
------ rule  part ------
    .*"\t".*"\t"{IS_NUMBER}"\t" {
------   action   ------
          preterm=NUM;
    }
           ...
}

def analyze(input_str){
   scan(input_str);
}
```

Fig. 2: Input for `re2c` and flex

3.2 New formalism for lexical analysis

The `Synt` parser can be used for many languages. The problem is that lot of language specialists (linguists) have basic or no skills in programing. Because of that we decided to create a preprocessing script that converts an input mapping definition file into a `re2c` source file. The definition file is a simple plain text file containing lines with lexical rules mapping a tag-, lemma- and word-triple to the respective pre-terminal.

Apart of the mapping lines, the definition file may contain comments and macros that are expanded in the mapping lines. Comment lines start with `#=`, macro lines take the format `m=Macro RegularExpresion` (replace any `Macro` instances with `RegularExpression` in the mapping lines (see example in Figure 3).

```
#=top comment
m=IS_NUMBER     [0-9]+    #=line comment
```

Fig. 3: Macro example with user comments

The lexical rule (exemplified in Figure 4) contains two parts: the first part describes the pattern to be matched and the second part is action to be executed upon the first part is successfully matched. The first part of the lexical rule contains tag, lemma and word in this order. The tag, lemma and word are regular expressions where predefined macros can be used, finally the 4^{th} field (second part of the lexical rule) is the action prescribing the pre-terminal to

be assigned (though it might be a short C snippet too). All the fields are tab-separated.

```
#=top comment
.*   {IS_MONTH}   .*   preterm=MONTH #=line comment
```

Fig. 4: Lexical rule example with user comments

In the lexical rules and macros predefined variables for word, lemma, tag, word index (denoted as `wi`) and lemma index (denoted as `li`) might be used – word, lemma and tag variables contain the respective input fields, the word (lemma) index is an integer representing word (lemma) position in the input sentence. For example if one wants to write a lexical rule for words that do not occur at the first position of the sentence, the following lexical rule might be appropriate:

```
"k1".*  .*  {ONLY_FIRST_UPPER} if(wi>0){preterm=NPR;}
```

Fig. 5: Lexical rule with predefined variables

4 Evaluation

Hereby we provide both a comparison of `flex` and `re2c` scanner generation and the actual speed of running the lexical analysis on a sample set of 6,160 sentences (see Table 1). From this evaluation it follows that the new lexical analysis system is not only easier to maintain and more portable, but also slightly faster than its hard-coded predecessor.

Table 1: Time of generating scanner from description of scanner

Lexical analyzer	Time of scanner generation
`flex`	22.9 min
`re2c`	2.5 min

Time of syntactic analysis with old and new lexical analysis

Lexical analyzer	Lexical analysis time
original `Synt` lexer	3.33 min
`re2c`	3.19 min

5 Conclusions

In this paper we have presented new lexical analysis used in the Czech parser `Synt`. We have discussed in detail the motivation behind this work: to have a portable, easy to maintain and fast to evaluate formalism for mapping input tokens to grammatical pre-terminals. We are convinced that the new approach based on the `re2c` lexical scanner will help us to port the parser to new languages faster and will be also less error-prone.

Acknowledgements This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin project LM2010013.

References

1. Kadlec, V.: Syntactic analysis of natural languages based on context-free grammar backbone. PhD thesis, Fakulta informatiky, Masarykova univerzita, Brno (2007)
2. Jakubíček, M., Horák, A., Kovář, V.: Mining phrases from syntactic analysis. In: Text, Speech and Dialogue. (2009) 124–130
3. Šmerk, P.: Fast Morphological Analysis of Czech. In: Proceedings of the Raslan Workshop 2009, Brno (2009)
4. Jakubíček, M., Kovář, V., Šmerk, P.: Czech Morphological Tagset Revisited. Proceedings of Recent Advances in Slavonic Natural Language Processing 2011 (2011) 29–42

Part II

Semantics

Acquiring Data for Textual Entailment Recognition

Zuzana Nevěřilová

NLP Centre, Faculty of Informatics,
Masaryk University, Botanická 68a, 602 00 Brno, Czech Republic

Abstract. Language resources are hardly ever large enough. Building language resources that can be used as a gold standard for semantic analysis requires effort and investment. We present a prototype for acquiring language resources by means of a language game which is a cheap but long-term method.
Games employed to acquire language resources are not new. For example games with a purpose are used for collecting common sense knowledge. The game presented in this paper is a work in progress. It collects annotated pairs text–hypothesis suitable for recognizing textual entailment in Czech.
The game narrative is based on Sherlock Holmes and dr. Watson dialogues. For generating the dialogue line we use rule-based approaches such as syntactic analysis, anaphora resolution, synonym and hypernym replacement, word order rearrangement and verb frame based inference. To generate natural sounding sentences we added a language model score (based on n-gram frequencies in a corpus).

Key words: textual entailment, language game, games with a purpose, GWAP

1 Language Resources and Data Complexity

Although Czech is spoken only by about 10 million people it cannot be considered as a less resourced language. However, Czech language resources (LR) follow the typical distribution when sorted by their complexity: the more complex a resource is the smaller it is.

N.B. that the term *language resource complexity* is not defined but it is often mentioned when describing a LR. According to [16] LR complexity means the data size as well as its characteristics relevant to annotation.

Currently no LR for recognizing textual entailment is available for Czech.

2 Recognizing Textual Entailment

"A fundamental phenomenon of natural language is the variability of semantic expression, where the same meaning can be expressed by, or inferred from, different texts." [2, p. 2]

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2013, pp. 29–37, 2013.

Recognizing Textual Entailment (RTE) is defined as follows: "A text t entails a hypothesis h ($t \Rightarrow h$) if humans reading t will infer that h is most likely true." [3, p. 18]

Although RTE seems to be defined imprecisely ("humans will infer", "most likely"), it is one of the most well defined problems in semantic analysis. RTE systems are evaluated by a collection of pairs text–hypothesis (h–t pairs). Each pair can be (repeatedly) annotated either as true (if t entails h) or false (if t does not entail h).

Building a collection of h–t pairs of a considerable size and diversity is a challenging task. Possible resources include (manually prepared) reading comprehension tests for children and for adults (such as PISA[1] or PIAAC[2]) as well as automated techniques. [2] describes four scenarios that lead to creation of h–t pairs in RTE2 challenge[3]. These scenarios are less applicable to Czech since many tools are in development (e.g. information retrieval system for Czech described in [7]). We therefore propose an alternative method for obtaining annotated h–t pairs by means of a game.

3 Acquiring Annotated Data

Games with a purpose (GWAP) is a new concept [1] in the field of *collaboratively constructed language resources* (CCLR). The idea is based on collective "human computation" where peoples' brains are used for solving problems that are difficult for computer programs (such as natural language understanding or image content recognition). Because GWAPs are games, the main motivation for contributors is the fun.

X-plain [12] is a game for one player whose purpose is to collect common sense propositions. It can be played in Czech and Slovak [8]. Three years after its first release X-plain collected 14,898 unique assertions in Czech and 5,703 unique assertions in Slovak. Some of the assertions are entered repeatedly, for example the assertion "South is the opposite of North" was entered six times in Czech version. The average ratio of repeated assertion is 1.1025 for Czech version and 1.2372 for Slovak version. The number of assertions is increasing every month depending on players' interest in the game[4].

4 The Game

The results of this one-player game and its power to acquire LRs (in middle-term and long-term) encouraged us to develop a new game which purpose is to collect annotated h–t pairs. Both true and false entailments are needed,

[1] `http://www.oecd.org/pisa/`

[2] `http://www.piaac.cz/`

[3] `http://pascallin2.ecs.soton.ac.uk/Challenges/RTE2`

[4] The provided numbers date back to 2013-11-07.

however users are likely to annotate clear true entailments and feel many false entailments annoying.

The game is based on several existing modules for natural language analysis and generation such as morphological analyzer and syntactic parser.

4.1 The Game Narrative

The game narrative refers to one element of detective stories: a dialogue between the detective and his/her assistant. The purpose of the dialogue is to explain the detective's reasoning that lead to the criminal case solving to readers.

In the game, the dialogue starts with a (short) criminal case the detective (human player) presents to his assistant (the program). The assistant tries to reformulate the story and to infer new facts. The basic screen with sample dialog is shown in Figure 1. The player can judge assistant's effort true or false or mark a sentence (syntactically) wrong.

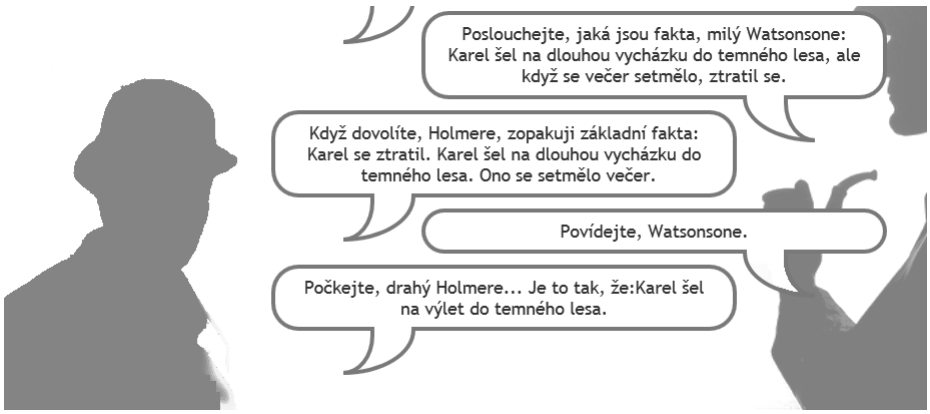


Fig. 1: The game environment is a dialogue between the detective Sherlock Holmer and his assistant dr. Watsonson.

From the RTE's point of view the human player enters a text t, the computer player proposes several hypotheses h and the human player annotates the entailment $t \Rightarrow h$. Depending on particular modules (see 4.3) h vary from simple paraphrases (i.e. syntactic rearrangements) to entailments.

Repeated annotations can be obtained as well. When a player is not about to tell a new story s/he can choose "get back to an old story". In this case a random story is selected from the story database.

4.2 User Experience and Data Complexity

Although the task of entailment is not even easy for humans (otherwise reading comprehension tests would not be used for testing people's understanding capabilities), the game is intended for non-expert users without any training. A progression loop is a typical game design element [17] and the game provides levels as usual. At each level more modules for paraphrasing and entailment are employed. We suppose the increase of complexity in each level reflects the intricacy of the entailments. Thus experienced players will be "trained" by the game itself.

The data complexity in relation with CCLRs is widely discussed in [16, p.10]. The players' task is somewhat similar to that in GWAPs but also to that of Wisdom of the Crowds (WotC). Unlike GWAPs no instant human feedback is present, but long-term feedback similar to Open Mind Common Sense [14] exists.

4.3 Modules

The application is based on several modules. The stories and entailments are not represented by a formalism (such as first order logic). Instead step-by-step the input sentences are transformed syntactically. For further processing, each transformation records its originator.

Parsing and partial anaphora resolution Players (in the detective's role) are asked to input a short story, thus a few sentences. The SET parser [9] divides each sentence on clauses if necessary and represents each clause as a set of sentence constituents (verb phrases, noun phrases, prepositional phrases, adverbial phrases, coordinations). Not individual words but phrases are subject of further processing.

At this phase the Czech anaphora resolution system Saara [13] supplements unexpressed subjects and replaces demonstrative pronouns with their antecedents. Table 1 outlines the processing of an example story. All other modules do not process sentences but these phrasal representations of individual clauses.

Individual phrases are marked according their syntactic roles, e.g. if the phrase's case is nominative it is marked as subject (SUBJ), if the phrase is an adverb it becomes the adverbial (ADV). At this level we cannot distinguish between an adverbial and an object therefore *do temného lesa* (in the dark forest) is marked as object (OBJ) although it is an adverbial.

Word reordering Czech is a (so called) free word order language i.e. *nearly* all orders of sentence constituents are allowed. Several aspects of word ordering in Czech (e.g. clitics, modal verbs in verb phrases, reflexive particles) are discussed in [5]. Different word orders signal different discourse structures but do not change the truth value. Thanks to this fact further processing leads to mostly correct results.

Table 1: Story representation: each sentence is divided in clauses, each clause is parsed on phrases. Phrases are marked according their syntactic roles: SUBJ(ect), VERB phrase, OBJ(ect), REFL(exive particle), ADV(erbial).

Karel šel na dlouhou vycházku do temného lesa, ale když se večer setmělo, ztratil se. Karel went for a long walk in a dark forest but when it got dark in the evening he got lost.										
Karel šel na dlouhou vycházku do temného lesa Karel went for a long walk in a dark forest				ono se večer setmělo it got dark in the evening				Karel se ztratil Karel got lost		
Karel	jít	(na) dlouhá vycház-ka	(do) temný les	on	se	večer	setmět	Karel	se	ztratit
Karel	go	(for) long walk	(in) dark forest	it		in the evening	get dark	Karel		get lost
SUBJ	VERB	OBJ1	OBJ2	SUBJ	REFL	ADV	VERB	SUBJ	REFL	VERB

Synonym and hypernym replacement We use Czech WordNet [15] for synonym replacement. No word sense disambiguation method is used and therefore false paraphrases are generated as long as the module replaces all possible word expressions of the story with their synonyms in all senses registered in Czech WordNet.

Since all transformations originators are recorded we can later discover WordNet synonyms unlikely in stories. For example *pes* has two senses: one corresponds to the synset `dog:1, domestic dog:1, Canis familiaris:1` in Princeton WordNet [4], the other corresponds to `martinet:1, disciplinarian:1, moralist:2`. A preliminary search in existing h–t pairs indicates the unlikely occurrence of the second sense in stories. In fact, none of the hypotheses generated with the replacement *pes–moralista* (moralist) were judged true. An example synonym replacement is shown in Table 2.

Similarly to synonym replacement phrases or their parts are replaced by their hypernyms. In this case two restrictions apply. First, we do not replace word expression by all hypernyms but omit those from the WordNet Top Ontology. Such replacement (e.g. replace "student" by "living entity") will never generate a natural sounding expression. Second, we do not replace by hypernyms in sentences with negative polarity. While in positive sentences (such as "He came in his new coupe.") the hypernym replacement (replace "coupe" by "car") is valid, in negative sentences the replacement results always in false entailments ("He did not came in his new coupe." does not entail "He did not came in his new car.").

Table 2: Synonym replacement using Czech WordNet: "vycházka" (walk) was replaced by "výlet" (trip). N.b. that the modifier "dlouhý" (long) had to be modified to fulfill the grammatical agreement with "výlet" (trip).

Karel	jít	(na) dlouhá vycházka	(do) temný les
Karel	go	(for) long walk	(in) dark forest
SUBJ	VERB	OBJ1	OBJ2
Karel	jít	(na) dlouhý výlet	(do) temný les
Karel	go	(for) long trip	(in) dark forest

Verb frame inference Word reordering and synonym replacement result in paraphrases while verb frame inference can result into new facts. In this module we take advantage of the Czech verb valency lexicon VerbaLex [6] and use verb valency frames for inferences of three types: equality, effect, precondition. The inference process is described in detail in [11]. It consists of several transformations of all phrases matched by the verb frame as shown in Table 3.

Table 3: This verb frame inference corresponds to the common sense inference "If someone gets lost s/he becomes unhappy."

Karel	se ztratil
Karel	got lost
SUBJ → SUBJ	ztratit se → být nešťastný
SUBJ → SUBJ	get lost → to be unhappy
Karel	byl nešťastný
Karel	was unhappy

Natural sounding sentences The system generates sentences using one, two or three modules. Even with only these three modules (word order, synonymy and hypernymy, verb frame inference) the application can produce tens to hundreds of sentences at one time. Since players find annotation of many sentences (that may be very similar) annoying we use a language model to select the most natural sounding sentences.

The appropriate n-gram frequencies were calculated using the Czes corpus, normalized (divided) by $corpsize$[5]. The resulting score is calculated according to Equation 1 where $ngrams$ means the n-gram raw frequency. n-grams of higher

[5] 465,102,710 tokens in 2013-11-07

n are more important for natural sounding sentences therefore they obtain a higher weight (by multiplication by a higher number).

$$score = 10^2 \sum \frac{2grams}{corpsize/2} + 10^3 \sum \frac{3grams}{corpsize/3} + \cdots + 10^5 \sum \frac{5grams}{corpsize/5} \quad (1)$$

Sentence Generation Each module uses an independent module for generating syntactically well-formed sentences. Sentence generation in Czech is a complex task because of grammar agreements between the verb and the subject and between noun phrase and its modifiers. The module for sentence generation declines all noun phrases and prepositional phrases using the application described in [10] and conjugates verb phrases as well.

4.4 The Game Loop

First, the player is asked to input a story or to choose a random existing story. Second, the story is scored (according to the number of its clauses, the number of different verbs and the number of named entities recognized). The player gains points for a new story or less points for choosing an existing story. Third, the parsed story is reproduced using the generation module. The player is asked to evaluate the reproduced story. Fourth, paraphrases and entailments ordered by its natural sounding score are proposed to the player for evaluation. For each annotation the player gets a point. In case the entailment (with the same annotation) is already in the database the player gains two points. Afterwards, the player can either add sentences to the story or begin a new game loop.

5 Conclusion and Future Work

We presented a new annotation game which aims to create a collection of h–t pairs for future Czech RTE system. The prototype is working although it misses modules for specific types of inference (see below). It can be found at `http://nlp.fi.muni.cz/projekty/watsonson/`. Our outlook is that in a few years we can obtain a large collection of stories, hypotheses and their annotations as well as information about the way the hypotheses were generated. The contribution of this work is therefore twofold: create a new CCLR and provide feedback to diverse software tools contributing to the generation process.

Feedback for existing software tools New sentences annotations provide information about:

- the distribution of correct and natural sounding word orders
- the distribution of Czech WordNet senses in stories
- the quality of syntactic parsing using SET
- the quality of anaphora resolution using Saara

Future Work We plan to add modules for entailments about time, locality, modality as well as involve the encyclopedic knowledge to the entailments.

With new modules transforming relative time to absolute time and vice versa can be entailed (e.g. transform "last Christmas" to "2012-12-25"). With encyclopedic knowledge module transformations like "Edvard Munch's The Scream" to "Edvard Munch painted The Scream" will be possible.

Acknowledgements This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin project LM2010013.

References

1. Ahn, L.v.: Games with a purpose. Computer 39(6), 92–94 (2006)
2. Dagan, I., Dolan, B., Magnini, B., Roth, D.: Recognizing textual entailment: Rational, evaluation and approaches. Natural Language Engineering 15, i–xvii (10 2009), http://journals.cambridge.org/article\char‘_S1351324909990209
3. Dagan, I., Roth, D., Zanzotto, F.M.: Tutorial notes. In: 45th Annual Meeting of the Association of Computational Linguistics. The Association of Computational Linguistics (2007)
4. Fellbaum, C.: WordNet: An Electronic Lexical Database (Language, Speech, and Communication). The MIT Press (May 1998)
5. Grepl, M., Karlík, P.: Skladba spisovné češtiny. Edice Učebnice pro vysoké školy, Státní naklad. (1986), http://books.google.cz/books?id=yV1iAAAAMAAJ
6. Hlaváčková, D., Horák, A.: Verbalex – new comprehensive lexicon of verb valencies for Czech. In: Proceedings of the Slovko Conference (2005)
7. Ircing, P., Pecina, P., Oard, D., Wang, J., White, R., Hoidekr, J.: Information retrieval test collection for searching spontaneous Czech speech. In: Matoušek, V., Mautner, P. (eds.) Text, Speech and Dialogue, Lecture Notes in Computer Science, vol. 4629, pp. 439–446. Springer Berlin Heidelberg (2007), http://dx.doi.org/10.1007/978-3-540-74628-7\char‘_57
8. Kostolná, M.: Vylepšení hry X-Plain (X-plain Enhancement). Bachelors thesis, Masarykova univerzita, Fakulta informatiky (2013)
9. Kovář, V., Horák, A., Jakubíček, M.: Syntactic analysis using finite patterns: A new parsing system for Czech. In: Human Language Technology. Challenges for Computer Science and Linguistics. pp. 161–171. Springer, Berlin/Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-20095-3\char‘_15
10. Nevěřilová, Z.: Declension of Czech noun phrases. In: Radimský, J. (ed.) Actes du 31e Colloque International sur le Lexique et la Grammaire. pp. 134–138. Université de Bohême du Sud à České Budějovice (République tchèque), České Budějovice (2012)
11. Nevěřilová, Z., Grác, M.: Common sense inference using verb valency frames. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) Proceedings of 15th International Conference on Text, Speech and Dialogue. pp. 328–335. Springer, Berlin / Heidelberg (2012)
12. Nevěřilová, Z.: X-plain – a game that collects common sense propositions. In: Sharp B., Zock M. (eds.) Proceedings of NLPCS. p. 47–52. SciTePress, Funchal, Portugal (2010)

13. Němčík, V.: Saara: Anaphora resolution on free text in Czech. In: Horák, A., Rychlý, P. (eds.) Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2012. pp. 3–8. Tribun EU, Brno (2012)
14. Speer, R.: Open mind commons: An inquisitive approach to learning common sense. In: Workshop on Common Sense and Intelligent User Interfaces (2007)
15. Vossen, P.: EuroWordNet: A Multilingual Database with Lexical Semantic Networks. Computers and the humanities, Springer (1998), http://books.google.cz/books?id=-qEep-1ib8UC
16. Wang, A., Hoang, C., Kan, M.Y.: Perspectives on crowdsourcing annotations for natural language processing. Language Resources and Evaluation 47(1), 9–31 (2013), http://dx.doi.org/10.1007/s10579-012-9176-1
17. Werbach, K.: Gamification: Course wiki. online (2013), [accessed 2013-11-07 from https://share.coursera.org/wiki/index.php/Gamification:Main]

Semi-automatic Theme-Rheme Identification

Karel Pala and Ondřej Svoboda

NLP Centre
Faculty of Informatics, Faculty of Arts
Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
{pala, xsvobo15}@fi.muni.cz

Abstract. In this paper we start from the theory of the Functional Sentence Perspective developed primarily by Firbas [1], Svoboda [2] and also Sgall, Hajičová [3] and make an attempt to formulate a procedure allowing to semi-automatically recognize which sentence constituents carry information that is contextually dependent and thus known to an adressee (*theme*), constituents containing new information (*rheme*), and also constituents bearing non-thematic and non-rhematic information (*transition*). Having themes and rhemes recognized as successfully as possible we also hope to investigate thematic progression (thematic line) in texts in the future. The core of the procedure and its experimental implementation for Czech (using the bushbank corpus CBB.Blog [4] as a data source) are described in the paper. Since the task is really complicated we only offer basic evaluation, which, in our view, shows that the task is feasible.

Key words: theme-rheme, Functional Sentence Perspective, topic-focus articulation

1 Introduction

The theory of the Functional Sentence Perspective (further FSP, [1,2,3]) has its origin in Czech linguistics, particularly in Prague Linguistic Circle. It states that in natural language sentences one can distinguish known, and new information. This is in agreement with our intuition, which reflects a sequential processing of the language data we are producing and receiving in the course of an information exchange. Terminologically, this is grasped in the following way: sentence constituents bearing known or contextually dependent information are called *themes*, elements serving as a backbone of a sentence are characterized as *transitions* and constituents carrying communicatively new (dynamic) information are called *rhemes*. Within thematic elements we can further distinguish *themes proper* (ThPr) and *diathemes* (DTh) which carry new information or refer to the new information from the previous text. Being important to the word order, *transitions proper* (TrPr) and *rhemes proper* (RhPr) are also recognized among transitional and rhematic elements. With regard to the FSP theory a natural question can be asked: is it possible to implement a procedure for identification of these elements in natural language sentences

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2013, pp. 39–48, 2013.

and texts semi-automatically? Such a procedure would be obviously useful for any kind of information processing, information extraction and observing thematic progression in texts. Some results by Karlík and Svoboda [5, in Czech] offer a solution which may lead to the more formal formulation of the procedure able to identify FSP elements in a sentence. They offer rules describing word order positions which can be occupied by individual sentence constituents and depending on their nature allowing to decide whether they can be labelled as thematic, transitional or rhematic. Here we try to reformulate their rules in a more formal way to be able to solve the task algorithmically in Czech sentences.

With regard to the work done in the FSP area we are following Firbas' and Svoboda's terminology (theme, rheme) while Prague group (Hajičová, Sgall and others) uses different terms *topic*, *focus* and call the area *topic-focus articulation* (TFA). We have to mention past attempts to propose the automatic procedure for FSP by Hajičová et al [6] and Steinberger et al [7]. Steinberger's attempt was designed for German, Hajičová's proposal dealt with simple English sentences. For both papers it is common that they do not offer any evaluation, thus it is difficult to judge at least approximately how successful the mentioned procedures were. Due to the time of their origin (approx. 20 years ago) they were not related to any corpora. Prague group members have published many papers related to the various aspects of the FSP theory, here we would like to mention especially the work related to the manual annotation of FSP (TFA) in PDT 2, see [8, in Czech]. Thus PDT contains labelling of the sentence constituents as thematic and rhematic elements and initially, we have considered the idea of the comparison of our results obtained automatically with PDT annotation obtained manually. After a closer look at the PDT annotation we, however, have come to the conclusion that this is a task for a separate paper – first, the differences in the notation have to be analyzed and only then the comparison can be tried. In any case we will pay attention to the comparison in the near future.

2 Motivation

The task described above has been considered difficult. Its successful solution will make it possible to obtain better insight into the information structure of utterances, which should allow more accurate information extraction as well as meaningful understanding of the thematic progression in natural language texts [9]. Our ambition in this paper is to show that the task is feasible, semi-automatically at least. We will concentrate on the basic aspects of the problem but we are aware of the wider context (e.g. anaphors or particles – rhematizers).

3 Word order positions

The free word-order in Czech makes it possible to combine sentence constituents rather freely. It can be observed that a finite verb takes the medium po-

sition in sentences in approx. 60%. Noun and adverbial phrases occur either before the verb or behind it but sentences with a verb at the beginning or the end appear regularly as well. The morphosyntactic cases in Czech permit to have an *object* at the beginning of the sentence and a *subject* at the end frequently. Thus we deal with sentences displaying various word-order patterns, particularly with their main types. As a resource we use Czech BushBank [10] with 31,822 syntactically tagged sentences. We consider the following basic word-order patterns:

- s(VP ADP), s(VP NP ADP), s(VP ADP NP), . . . , verb is in the initial position: 6,410 (20.1%),
- s(NP VP NP), s(ADP NP VP NP), . . . , verb is in the medial position 18,746: (58.7%),
- s(VP), s(ADP VP), s(NP ADP NP VP), s(NP NP VP), . . . , verb is in the final position: 6,666 (20.9%).

We distinguish up to five word-order positions in Czech sentences: pre-initial (usually occupied by conjunctions that are not a part of a clause), initial, post-initial (where enclitics follow Wackernagel's rule) medial and final. The order of enclitic elements in Czech is strictly given: auxiliary forms of verb *být – to be*) are followed by reflexives (pronouns or particles), then by personal, adverbial and demonstrative pronouns.

Unlike the medial position, the initial and the final positions must always be present (even in the form of a merged initial-final position) and can contain only one sentence constituent. The initial, medial and final positions may be occupied by a noun phrase or an adverbial phrase, or a verb. A conjunction or a particle may occur in the pre-initial position, affecting e.g. the modality of the sentence.

So we can split the problem into three tasks:

- recognizing the sentence constituents (using the IOBBER chunker[10] and SET parser [11],
- segmenting a sentence into word-order positions (pre-initial, initial, post-initial, medial, final),
- identifying what sentence constituents occupy them and deciding if they contain thematic, transitional, or rhematic elements.

3.1 Recognition of the sentence constituents

For identifying the sentence constituents and word-order positions they belong into we use partial syntactic annotation in the bushbank originally supplied by a statistical parser IOBBER [10] (used for noun phrases) and rule-based SET parser (verb phrases and clauses), disambiguated manually afterwards. At the experimental stage partial syntactic information was sufficient. Morphological information was also necessary for the task: POS of the constituents plus all respective grammatical categories contained in tags. For example, the Czech

noun *žárovky* (*light bulbs*) has the tag k1gFnPc1, expressing the corresponding categories, i. e. noun, feminine, plural, nominative. For a verb *mluvíme* (*we speak*) the respective tag is k5eAaImIp1nS, which provides grammatical categories relevant from the FSP point of view:

- tense and modality (present tense – aI, indicative – mI, further Temporal and Modal Exponents – TMEs),
- person and number (1st person – p1, singular – nS, further Personal and Number Exponents – PNEs).

They are needed for recognizing some thematic and transitional elements which are a part of a verb form in Czech.

There are, however, problems with prepositional noun phrases – it is difficult to recognize which sentence constituent they belong to. So far we decided to work with the longest possible constituents but we are aware that this is just a preliminary solution, which has to be tested in detail. The accuracy of the used parsers output obviously influences the assigning of the thematic and rhematic labels to the constituents but in our view this is not so critical, though the accuracy of the parsers does not exceed 89%.

We also have to mention some particles which play a relevant role in FSP tagging and are problematic also in parsing. These particles are called rhematizers (e.g. *jen (only), právě (just), ...*) and they indicate that a sentence constituent which follows them has to be labelled as rhematic. This is captured in rules (particularly in rule 7 given below in the next Section 4. The list of rhematizers in Czech is rather small, approximately not more than 10. We do not deal with rhematizers in detail since they are handled by the parser SET as units hanging on sentence constituents with rather unspecified status. To explore rhematizers in detail is a task for future research.

An important point has to be made here – due to the complexity of the task of Th/Rh labelling we have decided to work with simple sentences at the beginning. The rule was to allow no punctuation thus, among correctly formed sentences, avoiding subordinate clauses (which would, in cases, occupy word-order positions by their own) and the need for full syntactic analysis in which case a treebank corpus or parsers would have to be used. This is motivated by the fact that we have to answer simple questions first to gain firm ground for solving more complex parts of the problem. After having managed simple sentences we can come to complex clauses taking a full approach to the problem breaking any artificial limitations we used in the experiment.

4 A procedure for assigning themes and rhemes

Having the information mentioned above we can try to formulate basic rules for determining thematic, transitional and rhematic elements in a Czech sentence:

1. The first step is to recognize clause boundaries by finding the pre-initial position occupied with a clause conjuction,

2. If an adverb of time or place appears in the initial or medial position it is labelled as DTh (diatheme),
3. Enclitics (personal and other pronouns and auxiliary forms of *být – to be*) always take the post-initial position and are labelled as ThPr (theme proper), they are also anaphoric expressions. For dealing with them we should be able to recognize anaphors and their antecendents, however, this aspect of the issue calls for more detailed analysis. There is a tool for Czech able to handle anaphors but its integration into the FSP area has to be further explored in future. This rule has strongly deterministic character,
4. Any constituent in the final position is labelled a RhPr (rheme proper) – the rule seems to work very reliably,
5. A finite verb expressing grammatical categories of the subject is labelled as ThPr (theme proper) as well as TrPr (transition proper) for bearing temporal and modal (TMEs) categories,
6. Noun phrases in the initial or medial position are usually labelled as DTh (diatheme), noun phrases in the final position are most frequently labelled as RhPr. This rule appears to be almost universal.
7. If a rhematizer occurs in a sentence it indicates that a sentence constituent which follows it has to be labelled as rhematic.

4.1 An Example

The example sentence is taken from the corpus CBB.Blog [10] (shown in the original vertical format):

Motorka si razí cestu temným údolím a na plastovou pokrývku doráží neúprosný déšť. (*A motorbike was making its way through a dark valley and rain drops were beating the plastic cover mercilessly.*)

The individual clauses are found by means of finding sentence-level coordinate conjunctions which belong to pre-initial positions (*a* by rule 1). In these simple clauses, noun (*Motorka*) and prepositional (*na plastovou pokrývku*) phrases are recognized as sentence constituents on their own, occupying initial positions and by rule 6 they are found diathematic (DTh). A reflexive pronoun *si* in the first clause falls by Wackernagel's rule in the post-initial position and is considered a theme proper (ThPr, rule 3). By the same rule 6, the noun phrases *cestu* and *temným údolím* at the end of a clause (final position) are labelled rhemes proper (RhPr). In medial positions, finite verbs *razí* and *doráží* bearing TMEs (`mI` present tense, indicative mode, `aI` imperfect aspect) are assigned transitions proper (TrPr) and themes proper (ThPr) for having PNEs (`p3` 3rd person, `nS` singular).

The segmentation into word-order positions is a task for the first tool, saving also a summary of the contents, whether e.g. a sentence-level conjunction, a noun phrase, an adverb or a verbal exponent (PNE or TME) is present. The second tool then uses the combination of the supplied information and the position in a sentence to assign one or more FSP elements if a rule is defined. Development was carried in Python for the speed of development and the

```
<s>
Motorka     Motorka   k1gFnSc1P?       clause:ff.syntax.1268  k1c1gFnS:ff.syntax.1270
si          se        k3xPyFc3P-       clause:ff.syntax.1268  yFxPk3c3:ff.syntax.1271
razí        razit     k5eAaImIp3nSP?   clause:ff.syntax.1268  vp:ff.syntax.1269
cestu       cesta     k1gFnSc4P-       clause:ff.syntax.1268  k1c4gFnS:ff.syntax.1272
temným      temný     k2eAgNnSc7d1P-   clause:ff.syntax.1268  k1c7gNnS:ff.syntax.1273
údolím      údolí     k1gNnSc7P-       clause:ff.syntax.1268  k1c7gNnS:ff.syntax.1273
a           a         k8xCP-
na          na        k7c4P-           clause:ff.syntax.1274  k7c4:ff.syntax.1276
plastovou   plastový  k2eAgFnSc4d1P-   clause:ff.syntax.1274  k7c4:ff.syntax.1276
pokrývku    pokrývka  k1gFnSc4P-       clause:ff.syntax.1274  k7c4:ff.syntax.1276
doráží      dorážet   k5eAaImIp3nSP?   clause:ff.syntax.1274  vp:ff.syntax.1275
neúprosný   úprosný   k2eNgInSc1d1P-   clause:ff.syntax.1274  k1c1gInS:ff.syntax.1277
déšť        déšť      k1gInSc1P-       clause:ff.syntax.1274  k1c1gInS:ff.syntax.1277
.           .         kIx.P-
</s>
```

Fig. 1: Output of the parsers IOBBER and SET as stored in the CBB Corpus

temporary nature of both tools as we plan to offer basic FSP annotation directly in a parser's output.

5 Results and Evaluation

Presently, we have performed a basic experiment, in which FSP labels have been assigned to sentence constituents with some limitations: in the experiment we decided to work only with simple sentences containing coordinate clauses and with omitted punctuation, for which our two tools word-order segmenter and FSP tagger (both written in Python) provide: identification of the word-order positions in sentences taken from the CBB corpus (9,513 sentences) (by segmenter), assigning thematic and rhematic labels to the sentence constituents occuring in these sentences (by FSP tagger, see Figure 2 and Figure 3).

Table 1: The results of first the experimental Th/Rh labelling (recall)

All sentences in CBB.Blog	32,287	
Simple sentences	9,513	100.0%
Labelled (fully or partially)	8,481	89.2%
Not labelled	1,032	10.8%

The results in Table 1 are very basic by their nature, they say that recall is 89.2% which is the result that has exceeded our expectations. It is partially limited by the quality of the parser's output.

```
<s>
    <initial NP="k1c1gFnS" diatheme="NP">
Motorka     Motorka    k1gFnSc1P?       clause:ff.syntax.1268  k1c1gFnS:ff.syntax.1270
    </initial>
    <post-initial theme-proper="post-initial position">
si          se         k3xPyFc3P-       clause:ff.syntax.1268  yFxPk3c3:ff.syntax.1271
    </post-initial>
    <medial PNE="k5eAaImIp3nSP?" TME="k5eAaImIp3nSP?" theme-proper="PNE"
            transition-proper="TME" verb="razit">
razí        razit      k5eAaImIp3nSP?   clause:ff.syntax.1268  vp:ff.syntax.1269
    </medial>
    <medial NP="k1c4gFnS" diatheme="NP">
cestu       cesta      k1gFnSc4P-       clause:ff.syntax.1268  k1c4gFnS:ff.syntax.1272
    </medial>
    <final NP="k1c7gNnS" rheme-proper="final position">
temným      temný      k2eAgNnSc7d1P-   clause:ff.syntax.1268  k1c7gNnS:ff.syntax.1273
údolím      údolí      k1gNnSc7P-       clause:ff.syntax.1268  k1c7gNnS:ff.syntax.1273
    </final>
    <pre-initial conjunction="a">
a           a          k8xCP-
    </pre-initial>
    <initial NP="k7c4" diatheme="NP">
na          na         k7c4P-           clause:ff.syntax.1274  k7c4:ff.syntax.1276
plastovou   plastový   k2eAgFnSc4d1P-   clause:ff.syntax.1274  k7c4:ff.syntax.1276
pokrývku    pokrývka   k1gFnSc4P-       clause:ff.syntax.1274  k7c4:ff.syntax.1276
    </initial>
    <medial PNE="k5eAaImIp3nSP?" TME="k5eAaImIp3nSP?" theme-proper="PNE"
            transition-proper="TME" verb="dorážet">
doráží      dorážet    k5eAaImIp3nSP?   clause:ff.syntax.1274  vp:ff.syntax.1275
    </medial>
    <final NP="k1c1gInS" rheme-proper="final position">
neúprosný   úprosný    k2eNgInSc1d1P-   clause:ff.syntax.1274  k1c1gInS:ff.syntax.1277
déšť        déšť       k1gInSc1P-       clause:ff.syntax.1274  k1c1gInS:ff.syntax.1277
    </final>
.           .          kIx.P-
</s>
```

Fig. 2: Word-order positions identified and FSP elements recognized

As to the assessment of the accuracy of the Th/Rh labelling, i.e. to evaluation how successful the labelling was, we have selected a sample containing 300 sentences each with assigned Th/Rh and evaluated it manually. The results can be seen in Table 2 and they show that our experiment is making sense.

Table 2: The accuracy of the first experimental Th/Rh labelling

	Sample sentences	300	100.0%
A	Correctly labelled sentences	203	67.6%
Ap	Correctly labelled sentences, partly	61	20.2%
N	Sentences with Rh not recognized	18	6.0%
Np	Sentences with errors in labelling	19	6.2%

The presented results can be characterized as more than promising – first two lines include sentences in which the Th/Rh labels have been assigned sucessfully, the line A includes sentences in which Th/Rh labels have been assigned completely, line Ap contains sentences in which label Rh is assigned correctly but some sentence constituents are not labelled, however, the result can be still considered acceptable. Similarly, line N comprises sentences in which no Th/Rh labels are assigned to sentence constituents, i.e. this result is completely negative. In Np there are sentences where some sentence constituents are labelled correctly but Rh is not identified. On the whole, we can take A and Ap together obtaining 87.7% sentences processed successfully. This result can be considered suitable for possible future applications.

It has to be remarked that there is a phenomenon that lowers accuracy of tagging – it is related to the infinitives in Czech. They can be parsed in different ways that call for a deeper analysis – this causes errors in Th/Rh labelling.

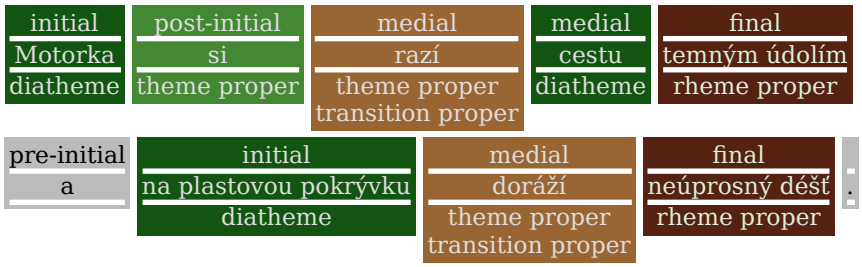


Fig. 3: Graphical output of the FSP parser

It can be seen how the Th/Rh labels are assigned to the individual sentence constituents and how the word-order positions are recognized. In the applica-

tion it is possible to click on words in the sentence – then tags become visible. In this way one can observe the necessary details of the analysis.

6 Conclusions

In the paper we have been dealing with the task consisting of the identification of word-order positions and semi-automatic theme-rheme tagging in Czech. Starting from the work of Karlík and Svoboda [5] we formulate rules capturing behaviour of the constituents in Czech sentences with regard to the word-order positions they occupy. The rules form a procedure for labelling thematic, transitive and rhematic elements in Czech sentences. The experimental version of the procedure has been implemented as a tool having two modules: the segmenter processing simple Czech sentences with the standard word-order and FSP tagger tagging sentence constituents as thematic and rhematic. We are well aware of the experimental character of the presented results but, in our view, they show that it makes sense to go in the indicated direction.

Acknowledgements This work has been partially supported by the Ministry of Education of CR under the LINDAT-Clarin project LM2010013.

References

1. Firbas, J.: Functional sentence perspective in written and spoken communication. Cambridge University Press (1992, reprinted 1995)
2. Svoboda, A.: Kapitoly z funkční syntaxe (Chapters from the Functional Syntax). In: Spisy pedagogické fakulty v Ostravě (Writings of the Pedagogical Faculty in Ostrava), svazek (vol.) 66. (1989)
3. Hajičová, E., Buráňová, E., Sgall, P.: Aktuální členění věty v češtině (Functional Sentence Perspective in Czech). Academia, Prague (1980)
4. Grác, M.: Rapid Development of Language Resources. PhD thesis, Masaryk University, Brno (2013) Available on-line: `http://is.muni.cz/th/50728/fi_d/?lang=en`.
5. Karlík, P., Svoboda, A.: Skladba češtiny pro cizince (Czech Syntax for Foreigners), Brno (1982)
6. Hajičová, E., Sgall, P., Skoumalová, H.: An automatic procedure for topic-focus identification. In: Journal of Computational Linguistics, Vol. 21, Issue 1, March 1995, MIT Press Cambridge (1993) 81–94
7. Steinberger, R., Bennett, P.: Automatic Recognition of theme, focus and Contrastive stress. In: Proceedings of the Conference Focus and NLP. (1994)
8. Veselá, K., Havelka, J.: Anotování aktuálního členění věty v Pražském závislostním korpusu (2003) ÚFAL/CKL TR-2003-20, available on-line: `http://ufal.mff.cuni.cz/pdt2.0/publications/VeselaHavelkaTR2003.pdf`.
9. Svoboda, A.: Diatheme: a study in thematic elements, their contextual ties, thematic progressions and scene progressions based on a text from Ælfric. Univerzita J.E. Purkyně, Brno (1981)

10. Radziszewski, A., Grác, M.: Using Low-Cost Annotation to Train a Reliable Shallow Czech Parser. In: Proceedings of the TSD Conference, Heidelberg, Springer (2013) 575–584 Available on-line: http://www.phil.muni.cz/plonedata/wkaa/BSE/BSE_2003-29_Scan/BSE_29_09.pdf.
11. Kovář, V., Horák, A., Jakubíček, M.: Syntactic Analysis Using Finite Patterns: A New Parsing System for Czech. In: Human Language Technology: Challenges for Computer Science and Linguistics. (2011) 161–171
12. Golková, E.: Bibliography of the publications of Professor Jan Firbas. In: Brno Studies in English: 29. (2003) 99–108 Available on-line: http://www.phil.muni.cz/plonedata/wkaa/BSE/BSE_2003-29_Scan/BSE_29_09.pdf.
13. Šmerk, P.: Unsupervised Learning of Rules for Morphological Disambiguation. In: Lecture Notes in Computer Science, Springer Verlag (2004) 211–216

Part III

Text Corpora

Intrinsic Methods for Comparison of Corpora

Vít Baisa and Vít Suchomel

Masaryk University,
Botanická 68a
Brno, Czech Republic
xbaisa@fi.muni.cz
xsuchom2@fi.muni.cz

Abstract. Since there are only very few techniques for quantitative and systematic comparison of text corpora we proposed and implemented several novel methods. The procedures were applied to comparing two very large web based Czech text corpora: czTenTen12 and Hector with more than 4.47 and 2.65 billion words, respectively. All methods are fully automatic and some of them are even language independent. We released some of them so they can be used instantly for comparison of other corpora.

Key words: text corpus, corpora comparison

1 Introduction

Nowadays, thousands of new corpora are built each month. using automatic methods like WebBootCaT [1] and similar. In some systems, creating a new corpus is a matter of several mouse clicks. But despite the overwhelming amount of corpora available now there is no method for their comparison.

It clearly depends on the purpose and usage of the corpus: sometimes, just the size matters, sometimes texts in colloquial or internet language are required etc. In this paper we describe intrinsic methods for comparing corpora.

We were interested especially in comparing two recent very large web-based Czech text corpora – czTenTen12 [2] and Hector [3,4].

Methods presented in this paper are divided into several groups: a) general intrinsic properties, b) text cleaning and processing, c) wordlist-based methods and d) syntactic analysis.

Some initiatives dealing with comparing corpora were [5,6] and [7] but in general not much attention was paid to this topic.

2 General intrinsic techniques

2.1 Size

The basic intrinsic measure is the size of the corpus (number of words or tokens in the data). Generally, the larger the corpus, the better: 'Most phenomena

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2013, pp. 51–58, 2013.

in natural languages are distributed in accordance with Zipf's law, so many words, phrases and other items occur rarely and we need very large corpora to provide evidence about them' [8]. There are 1.71 times more words and 1.39 times more sentences in czTenTen12 than in Hector according to Table 1. The measurement of words, tokens and sentences depends on the means of tokenization and sentence detection algorithms used for processing corpus data. These algorithms may differ in various corpora.

Table 1: Comparison of corpus size – gigabytes of textual data, billions of tokens, billions of words, millions of sentences

CORPUS	BYTES	TOKENS	WORDS	SENTENCES
Hector	17 GB	3.285 bn	2.607 bn	219 m
czTenTen12	31 GB	5.437 bn	4.458 bn	303 m

2.2 Diversity of sources

Unlike czTenTen12, Hector was constructed from manually selected web sites with large and good-enough-quality textual content (e.g. news servers, blog sites, discussion fora) [4]. Although such selection of particular documents may contribute to text quality, it may also decrease text diversity, e.g. genres like novels, legal documents or descriptions of goods are completely omitted. Therefore diversity of sources should be taken into account when building web corpora.

To measure the diversity of corpus sources, one could count number of web pages, web domains and top level domains represented in the corpus. The more diverse source of the data, the better coverage of language by the corpus may be expected. Since Hector was not available with necessary metadata, only the diversity of czTenTen12 could be evaluated and displayed in Table 2.

Spreading corpus sources over many top level domains may be useful for languages spoken all over the world (e.g. English) or to obtain variants of the language spoken in diferrent countries (e.g. Brasilian vs. European Portuguese). However, it may not help to find good quality texts in languages spoken in a single country like Czech.

Table 2: Diversity of sources – web pages, web domains, average number of pages per domain, median of pages per domain, top level domains

CORPUS	PAGES	DOMAINS	AVG	MED	TLDS
czTenTen12	9,747,315	233,122	42	4	97.6 % cz

3 Text processing and cleaning metrics

3.1 Sentence length

In the footsteps of [4] comparing sentence length of Czech corpora SYNT2005 and Hector, czTenTen12 is added to the comparison in Figure 1.

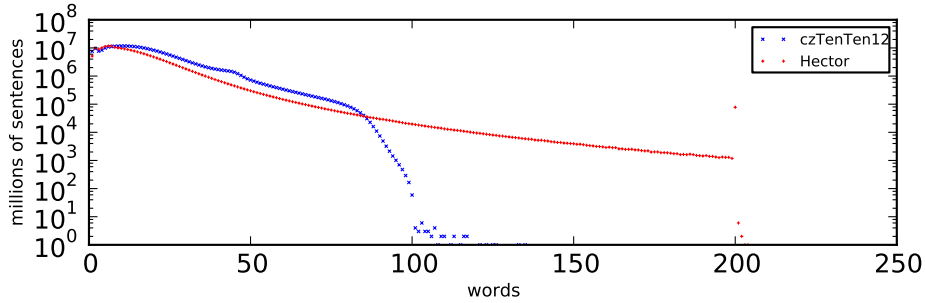


Fig. 1: Distribution of number of words in sentences (logarithmic y scale)

It can be seen the algorithm for detection of sentence borders trims sentences with more than 200 words in case of Hector. Another approach is used in case of czTenTen12 – the algorithm is more likely to end sentences longer than 80 words, see Table 3

Table 3: Sentence length metrics – peak length, average length, median length, observed threshold of breaking long sentences

CORPUS	PEAK	AVG	MED	LONG SENTENCES
Hector	7	15.0	12	hard limit of 200 words
czTenTen12	10	17.9	16	less strict rules after 80 words

3.2 Data duplicity

The less duplicate texts in a corpus the better. However, a very strict deduplication results in removing usable data needlessly. The deduplication strength of both examined corpora was intentionally selected by respective corpus designers. Duplicate and near duplicate texts were avoided in both corpora using a n-gram comparison method: paragraphs containing more than 30% seen 8-grams were removed from Hector, while paragraphs containing more than 50% seen 7-grams were removed from czTenTen12. Although both methods are similar, particular algorithms are different. We propose to compare the degree of deduplication based on a stricter n-gram comparison. Table 4 contains results of the

experiment performed using onion[1] set to remove sentences consisting of 50% seen 5-grams of sentences (smoothing disabled). Since the observed size drops are not large, it can be concluded both corpora are deduplicated sufficiently. CzTenTen12 was deduplicated more strictly than Hector.

Table 4: Corpus size difference after a strict deduplication of sentences

CORPUS	BYTES	TOKENS	SENTENCES
Hector	-23.3 %	-25.8 %	-23.6 %
czTenTen12	-17.6 %	-18.7 %	-18.4 %

4 Wordlist based techniques

4.1 The test

One of several steps in building a new corpus is language filtering. The aim is usually to have only one language in corpus so the language filter must identify texts out a desired language and remove them from the corpus. A problem might emerge if there are many small portions of foreign language texts below paragraph level. In this case one needs to set a level of granularity for the language filtering. But in general: the less words from a foreign language in your corpus the better filtered it is. Language statistics are not worsened by noise from a foreign language.

The test takes positions of all variants of English determiner *the* (THE, the, The, thE etc.) from a wordlist. These positions are then compared between examined corpora. The determiner was chosen since it is the most frequent word in English texts.

Table 5: The test results for Hector and czTenTen12

czTenTen12		**Hector**	
The	941	The	757
the	1,109	the	1,185
THE	24 k	THE	12 k
THe	942 k	ThE	264 k
tHe	2.4	tHe	314 k
ThE	2.7 M	THe	435 k
tHE	4.8 M	thE	654 k
thE	4.8 M	tHE	847 k

[1] http://nlp.fi.muni.cz/projects/onion

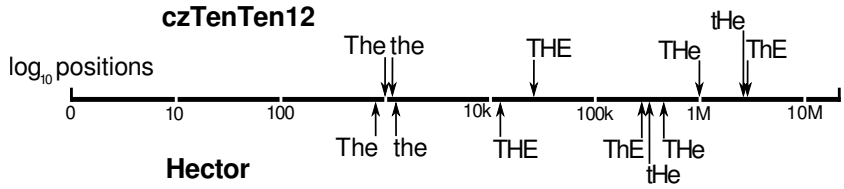


Fig. 2: Visualization of The test

You can see results in Table 5 and visualization in Figure 2. Most of variants of the determiner in Hector are more frequent then in czTenTen12 wordlist which could be interpreted as that czTenTen12 contains less portions of English texts.

4.2 Filtering wordlists

The motivation for this method is very similar to The test – we want to know if a corpus contains only the desired language. For this purpose we use morphologic analyzer to filter out all unknown words from wordlists and then check how many words remained.

It is not much important if the analyser can recognize all Czech words in wordlists. It is fair that the same filtering is applied on both corpora. We used Czech fast analyser Majka [9].

Results of this method also reveal problems in missing diacritics, wrong encoding of texts, number of typos – all these are not recognised by the analyzer and at the same time are not desirable in corpora.

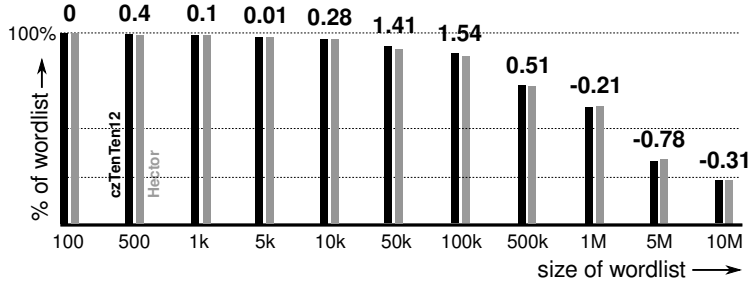


Fig. 3: Visualization of filtering wordlist test

In Figure 3 you can see results of filtering wordlists for Hector and czTenTen12. On x axis there are various lengths of wordlists: we filtered top parts of wordlists to see how the filtering is changing from top to bottom of wordlists.

The lengths of filtered resulting lists are very similar: the height of a rectangle is ratio between length of unfiltered and filtered list. The number

above each column is difference between ratio of czTenTen12 and Hector. If the number is positive then the respeective part of czTenTen12 filtered list is bigger then of the Hector's and means that Hector was filtered more.

First 500,000 words from both wordlists are less filtered in czTenTen12 but the rest of wordlists are less filtered in Hector. The interpretation of these results is not straightforward but we can conclude that from the perspective of this method both corpora are very similar and Hector is slightly better for low frequency words.

4.3 Keyword comparison

Following [7], we extracted lowercase keywords from czTenTen12 with Hector as the reference corpus (and vice versa) to explore in which words these corpora differ the most. It can be observed both recent web corpora contain more data from internet message boards and less news documents than the Czech National Corpus. In addition, there is much text from women fora in Hector.

Notable top keywords from Hector vs. czTenTen12:

- blog, holky, teda, taky, ahoj, fakt, ahojky, super, moc, ráda, takže – mostly informal, some in feminine gender (discussions of women)
- chtěla, řekla – verbs in feminine gender
- jdu, budu, máš, mám, jsi, nevím, doufám, jsem – 1st/2nd person (discussions)
- dneska, zítra, sem, teď, včera, pořád, tady, nějak – adverbs (blogs, discussions)

Notable top keywords from czTenTen12 vs. Hector:

- http, kdyz – poor tokenization, missing diacritics
- již, lze, dále, mohou, zejména, především – standard language (books, news)
- společnosti, oblasti, města, společnost, projektu, řízení, prostředí – society (news)
- zařízení, systém, nabízí, služby, informace – business
- této, tato, tyto, těchto, tohoto – demonstrative adj., standard language (news)

Notable top keywords vs. SYN2000 (Czech National Corpus) [10]:

- Hector vs. SYN2000: taky, teda, ahoj, holky, mám, fakt, moc, sem, dneska, takže, blog, nevím, máš, super, ráda, ahojky (discussions of women)
- czTenTen12 vs. SYN2000: taky, můžete, moc, děkuji, takže, cca, mám, dobrý, opravdu, dle, ahoj, bych, jestli, díky, hodně, super (discussions)
- SYN2000 vs. Hector and czTenTen12: praha, včera, korun, procent, české, vlády, státní, miliónů, zákona, trhu, ministr, ředitel, výstava, společnost, nato, prezident, čtk – standard language, news, Prague

5 Syntactic analysis

5.1 Syntactic functions

A good general corpus should consist mostly of syntactically correct sentences. It is not our intention to filter out syntactically problematic but otherwise quite common and understandable sentences. We aim to detect web garbage such as page navigation and labels, tables consisting of single words or numbers, computer program code samples, keywords used to increase page rank, link spam, artificially generated texts.

Let us declare a nice sentence contains the main syntactic roles – subject and predicate. Although this strict definition does not allow many correct sentences, it surely rules out unwanted content stated above.

Syntactic analysis tool Set [11] was used to carry out the experiment. Subsets of 15 million random sentences from examined corpora were syntactically tagged. Presence of subject and predicate was evaluated for each clause in all sentences. Table 6 reveals czTenTen12 contains slightly more sentences having the subject – predicate couple than Hector. A significant presence of web discussions in Hector is most likely the cause.

Table 6: Ratio of nice clauses in examined corpora – nice clauses (NCL), sentences with all clauses nice (NSEN), sentences with some but not all clauses nice (PNSEN)

CORPUS	NCL	NSEN	PNSEN
Hector	36.6 %	19.0 %	23.7 %
czTenTen12	39.6 %	23.6 %	29.2 %

6 Future work

We plan to implement other intrinsic methods using e.g. language models trained on different corpora: given a language model trained on corpus A we can then measure perplexity of the respective language model using corpus B and vice versa.

Another intrinsic methods to be developed are finding topics in corpora using available tools as e.g. Gensim [12] and measuring homogenity of corpora.

We have already tried some extrinsic methods for comparing corpora: one of them is Word sketch evaluation described in article *How to compare corpora* already submitted to LREC 2014 – the method is based on automatic extraction of good collocations from corpora.

We have also carried out extrinsic method described in [13] for these two corpora. Results will be published in a separate paper soon.

7 Conclusion

We described eight methods which can be used for a general systematic comparison of text corpora. We provided also results of these methods based on comparison of two very large Czech text corpora czTenTen12 and Hector.

The methods are ready to be used and you can download related tools and data from website of Natural Language Processing Centre.[2]

Acknowledgements This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin project LM2010013.

References

1. Baroni, M., Kilgarriff, A., Pomikálek, J., Rychlỳ, P.: Webbootcat: instant domain-specific corpora to support human translators. In: Proceedings of EAMT. (2006) 247–252
2. Suchomel, V.: Recent Czech web corpora. In Aleš Horák, P.R., ed.: 6th Workshop on Recent Advances in Slavonic Natural Language Processing, Brno, Tribun EU (2012) 77–83
3. Spoustová, J., Spousta, M., Pecina, P.: Building a web corpus of czech. (2010)
4. Spoustová, J., Spousta, M.: A high-quality web corpus of czech. In: LREC. (2012) 311–315
5. Kilgarriff, A.: Comparing corpora. International journal of corpus linguistics **6**(1) (2001) 97–133
6. Rayson, P., Garside, R.: Comparing corpora using frequency profiling. In: Proceedings of the workshop on Comparing Corpora, Association for Computational Linguistics (2000) 1–6
7. Kilgarriff, A.: Getting to know your corpus. In: Text, Speech and Dialogue, Springer (2012) 3–15
8. Pomikálek, J., Rychlý, P., Kilgarriff, A.: Scaling to billion-plus word corpora. Volume 41. (2009) 3–13
9. Šmerk, P., Rychlý, P.: Majka – rychlý morfologický analyzátor. Technical report, Masarykova univerzita (2009)
10. Ústav Českého národního korpusu FF UK, Praha: Czech national corpus – SYN2000. Online: http://www.korpus.cz (2000)
11. Kovář, V., Horák, A., Jakubíšek, M.: Syntactic analysis using finite patterns: A new parsing system for czech. In: Human Language Technology. Challenges for Computer Science and Linguistics, Springer (2011) 161–171
12. Řehůřek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks. (2010) 46–50
13. Baisa, V., Suchomel, V.: Large corpora for turkic languages and unsupervised morphological analysis. In: Proceedings of LREC 2012 Workshop on Language Resources and Technologies for Turkic Languages, pp. 28–32. Istanbul, Turkey, 2012.

[2] `http://nlp.fi.muni.cz/projekty/corpora_comparison`

Typos in Czech Corpora

Marek Grác

NLP Centre, Faculty of Informatics
Masaryk University
Botanická 68a, Brno, Czech Republic
grac@fi.muni.cz

Abstract. The extended usage of written corpora not only for manual querying but also for machine learning led to the creation of massive corpora. These corpora are almost solely crawled from the internet and contain texts of various quality. Corpora that contain more typos or ungrammatical texts are more difficult to use for computational linguists and are thus a major obstacle in automatic development. In this paper we attempt to qualify some of existing Czech corpora using manually created wordlist. We will show that building such a list of frequent typos can be done without major investing when agile techniques are used.

Key words: text corpus, errors in text

1 Introduction

In the last few years the popularity of the multi-billion corpora rapidly grew. These corpora are usually built from documents that are crawled almost exclusively of internet. The language of internet differs from one which is mapped by manually compiled corpora. The main negative of corpora based on internet documents is the fact that unlike documents in manually compiled corpora (mainly newspaper, literature, ...) we are working with documents which did not pass any proofreading and some of them e.g. posts in the discussion forums are not intended to be more than a online form of communication where users do not care so much about grammar, typos and other errors. In case of English corpora we usually want to differ between documents written by a native (or qualified) speaker and those who use a mix of English and their native language [4]. For smaller languages, like Czech, we do not have to solve this problem because there are few people who actively write in Czech and are not native speakers. We still need corpora which do not contain that many various errors, for example we need to remove non-Czech documents.

For NLP applications we prefer corpora which contain only generally acceptable language without too many deviations, due to the fact that existing tools have trouble handling even "correct" language. Once we are close to solving this issue, we can try to work with the more difficult case of "real-world" language.

Based on the Czech corpora (e.g. [5], [8]), we can see that quality of corpora obtained by various methods in different times fluctuates. One of the few tests

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2013, pp. 59–62, 2013.

done on Czech corpora is the *the-test* [7] which counts how often is token "the" used in corpora. This works because "the" is not a valid Czech word. Token "the" was used because one of the problems in crawling Czech corpora is obtaining also documents in English were "the" should occur quite frequently. Of course, even that "the" is not a valid Czech word, it should occur in large corpora, mainly in form of snippets of English texts or named entities like movies or bands. The main advantage of this text is the fact that it is very cheap to test an existing corpora. One simple query and we have results. But it tests the presence of English in corpora, what is just a part of a possibly "broken" texts.

2 Building typo language resource

In our research we would like to have a more fine-grained testing. We have decided to create a database of the most frequent Czech typos. Creating such database requires a lot of resources because manual annotation is necessary - as we do not known how to distinguish unknown word from incorrect one. We have decided to reuse techniques from agile development for creating a language resource based on proposals in [2]. The main points which we have to satisfy:

- **have an application for data:** Test quality of corpora to help us remove "broken" sentences or documents, so we can focus on problems of existing tools like morphological desambiguation, chunkers or syntactic parsers.
- **obtain a data to annotate automatically:** Data were obtained from large crawled corpus czTenTen which contains over five billion tokens. Two possible approaches can be used here. We can choose if we want to work with tokens or lemmas. If we decide to go with the lemma, then number of annotation data can be simplified, but we will have problems with unknown words which are guessed by specialized tools. This could results in creating a correct word lemma from ungrammatical token. This was the reason why we have decided to work with tokens directly. We have obtained the most frequent 100,000 tokens which are not in database of morphological analyser majka [6].
- **data have to be annotated in a simple environment:** Checking whether token is a valid one or not, should be (in most of the cases) easy task for native speaker. We have decided to not use context for given token. Annotators used an existing tool SySel [3] which allows them to confirm/deny if presented token is a valid token. Due to use of this existing resource and its simple interface the whole process of annotating data was done in 100 man-hours taken into account that each token was annotated by two different annotators.

After we have obtained a set of tokens that look like typo for annotators, we have selected a subset of these which we have agreed upon from the preselected set it was 32,766 tokens. This means that more than 66% of original data are unknown to used morphological analyser but they are still valid according

to annotators. In the set of typo tokens, we can see several patterns like missing diacritics, tokens in non-latin alphabets or foreign words (mainly English ones). This data could be used for further research if we would like to fix them to correct versions what should be possible even without context for at least a portion of them.

3 Testing quality of corpora

Quality of corpora can be measured from different views but there are just a few of them which are easy to compute automatically. We have selected to use *the-score* metrics [7], a simple metric which measures contamination of the corpus by English words. The *the-score* is the rank of the word "the" in a list of tokens (originally words) sorted by frequency starting from the most frequent one. The higher values should implicate that the corpus is not polluted by English documents. We have run these tests on a selection of available corpora. The first ones DESAM [5] and CBB.blog [1] which represent small manually compiled corpora, SYN2K12 is an example of large balanced corpora and the other ones are taken from the web. At the table 1, as we have deeper user knowledge of these corpora we can generally agree with the-score with exception of CBB.blog. This corpus shows the-score which is similar to web crawled corpora but its language purity is much higher than in czTenTen12. Main reason for such high score is fact that corpus contains album and movies reviews which are often mentioned also by original (English) name.

name of corpus	the-score	absolute frequency
DESAM	9,200	12
CBB.blog	1,418	48
SYN2K12	7,897	18,847
czes2	56	530,289
czTenTen12	1,331	346,706
czTenTen12.clean	2,087	216,940

We have created a new metric based on the ratio of the known typos to all tokens in the corpus. The lower values represents that corpus is cleaner, we expect that there will be a correlation with quality of texts itself. In the table 2, results of this measurement are presented. We can see that the corpora are in similar order as in table 1 with CBB.blog exception which is ranked on the position which can be expected from manually collected data. In this also very interesting to see that typo-ratio of *clean* version of czTenTen12 is almost half of the original version with very similar size. The removed part of corpora has typo-ratio 12.24 % what makes this part of the corpus almost unusable for general usage.

name of corpus	count of typos in thousands	corpus size in millions	typo-ratio
DESAM	1.65	1.04	0.16 %
CBB.blog	2.32	0.81	0.29 %
SYN2K12	15,710	1294	1.21 %
czes2	7,053	465	1.52 %
czTenTen12	55,650	5436	1.02 %
czTenTen12.clean	28,482	5214	0.55 %

4 Conclusions and Future work

This paper introduces a new language resource of non-Czech tokens. The resources are built on corpora data and is highly reliable as each token was confirmed by two independent annotators. We have presented that the *typo-ratio* on large corpora gives similar results to *the-score* but it works better for the small corpora. As it eliminates problem of very small snippets of English named entities in Czech documents.

With the existing resource we can clean existing Czech corpora by removing documents which have bigger *typo-ratio* then is acceptable. In the future we plan to enhance this list with additional data to qualify a reason of including token into our resource.

Acknowledgements This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin project LM2010013.

References

1. Marek Grác. Case study of bushbank concept. In *Pacific Asia Conference on Language, Information, and Computation, PACLIC*, 2011.
2. Marek Grác. *Rapid Development of Language Resources*. PhD thesis, Masaryk University, 2013.
3. Marek Grác, Adam Rambousek. Low-cost ontology development. In *GWC 2012 6th International Global Wordnet Conference*, 2012.
4. Simone Muller. *Discourse markers in native and non-native English discourse*, volume 138. John Benjamins, 2005.
5. Karel Pala, Pavel Rychlý, and Pavel Smrž. Desam – annotated corpus for Czech. In *SOFSEM'97: Theory and Practice of Informatics*. Springer, 1997.
6. Pavel Šmerk. *Towards morphological disambiguation of Czech*. PhD thesis, Ph. D. thesis proposals, Masaryk University, 2007.
7. Vít Suchomel. Recent Czech Web Corpora. In *Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN*, 2012.
8. Vít Suchomel and Jan Pomikálek. Efficient web crawling for large text corpora. In *Proceedings of the Seventh Web as Corpus Workshop (WAC7)*, 2012.

Fast Construction of a Word↔Number Index for Large Data

Miloš Jakubíček, Pavel Rychlý, and Pavel Šmerk

Natural Language Processing Centre
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
{jak,pary,xsmerk}@fi.muni.cz

Abstract. The paper presents a work still in progress, but with promising results. We offer a new method of construction of word to number and number to word indices for very large corpus data (tens of billions of tokens), which is up to an order of magnitude faster than the current approach. We use HAT-trie for sorting the data and Daciuk's algorithm for building a minimal deterministic finite state automaton from sorted data. The latter we reimplemented and our new implementation is roughly three times faster and with smaller memory footprint than the one of Daciuk. This is useful not only for building word↔number indices, but also for many other applications, e.g. building data for morphological analysers.

Key words: word to number index, number to word index, finite state automata, hat-trie

1 Introduction

The main area of interest of this work lies in computer processing of large amounts of text (text corpora) with heavy annotation using a corpus management system that provides the user with fast and efficient search in the text data. The primary usage focuses on research in natural language processing, both from a more linguistically motivated or more language engineering oriented perspective, and on the exploitation of these tools in third-party industry applications in the domain of information systems and information extraction.

For any such system to perform well on large data, complex indexing and database management system must be in place – and so is this the case of the Manatee corpus management system which was the subject of our experiments.

Any reasonable indexing of text data by means of individual words (tokens in text) starts with providing a fast word-to-number and number-to-word mapping that allows to build the database indices on numbers, not words. This enables faster comparison, search and sort, and is also much more space efficient.

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2013, pp. 63–67, 2013.

In this paper we particularly focus on constructing such word↔number mapping when indexing large text corpora. We first describe the current procedure used within the Manatee corpus management system and discuss its deficiencies when processing very large input data – here by large we refer to text collections containing billions of tokens. Then we present a new implementation exploiting a HAT-trie structure and provide an evaluation showing a significant speedup in building the mapping and henceforth also indexing of the whole text corpus.

2 Word↔number mapping in Manatee

2.1 Lexicon structure

The corpus management system Manatee uses the concept of a *lexicon* for providing the word↔number mapping, thus implementing two basic operations:

- `str2id` – retrieving an ID according to its word string
- `id2str` – retrieving a word according to its ID

The lexicon is constructed from source data when compiling all corpus indices and consists of three data files:

- `.lex` file – a plain text file containing the word strings separated by a NULL byte, in the order of their appearance in the source text.
- `.lex.idx` file – a fixed-size (4 B) integer index containing offsets to the `.lex` file. The `id2str` operation for a given ID n is implemented by retrieving the string offset at the $4 \cdot n^{\text{th}}$ byte in this file and reading at that offset in the `.lex` file (until the first NULL byte).
- `.lex.srt` file – a fixed-size (4 B) integer index containing IDs sorted alphabetically. The `str2id` operation for a given string s is implemented by binary search in this file (retrieving strings for comparison as described above).

2.2 Building the lexicon

When compiling corpus indices, new items are added to the lexicon in the order as they appear in the source texts and the lexicon is used for retrieving the ID of items already added to the lexicon. The system keeps two independent caches to speed up the process: one contains recently used lexicon items, another items that were recently added. As soon as the latter one reaches some threshold size, the cache is cleared – written to the lexicon and the lexicon must be re-sorted. This is a significant time bottleneck and as the lexicon grows, the time spent on its sorting grows rapidly too.

For more than two decades the data sizes of text corpora allowed not to care about the compilation time much, it was mainly the runtime of the database (i.e. querying) that mattered and that was subject to development. As data sizes of current text corpora grow to dozens of billions of tokens [1], the compilation time is being counted in days and starts to be an obstackle for data maintenance. Therefore we considered alternative implementations to overcome this issue.

3 Experiments and results

We demonstrate our results on three sets of corpus data. As can be seen in the Table 1, the sets differ not only in size: Tajik language uses Cyrillic, which means the words are two times longer (counted in bytes) only due to the encoding, and the French corpus from OPUS project[1] obviously uses rather limited vocabulary.

Table 1: Data sets used in the experiments.

data set	size	words	unique	size	language
100M	1148 MB	110 M	1660 k	31 MB	Tajik
1000M	5161 MB	957 M	1366 k	14 MB	French
10000M	69010 MB	12967 M	27892 k	384 MB	English

HAT-trie [2] is a cache-conscious data structure which combines trie and hash and allows sorted access. In general, for indexing the natural language strings, it is among the best solutions regarding both time and space. We used it[2] to create files described in the previous section. Results in the Table 2 show that hat-trie is up to an order of magnitude faster than the current solution `encodevert`.

Table 2: Comparison of encodevert and hat-trie.

	`encodevert`		**hat-trie**		
data set	**time**	**memory**	**time**	**memory**	**output size**
100M	3:11 m	0.44 GB	26.5 s	0.12 GB	44 MB
1000M	23:01 m	0.40 GB	2:21 m	0.04 GB	25 MB
10000M	7:38 h	0.98 GB	44:37 m	0.78 GB	607 MB

If a server is to support concurrent queries to multiple corpora, the indices for these corpora generated by `encodevert` (or now hat-trie) has to be loaded in memory. The last cell in the Table 2 indicates that for very large corpora it can consume a lot of memory, thus we tried to reduce this data. We used Jan Daciuk's `fsa` tools[3] which are able to convert a sorted set of strings to a deterministic acyclic finite state automaton usable for (static) minimal perfect hashing, i.e. string↔number translation, where the number is a rank in the sorted set of strings. We started with version 0.51 compiled with STOPBIT and NEXTBIT options, but because the original tools were rather memory and time consuming, we reimplement it and significantly reduce both time and space

[1] `http://opus.lingfil.uu.se/`, mostly legal texts.
[2] We use a free implementation from `https://github.com/dcjones/hat-trie`.
[3] `www.eti.pg.gda.pl/katedry/kiw/pracownicy/Jan.Daciuk/personal/fsa.html`

required for the automaton costruction (we did not change the output format). The Table 3 compares the results of the original version of `fsa_ubuild` acting on unsorted corpus data and our new approach. The last column shows new sizes of indices.

The last table, Table 4, compares only the original and reimplemented algorithm for sorted data. The hat-trie sort column are the costs of using hat-trie as a data pre-sort. Two results are obvious: firstly, having such an effective sort algorithm, to sort data and then use the algorithm for sorted data is always better than `fsa_ubuild`, secondly, to reduce the used memory it is better to flush sorted data to hard disk before fsa construction, as the time penalty is minimal.

Table 3: Building automata for perfect hashing from unsorted data.

	`fsa_ubuild`		**hat + new fsa**		
data set	**time**	**memory**	**time**	**memory**	**output size**
100M	*failed*		31.7 s	0.09 GB	15 MB
1000M	15:48 m	0.11 GB	2:34 m	0.06 GB	11 MB
10000M	7:44 h	31.01 GB	1:08 h	1.47 GB	363 MB

Table 4: Sorting data and building automata for perfect hashing from sorted data.

	hat-trie sort		`fsa_build`		**new fsa**	
data set	**time**	**memory**	**time**	**memory**	**time**	**memory**
100M	28.4 s	0.06 GB	12.4 s	0.21 GB	4.2 s	0.03 GB
1000M	2:51 m	0.04 GB	5.6 s	0.11 GB	1.8 s	0.03 GB
10000M	59:16 m	0.77 GB	35:15 m	27.07 GB	9:36 m	0.71 GB

4 Future work

The presented results are only preliminary, as it is only a proof of concept, not a final solution. We plan to further reduce both time and space of the automata construction, as well as their final size. The final automaton can be built directly from the input data which would cut the required memory to less than two thirds. The use of UTF-8 labels would reduce the space even further. We also want to employ some variable length encoding of numbers and addresses (similar to [3], but computationally simpler one). We suspect Daciuk's "tree index" used to discovering already known nodes during the automaton construction to be slow for large data and we hope that simple hash will decrease the compilation time significantly at the acceptable expense of some additional space.

Acknowledgements This work has been partly supported by the Ministry of Education of CR within the Lindat Clarin Center LM2010013.

References

1. Pomikálek, J., Rychlý, P., Jakubíček, M.: Building a 70 billion word corpus of English from ClueWeb. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12). (2012) 502–506
2. Askitis, N., Sinha, R.: Hat-trie: a cache-conscious trie-based data structure for strings. In: Proceedings of the thirtieth Australasian conference on Computer science-Volume 62, Australian Computer Society, Inc. (2007) 97–105
3. Daciuk, J., Weiss, D.: Smaller representation of finite state automata. Theoretical Computer Science **450** (2012) 10–21

Part IV

Language Modelling and Machine Translation

Expanding Translation Memories: Proposal and Evaluation of Several Methods

Vít Baisa, Josef Bušta, and Aleš Horák

NLP Centre, Faculty of Informatics
Masaryk University
Botanická 68a, Brno, Czech Republic
{xbaisa, xbusta1, hales}@fi.muni.cz

Abstract. Translation memories used in Computer-aided translation (CAT) systems are the highest-quality resources of parallel texts since they are carefully prepared and checked by professional human translators. On the other hand, they are quite small when compared with other parallel data sources. In this paper, we propose several methods for expanding translation memories using both language-independent and language-specific, linguistically motivated approaches with regard to preserving their high translational accuracy. We first briefly describe the methods and then we provide a detailed description and preliminary evaluation for two of them.

Key words: translation memory, computer-aided translation, expanding translation memories

1 Introduction

Translation memory is a set of *translation pairs* containing segments from text documents which were previously manually translated by human translators. These segments (which might refer to sentences, paragraphs, list items, headings, titles etc.) can be then reused within the CAT[1] process and save both time and effort of human translators.

Since translation memories are built manually by human expert translators, they are a) relatively small in comparison with other parallel resources, e.g. *OPUS* [1], *Europarl* [2], or *JRC-Acquis* [3], and b) usually are not available freely as being a property of a professional translation companies, despite some exceptions as e.g. *MyMemory* [4].

But at the same time (as for many other NLP[2] related tasks) this holds good: the bigger is a translation memory, the better is the CAT process. Where better here means faster, of higher quality etc.

The purpose of this paper is thus to present several methods for expanding translation memories using available resources and tools. These methods

[1] Computer-aided translation
[2] Natural language processing

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2013, pp. 71–78, 2013.

exploit different amount of linguistic knowledge: from purely statistical and *n*-gram based to syntactic-semantic processing of the input translation memories.

The first method with its variant is described in detail in this paper and its preliminary evaluation is given. We are interested mainly in English-Czech and Czech-English translation pairs.

2 Related Work

Translation memories (TM) are understudied resources in the realm of NLP. They are often presented [5] within a closely related field: *example-based machine translation* (*EBMT*) which uses a similar approach as CAT systems do – reusing samples of previously translated texts.

The TM related papers mainly focus on algorithms for searching, matching and suggesting segments within CAT systems [6] but not much work was devoted to the problem of expanding translation memories.

In [7], the authors have attempted to build translation memories from Web since they found that human translators in Canada use Google search results even more often than specialized translation memories. That is why the research team at the National Research Council of Canada developed a system called *WeBiText* for extracting possible segments and their translations from bilingual webpages. They state an important notice: it is always better to provide translators with a list of possible translations and let them find the correct one than to have nothing prepared. In other words, it is easier and faster for the translators to look up a good translation than to make up their own translation from scratch. Also, it is very important that the correct translation must be between the first 10 or 20 items in the suggested list.

WeBiText system successively tested two approaches: an on-demand version which took a user query (expression) and then asked a search engine for all results. For these results it then tried to find links to their mutation in a target language. This approach was very slow so they resorted to another approach: an off-line version with precompiled results.

In the study [8], the authors exploited two methods of segmentation of translation memories. Their approach is probably the most similar to our subsegment combination method presented below. The main difference is that we use statistical methods of the phrase-based machine translation (PBMT) approach [9] for extraction of new translation pairs of segments.

[10] describes a method of subsegmenting of translation memories which deals with the principles of EBMT. The authors of this study created an on-line system TransSearch [11] for searching possible translation candidates within all subsegments in already translated texts. These subsegments are linguistically motivated – they use a text-chunker to extract phrases from the Hansard corpus, a text corpus containing the Canadian parliamentary debates from 1803 to the present time.

3 Expanding Translation Memories

In following text, we describe four methods which deal with TM enlarging in this way: for a given translation memory TM and a document D to be translated, take TM and try to enlarge it for the purpose of translation of the document D. For this, either various additional resources (parallel corpora) and tools for generalising available data (morphological, syntactic and semantic analysis) are used extensively.

3.1 Method A – Subsegment Combination

The first method uses a parallel corpus (OPUS [1]) and trains a translation model M^t with the *GIZA++* tool [12,13]. Then it takes TM and extracts all consistent phrases from all aligned segments in TM using appropriate word matrices (see Figure 1) yielding TM_{sub}, a translation memory of subsegments.

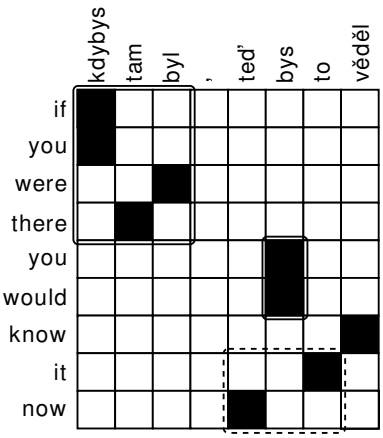


Fig. 1: Word matrix for two aligned sentences / segments.

Word matrices are built directly from the M^t translation model. M^t defines conditional translation probabilities between all pairs of source and target words from the parallel corpus (OPUS). E.g. $p(\text{pes}|\text{dog}) = 0.79$ means that there is 79% probability that the source word *dog* (English) will be translated into the target word *pes* (Czech).

If a pair of two words has sufficiently high probability (higher than a threshold) then in the corresponding word matrix there is a black cell for the pair (see Figure 1). The probability threshold was set experimentally to 0.01 and will be experimentally tuned in the future.

All consistent phrases are then extracted from the word matrices. Consistent phrase is pair of two ranges of words: words from i to j from a source sentence s (s_{ij}) and words from k to l from the corresponding target sentence t (t_{kl}). We

regard s_{ij} and t_{kl} to be consistent when all translations (represented as black cells) of words between the positions i and j are inside the interval from k to l in the target sentence.

Examples of two consistent phrases are displayed in Figure 1 and are outlined with solid line. The third dashed outlined phrase is an inconsistent phrase since it violates the condition. The extracted consistent phrases then form the TM_{sub} translation memory of subsegments.

The memory of subsegments then serves as a basis for building TM_{exp}, the expanded translation memory by combining subsegments that partially match with (sub)segments from the translated document. Each new segment in TM_{exp} must be created as a result of one of the following operations:

a) **join** – new segments are built by concatenating two other segments from TM and TM_{sub}
b) **substitute** – new segments can be created by replacing a part of one segment with a whole of another (sub)segment from TM and TM_{sub}.

An evaluation of the subsegment combination method is presented in section 4.

3.2 Method B – Subsegment Lexicalization

This method is a generalisation of the previous method using linguistic preprocessing: all segments are tokenized and lemmatized and the searching and matching operations work on lemmata. The two corresponding combination operations are now:

a) **ljoin** – concatenation of two different segments from TM and TM_{sub} but this time on lemmata; when concatenating into new resulting segments, appropriate word form (case, gender and number) is generated in the target language
b) **lsubstitute** – substitution of a part of target segment with another segment but again using lemmata and generating proper word forms with correct case, gender and number in the target language

With this method we expect increasing the recall (coverage) but at the same time not decreasing the translation accuracy of original segments from TM. So it is partially rule-based method.

3.3 Method C – Machine Translation of Subsegments

The process of this method follows the previous methods A and B with two other combining operations:

a) **substtran** – new segment is created by translating its part by a freely available machine translation systems

b) **lsubsttran** – combination of **substran** and **lsubstitute** operations – the translation is done for segment parts (phrases) in basic form and the translation result is then transformed into correct word forms.
For example, the Czech phrase *modré knížce* ("to [the] blue book") has its basic phrase form *modrá knížka* ("blue book") which is different from phrase *modrý knížka* where all words from the phrase are in the base form: gender agreement must hold for base forms of phrases.

Example: Let us have a sentence s_s in TM: *Návod na použití desinfekčního přípravku najdete na konci této brožury* together with its proper translation t_t: *You can find instructions for use of disinfectant at the end of this brochure*, and a sentence s_d in D to be translated: *Návod na použití kartáče na vlasy najdete na konci této brožury*. Given that subsegment *kartáče na vlasy* is not in previously built TM_{sub} we need to get translation of it. Google Translate gives us *hairbrush* as translation of the base form. So the only thing to do is to identify the subsegment, put it into its base form, translate it with some of MT systems and substitute the appropriate part of target segment to be able to translate the whole sentence s_s.

3.4 Collocation-Based Filtering to Expanding TM

The previous methods often generate too many candidates for the TM_{exp} expanded translation memory. The CAT systems offer the possibility to sort the translation memory *matches* expressed as a percentage of the correspondence between the segment from TM and segment from the translated document. We thus use a value of *collocability* of the target segment phrase as a base for this percentage. The collocability is a number showing how common the phrase and its parts are in the language. In this way, we prefer those translations that correspond to frequently used phrases.

4 Evaluation

Authors of [10] reported 28% coverage with precision 37% for 100 test sentences. For evaluation purposes, we have used a different test data so it is not straightforward to compare the two results.

As test data we have used a sample of translation memory TM^s and an example document D^s provided by one of the biggest Czech translation company.

The presented results have been obtained directly from the pre-translation analysis of the MemoQ CAT system.[3] The numbers express how many segments from the document D^s can be translated automatically by MemoQ. The automatic translation is done on the segment level and even on lower levels: on levels of subsegments. Various matches on lines in the table correspond to these

[3] `http://kilgray.com/products/memoq`

Table 1: TM analysis for the first phase of the subsegment combination method A, without the *join* operation

	TM^s				TM_{sub}				TM^s+TM_{sub}			
Match	**Seg**	**wrds**	**chars**	**%**	**Seg**	**wrds**	**chars**	**%**	**Seg**	**wrds**	**chars**	**%**
100%	23	128	813	0.4	2	5	23	0.01	25	133	836	0.38
95–99%	45	185	1 130	0.5	296	363	1 924	1.03	305	480	2 620	1.37
85–94%	4	21	155	0.1	20	54	337	0.15	24	75	492	0.21
75–84%	42	208	1 305	0.6	85	237	1 474	0.67	102	358	2 258	1.02
50–74%	462	1 689	10 293	4.8	772	4 031	24 826	11.47	784	4 449	27 370	12.66
any match	576	2 231	13 696	6.4	1 175	4 690	28 584	**13.33**	1 240	5 495	33 576	**15.64**

Table 2: TM analysis for the subsegment combination method A including the **join** operation

	TM^s				$TM_{subjoin}$				TM^s+$TM_{subjoin}$			
Match	**Seg**	**wrds**	**chars**	**%**	**Seg**	**wrds**	**chars**	**%**	**Seg**	**wrds**	**chars**	**%**
100%	23	128	813	0.4	2	5	23	0.01	25	133	836	0.38
95%–99%	45	185	1 130	0.5	296	363	1 924	1.03	305	480	2 620	1.37
85%–94%	4	21	155	0.1	20	54	337	0.15	24	75	492	0.21
75%–84%	42	208	1 305	0.6	88	255	1 565	0.73	102	358	2 258	1.02
50%–74%	462	1 689	10 293	4.8	787	4 256	26 136	12.11	798	4 629	28 423	13.17
any match	576	2 231	13 696	6.4	1 193	4 933	29 985	**14.03**	1 254	5 675	34 632	**16.15**

sublevels – 100% match corresponds to the situation when a whole segment from D^s can be translated using a segment from the available TM. Translations of shorter parts of the segment are then matches lower than 100%.

The analysis results provided by CAT systems are usually used for estimating the amount of work needed for the (human) translation of a given document and subsequently for estimating the price of the translation work. The higher number of segments which can be translated automatically, the lower is the price of the translation work. That is why the translating companies aim at the highest possible matches. Such result can be achieved with bigger translation memories which have higher coverage and it is also the aim of this paper.

The results deserve more detailed description. The analysis table columns are: **Match** – type of match between TM^s and D^s, **Seg** – number of segments identified in D^s, **wrds** – number of source words which are covered (translatable) by the TM^s, **chars** – number of source characters and **%** – percentage of coverage for the type of match in the first column.

In the evaluation process, we have tested the translation on a document with 4,563 segments, 35,142 words and 211,407 characters.

In the measurements, we have split the analysis for the subsegment combination method to the values obtained by a) the TM_{sub} translation memory of subsegments (consistent MT phrases from OPUS), i.e. without the *join* operation, see Table 1, and b) the $TM_{subjoin}$ memory further expanded by means of

the *join* operation,[4] see Table 2. The results in both tables display three subtables: the original analysis with the TM^s input translation memory, the analysis using only the new $\mathsf{TM}_{sub}/\mathsf{TM}_{subjoin}$ translation memory, and the most practical combination of $\mathsf{TM}^s + \mathsf{TM}_{sub}/\mathsf{TM}_{subjoin}$.

The most important are the boldface numbers at the bottom of the tables expressing the sum percentage of any of the translation match. The result in Table 1 is 15.64%. This means that with the TM_{sub} we can increase the coverage of TM^s by more than 9% which is a substantial improvement over using TM^s alone.

The results of the *join* operation in Table 2 further increase the total percentage of matches to 16.15%. When compared to the TM_{sub} results, this represents quite low improvement of 0.5%. The problem currently lies in the coverage of the current prototype implementation of the *join* operation. In the evaluated document, the $\mathsf{TM}_{subjoin} - \mathsf{TM}_{sub}$ phrases cover only 122 subsegments of the document, which is too low to generate a substantial increase in translation matches. With regard to the coverage, the subsegment lexicalization method B should also provide more interesting results. This remains, however, still to be implemented and evaluated.

5 Conclusion

In this paper, we have described several novel methods for expanding translation memories. We showed that we can effectively generate new high quality translation pairs which increase the efficiency of computer-aided translation by means of purely computational linguistically motivated techniques.

The presented results show and improvement of 10 percent in the translation matches, which already corresponds to substantial economic savings in the translation process.

In the future work, we will concentrate on the evaluation of the other presented methods and their application in a selected CAT system.

Acknowledgements This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin project LM2010013.

References

1. Tiedemann, J.: Parallel Data, Tools and Interfaces in OPUS. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012. (2012) 2214–2218 `http://opus.lingfil.uu.se`.
2. Koehn, P.: Europarl: A parallel corpus for statistical machine translation. In: MT summit. Volume 5. (2005) `http://www.statmt.org/europarl`.
3. Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufis, D., Varga, D.: The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. arXiv preprint cs/0609058 (2006)

[4] i.e. $\mathsf{TM}_{sub} \subset \mathsf{TM}_{subjoin}$

4. Trombetti, M.: Creating the world's largest translation memory. In: MT Summit. (2009) http://mymemory.translated.net.
5. Planas, E., Furuse, O.: Formalizing translation memories. In: Machine Translation Summit VII. (1999) 331–339
6. Planas, E., Furuse, O.: Multi-level similar segment matching algorithm for translation memories and example-based machine translation. In: Proceedings of the 18th conference on Computational linguistics-Volume 2, Association for Computational Linguistics (2000) 621–627
7. Désilets, A., Farley, B., Stojanovic, M., Patenaude, G.: WeBiText: Building large heterogeneous translation memories from parallel web content. Proc. of Translating and the Computer **30** (2008) 27–28
8. Nevado, F., Casacuberta, F., Landa, J.: Translation memories enrichment by statistical bilingual segmentation. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004. (2004)
9. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, Association for Computational Linguistics (2003) 48–54
10. Simard, M., Langlais, P.: Sub-sentential exploitation of translation memories. In: Machine Translation Summit VIII. (2001) 335–339
11. Macklovitch, E., Simard, M., Langlais, P.: TransSearch: A Free Translation Memory on the World Wide Web. In: Proceedings of the Second International Conference on Language Resources and Evaluation, LREC 2000. (2000)
12. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational linguistics **29**(1) (2003) 19–51
13. Och, F.J.: Giza++ software. http://www.statmt.org/moses/giza/GIZA++.html (2003)

Methods for Detection of Word Usage over Time

Ondřej Herman and Vojtěch Kovář

Natural Language Processing Centre
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
{xherman1,xkovar3}@fi.muni.cz

Abstract. From a natural language corpus, word usage data over time can be extracted. To detect and quantify change in this data, automatic procedures can be employed.
In this work, I describe the application of ordinary and robust regression methods to time series extracted from natural language corpora.

Key words: word usage, time series, regression methods, Theil-Sen estimator, Mann-Kendall test

1 Introduction

Historically, linguists used to characterize languages based on their own experience and introspection. This methodology can only reflect the nature of an idealized, subjective model, which is inherently frozen in time, unlike the empiric reality of an everyday speech act.

The recent development of large corpora allows us to have a convenient and easily quantifiable view of language change based on actual evidence. The amount of this data is too large to sift through manually, so having a way to summarize it and pinpoint interesting behavior is desirable.

2 Time series analysis

A time series is a sequence of discretely spaced observations (x_i, y_i), where y_i is the observation for the time period x_i. In the following text, x_i represents a period of time and y_i the amount of appearances of a word over x_i and n is the amount of samples.

2.1 Linear regression

In simple linear regression, it is assumed that the true relationship between two variables, x and y, is linear: $y_i = a + bx_i$. We are trying to estimate the unknown constants a, the slope, and b, the intercept. The values of y_i are not

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2013, pp. 79–85, 2013.

exactly known[1]: $y'_i = y_i + \epsilon_i$, where ϵ is an unpredictable error component and y'_i is the value observed at x_i.

To estimate the values of a and b from a set of observations, the method of least squares [1,2] can be employed. That is, $\hat{a}$ and $\hat{b}$ such that the sum of squared errors $e = \sum_{i=1}^{n} \epsilon_i^2$ is minimal are to be found:

$$\hat{b} = \frac{\sum_{i=1}^{n} (y'_i - \bar{y}')(x_i - \bar{x})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \tag{1}$$

$$\hat{a} = \bar{y}' - \hat{b}\bar{x} \tag{2}$$

where $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$ and $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$.

2.2 F-test

Even though the estimated parameters $\hat{a}$ and $\hat{b}$ are the best ones in the sense that they minimize the sum of squared errors, the chosen model might not actually describe the observations well. Namely, it is desirable to ensure that the slope of the regression line $\hat{b}$ is non-zero, and that its estimated value is significant compared to the random fluctuations present in the data. That is, the hypotheses to be tested are [1]

$$H_0 : \hat{b} = 0 \tag{3}$$

$$H_1 : \hat{b} \neq 0 \tag{4}$$

One way to obtain a test statistic for (3) is the F-test:

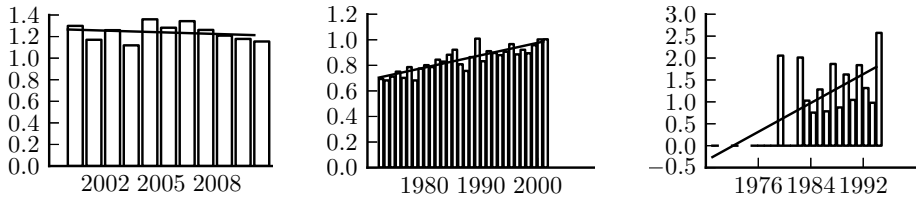
$$F_0 = \frac{\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 / (n-2)} \tag{5}$$

Assuming that the null hypothesis holds, F_0 follows the F distribution with 1 and $n-2$ degrees of freedom, therefore the series is considered to exhibit a statistically significant trend when $|F_0| > F_{1-\alpha,1,n-2}$ and the null hypothesis is rejected.

The series shown in Figure 1(a) does not show any evidence of trend. On the other hand, the series in Figure 1(b) shows a very significant trend. According to the result of the F-test in the case of the series shown in Figure 1(c) also exhibit a trend, but its steepnes in this case seems to be caused by the limited volume of text contained in the early years sampled by the corpus and the resulting non-normality of the data.

[1] It is assumed that the values of x_i are exactly known. Errors-in-variables models do away with this assumption.

[3] The unit of y is the logarithm of the relative frequency per million words, for the reasons explained in 2.3



(a) Oxford English Corpus, $p = 0.514$[3]

(b) Google ngrams, $p = 4.3 \times 10^{-10}$

(c) British National Corpus, $p = 0.009$

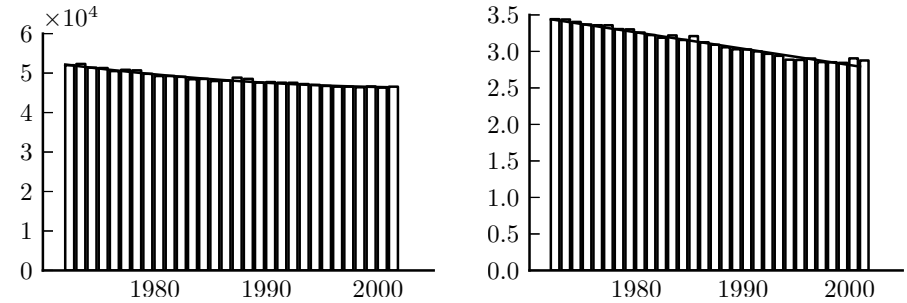
Fig. 1: Linear regression models and the respective p-values obtained using the F-test calculated for the word 'carrot'

2.3 Weighted linear regression

It is possible to extend the least squares method to fit a higher degree polynomial to the data, and also to weight the samples to account for heteroscedasticity[4]. As discussed in [3], this does not provide a significant improvement over the ordinary least squares.

The adjusted coefficient of determination R^2_{adj} [1] can be used to find a suitable degree of the polynomial to fit to the time series. For most of the series examined, the value of R^2_{adj} reaches the maximum for quadratic polynomials. Applying a logarithmic transformation linearizes the regression line, as can be seen in Figure 2. Treating the models as multiplicative therefore yields better results.

In almost all other cases, the higher-order models do not accurately describe the time series.



(a) the original series

(b) log-transformed series

Fig. 2: 'the' from Google Ngrams

[4] In heteroscedastic data, the sample variances are not equal.

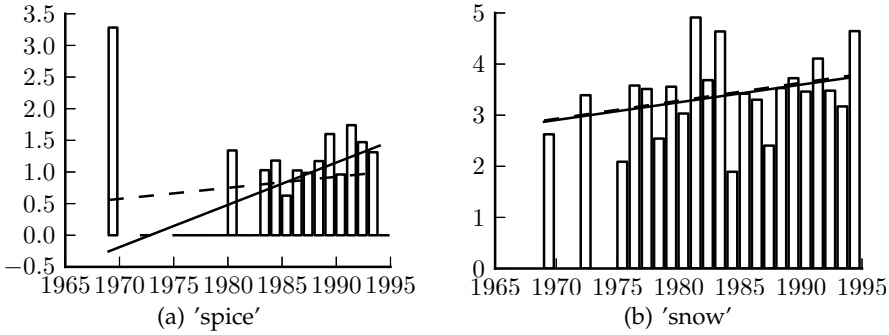
2.4 Theil-Sen estimator

The least squares methods are based on some assumptions that cannot always be met in practice. Namely that the error terms are normally distributed with known variances and mean zero. Rank-based robust methods do away with these requirements and are also less sensitive to the presence of outliers.

The Theil-Sen estimator [4,5,6,7] is a statistic used to estimate the slope of the regression line. It is model-free and non-parametric. The resulting estimate is a linear approximation of the trend line.

The Theil-Sen estimator is defined as the median of the pairwise slopes of the samples[8]:

$$\hat{\beta}_{ts} = \operatorname{med} \frac{y_i - y_j}{x_i - x_j}, \quad i \neq j \tag{6}$$



(a) 'spice' (b) 'snow'

Fig. 3: Behavior of the Theil-Sen estimator for words encountered in the British National Corpus

As shown in Figure 3(a), outliers can easily confuse the ordinary least squares estimator represented by the dashed line, while the Theil-Sen estimator is able to ignore them and estimate the trend better.

On lower quality data, this estimator provides superior estimates of the slope compared to standard regression models. Another benefit is that it does away with the assumption that the data follows a predetermined model, so any monotonic trend can be estimated, therefore it works just as well even on non log-transformed data.

2.5 Mann-Kendall test

To test the significance of a model obtained using the Theil-Sen estimator, the Mann-Kendall test statistic [2,9] can be used:

$$S = \sum_{i=1}^{n} \sum_{j=1}^{i} \operatorname{sgn}(x_i - x_j) \operatorname{sgn}(y_i - y_j) \tag{7}$$

A top score of $S = \binom{n}{2}$ indicates that the series is increasing everywhere while $S = -\binom{n}{2}$ means that the series is decreasing.

Under the null hypothesis S has the following properties[9]:

$$E[S] = 0 \tag{8}$$

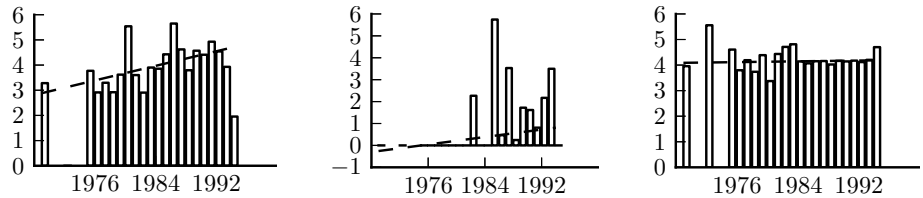
$$V[S] = \frac{n(n-1)(2n+5) - \sum_{i=1}^{n} t_i(i)(i-1)(2i+5)}{18} \tag{9}$$

where t_i is the number of tied values in the i-th group[5]

The standardized[6] Z statistic is computed as

$$Z = \begin{cases} \frac{S-1}{\sqrt{V[S]}} & S > 0 \\ 0 & S = 0 \\ \frac{S+1}{\sqrt{V[S]}} & S < 0 \end{cases}$$

The null hypothesis is to be rejected if $|Z| \geq u_{1-\frac{\alpha}{2}}$ at the significance level of α, where u_a is the quantile function of the standard normal distribution.



(a) 'oil', $p = 0.021$ (b) 'disk', $p = 0.009$ (c) 'slow', $p = 0.821$

Fig. 4: Words from the British National Corpus tested using the Mann-Kendall test with the trend line fitted using the Theil-Sen estimator

While a weighted linear model does not fit the series in the Figure 4(a) well (F-test $p = 0.24$), the p-value obtained using the Mann-Kendall test is considerably more significant. For the series in the Figure 4(b), the situation is similar: the linear model tests at $p = 0.67$. Interestingly, the slope calculated using the Theil-Sen estimator is, in this case, zero. No trend is found in the series in 4(c) by any of the methods. On well-behaved series the behavior of this test is comparable to the standard linear F-test.

The significance test based on Spearman's ρ was also examined [3]. It behaves very similarly as the Mann-Kendall test [9,8,10].

[5] For example, the sequence [1,2,2,1,3,4,5,1,5,5] has 3 tied groups of lengths 3, 2 and 3.

[6] This statistic is only approximately normal.

3 Future work

The relationship between the word usage frequency and time is not inherently polynomial and would probably be better modeled as a sequence of possibly discontinuous linear segments.

Determining if and at which points the behavior of a time series changes is a well studied problem with a large body of research results available, such as [11], [12], [13] or [14]. These methods build on the framework described in this document and are likely to model the time series extracted from natural language corpora better than a single linear function.

4 Conclusion

The methods contained in this text were described with the potential application to the data from the Oxford English Corpus and the British National Corpus in mind.

Even though the ordinary regression models applied to log-transformed series work quite well, their use has more drawbacks than the robust methods have.

The most suitable method seems to be the Theil-Sen slope estimator, along with the Mann-Kendall or Spearman's ρ tests to investigate a possible trend present in the word usage data.

Acknowledgements This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin project LM2010013.

References

1. Montgomery, D., Johnson, L., Gardiner, J.: Forecasting and time series analysis. 2nd edition. McGraw-Hill (1990)
2. Forbelská, M.: Stochastické modelování jednorozměrných časových řad (Stochastic Modelling of one-dimensional time series, in Czech). Masarykova univerzita (2009)
3. Herman, O.: Automatic methods for detection of word usage in time (2013)
4. Rousseeuw, P.J., Leroy, A.M.: Robust regression and outlier detection. John Wiley & Sons, Inc., New York, NY, USA (1987)
5. Matoušek, J., Mount, D.M., Netanyahu, N.S.: Efficient randomized algorithms for the repeated median line estimator. Algorithmica **20** (1998) 136–150
6. Wilcox, R.R.: Fundamentals of Modern Statistical Methods: Substantially Improving Power and Accuracy. 2nd ed. Springer New York (2010)
7. Wilcox, R.: A note on the Theil-Sen regression estimator when the regressor is random and the error term is heteroscedastic. Biometrical Journal **40**(3) (1998) 261–268
8. Onoz, B., Bayazit, M.: The power of statistical tests for trend detection. Turkish Journal of Engineering and Environmental Sciences (27) (2003)
9. Neeti, N., Eastman, J.R.: A contextual Mann-Kendall approach for the assessment of trend significance in image time series. Transactions in GIS **15**(5) (2011) 599–611

10. Yue, S.: Power of the Mann Kendall and Spearman's rho tests for detecting monotonic trends in hydrological series. Journal of Hydrology **259** (2002) 254–271
11. Guralnik, V., Srivastava, J.: Event detection from time series data. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. KDD '99, ACM (1999) 33–42
12. Sinn, M., Ghodsi, A., Keller, K.: Detecting Change-Points in Time Series by Maximum Mean Discrepancy of Ordinal Pattern Distributions. ArXiv e-prints (2012)
13. Liu, S., Yamada, M., Collier, N., Sugiyama, M.: Change-Point Detection in Time-Series Data by Relative Density-Ratio Estimation. ArXiv e-prints (2012)
14. de Jong, P., Penzer, J.: Diagnosing shocks in time series. Journal of the American Statistical Association **93**(442) (1998)

Towards the Realistic Natural Language Representations

Petr Sojka

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
sojka@fi.muni.cz

Abstract. This essay suggests a way to derive a natural language representation from textual corpora into the connectionist, continuous representations. Based on the lexical priming theory and psycholinguistic evidence we discuss benefits and potential of alternative representations inspired by connectionist approaches towards *computation* of personalized mental lexicon from and during empirical language usage.

Key words: natural language representation, priming, lexical priming, semantic priming, data discretization, language modelling, representation of meaning, personal mental lexicon, empirical linguistics

On resiste a l'invasion des armees; on ne resiste pas a l'invasion des idees.
(Victor Hugo (1802–1885))

1 Striving to Getting an Insight

Linguists try to get insight into language communication by naming the language phenomena. They named morphology, syntax, semantics and pragmatics as levels of natural language processing and understanding, usually hoping that solving lower level is a prerequisite to tackle the higher one. They pose questions like "Do Word Meaning Exist?" [6]. They try to embrace the *knowledge of a language* as a set of grammars, dictionaries and the battery of rules to model the forms of communication via natural language.

Psycholinguistics is concerned with the ability of human brain to understand and generate language. It tries to understand the cognitive processes that make it possible *to communicate the thoughts and knowledge* via language. The recent research on associative, semantic and thematic *priming* effects [10] shows evidence that language lexicalization plays irreplaceable rôle in the *conceptual organization of knowledge*.

Computational linguists design algorithms to verify or deny theoretical linguists' theories usually by modelling the language usage from their surface form. They often use big corpora to build a language model based on statistics computed from texts by zillions of writers. The natural language representation is based on averaging of word usage. The *representation of knowledge of a language* is stored in the form of big corpora containing billions of words [16].

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2013, pp. 87–91, 2013.

Computer scientists are trying to understand the computation by designing appropriate data structures that allow appropriate representation of the problem in hand, so that algorithms are easily formulated. They have recently came up with a new definition of computation as any *process generating knowledge* [14]. It fits the view of natural language understanding as a computational process, during which word meanings and semantics gets computed on the fly during discourse, and the representation might be affected by such computation.

All research communities above strive to name, generate and compute knowledge of natural language understanding to get an insight on how to model natural language communication. The common sense is that the key to successful natural language processing is appropriate *natural language representation* (NLR).

You shall know a word by the company it keeps. (John Rupert Firth)

2 Development of Natural Language Representations

Chomskyan linguistic nativism [3] stressed the generative, formal qualities of language, ignoring the fact that people communicate successfully even using syntactically wrong discourse. Similarly, on a semantic level, the twentieth century prevalent view was that a word does objectively have several distinct *meanings* that could be enumerated as in a dictionary entry. The claim was that by solving the task of word sense disambiguation we will be close to natural language understanding. Just another example of another *discrete representation* in language modelling is represented by the view that some powerful logic will be sufficient to effectively represent discourse semantics.

Corpora linguists collected the evidence that language use is very variational and diverse, not fitting the boundaries of syntactic, grammatical, semantic structures and logical formalisms. Language is on move, with many irregularities that develop in time and space. By studying *word sketches* [12] we see that word meanings are subjective, hard to separate, and form collocates depending on context. Since the end of last millenium, there are linguists that do not believe in clear separation of word senses [11].

To anchor a word in a context, the theory of *lexical priming* has been coined by Hoey [8]. Backed up by evidence from psycholinguistics, he articulates and argues for a new theory where each occurrence of lexical item enforces 'priming' of it given a co-*locational* context. A Firth's 'word's company' is viewed broadly, as pervasive and subversive types of collocations on a sentence and higher levels. The word, or more precisely a lexical item, is learnt through encounters. Each "new encounter either reinforces the priming or loosens it," and make "drifts in the priming" [8].

Lexical priming theory is convincing in many aspects, especially that it allow the explanation of how different word meanings may come up based on previous word usage, and the whole context of lexical occurrence including the pragmatics. The contextual clues additively contribute to the on the fly

computation of every word meaning. This is in sync with all the WSD research to date. Natural question arises: how to implement the *computational lexical priming* for use in NLP tasks, and what representations should be used? Could word sense disambiguation problem be solved by lexical priming computed over appropriate data structures mimicking the way of processing we collect evidence from psycholinguistic research and novel view of computation?

> We should stop acting as if our goal is to author extremely elegant theories, and instead embrace complexity and make use of the best ally we have: the unreasonable effectiveness of data. (Peter Norvig *et al.*, 2009 [5])

3 Discrete and Continuous Language Representations

Most mainstream "scholastic" language representations used to date are *discrete* representations as lists, graphs or logics, and aims to capture a language as an objective, discrete, fossil structure. It does work to some extent for modelling of *conscious*, deductive reasoning, but leads to a very limited functionality and use cases.

In the real world, language nuances of every communication side are different, and it takes time before a word's usage and meaning will settle, converge into an entry in dictionary and will be understood during the man to man discourse. Workflow of language processing is usually layered into separate modules of morphology, syntax, and semantics, forgetting the primary communication goal of discourse via natural language. Syntax encodes information structure [2], only helps to resolve the main task of meaning disambiguation of a message.

One should be warned by adopting easy simplifications. "A linear ordering of a multi-parameter universe is usually nonsense" [15] does hold not only in an science impact measurements, but in the word meaning, or generally, in language modelling, too. A more complex representation is necessary.

> . . . if nature is really structured with a mathematical language and mathematics invented by man can manage to understand it, this demonstrates something extraordinary. The objective structure of the universe and the intellectual structure of the human being coincide. (Pope Benedict XVI [1])

4 The Unreasonable Effectiveness of Language Representations Computed from Corpora

Let us suppose that lexical items (single word, lemma or longer term) will be represented as a node in a neural network. Let synapses represent co-locative relations of different kinds, including perceptual clues from visual subsystem, trains of thoughts, coherence links between sentences etc.

Mutatis mutandis, methods like Hebbian learning [7], WebSOM [13] and random walking in graphs (explicit collocation representations) may be used in the computations of continuous representations of natural language.

There are well established methods of building a language models by various types of *smoothing*. Failure of 'semantic web' approaches to unify (discrete) keyword-based and ontology based semantics cause shifting towards (continuous) distributional semantics approaches.

There is an evidence of *conscious and unconscious* semantic priming [4]. Corresponding *discrete and continuous* data structures might help to proper modelling of personal mental lexicon. We are developing appropriate discretization algorithms specific for natural language tasks, based on random walking [9] in huge corpora towards this goal.

Get comfortable with paradoxes. (David Allen)

5 Conclusion

We have expressed our view of continuous and personal language representation motivated by Hoey's lexical priming theory. We argue that it will allow modelling of several language phenomena with ease. It is yet to be confirmed by computational experiments and computed representations appropriate for specific NLP tasks in natural language understanding.

Acknowledgements This work has been partially supported by the European Union through its Competitiveness and Innovation Programme (Information and Communications Technologies Policy Support Programme, "Open access to scientific information", Grant Agreement No. 250,503).

References

1. Message of His Holiness Benedict XVI to Archbishop Rino Fisichella, Rector Magnificent of the Pontifical Lateran University, on the Occasion of the International Conference "From Galileo's Telescope to Evolutionary Cosmology. Science, Philosophy and Theology in Dialogue" (Nov 2009)
2. Brown, M., Savova, V., Gibson, E.: Syntax encodes information structure: Evidence from on-line reading comprehension. Journal of Memory and Language 66(1), 194–209 (2012), http://dx.doi.org/10.1016/j.jml.2011.08.006
3. Chomsky, N.: Syntactic Structures. Walter de Gruyter (2002)
4. Dehaene, S., Naccache, L.: Imaging unconscious semantic priming. Nature 395(6702), 597 (1998)
5. Halevy, A., Norvig, P., Pereira, F.: The unreasonable effectiveness of data. IEEE Intelligent Systems 24(2), 8–12 (Mar 2009), http://dx.doi.org/10.1109/MIS.2009.36
6. Hanks, P.: Do word meanings exist? Computers and the Humanities 34, 205–215 (Apr 2000)
7. Hebb, D.: The Organization of Behavior. Wiley, New York, second edn. (1968)
8. Hoey, M.: Lexical Priming: A New Theory of Words and Language. Routledge (2012)
9. Hughes, T., Ramage, D.: Lexical semantic relatedness with random graph walks. In: Proceedings of EMNLP-CoNLLi 2007. pp. 581–589 (2007)

10. Jones, L.L., Estes, Z.: Lexical priming. Visual Word Recognition Volume 2: Meaning and Context, Individuals and Development 2, 44 (2012)
11. Kilgarriff, A.: I don't believe in Word Senses. Computers and the Humanities 31(2), 91–113 (1997)
12. Kilgarriff, A., Tugwell, D.: WORD SKETCH: Extraction and Display of Significant Collocations for Lexicography. In: Proceedings of the Workshop 'Collocation: Computational Extraction, Analysis and Exploitation' ACL, Toulouse, France. University of Brighton (2001), http://www.itri.bton.ac.uk/~David.Tugwell/colloc.ps
13. Lagus, K., Kaski, S., Kohonen, T.: Mining massive document collections by the websom method. Inf. Sci. 163(1–3), 135–156 (Jun 2004), http://dx.doi.org/10.1016/j.ins.2003.03.017
14. van Leeuwen, J., Wiedermann, J.: Computation as an unbounded process. Theoretical Computer Science 429, 202–212 (2012), http://dx.doi.org/10.1016/j.tcs.2011.12.040
15. Maurer, H.: A linear ordering of a multi-parameter universe is usually nonsense. Theoretical Computer Science 429, 222–226 (2012)
16. Pomikálek, J., Jakubíček, M., Rychlý, P.: Building a 70 billion word corpus of English from ClueWeb. In: Calzolari, N., Choukri, K., Declerck, T., Dogan, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S. (eds.) Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC '12). European Language Resources Association (ELRA), Istanbul, Turkey (May 2012)

Type-based Search of Idiomatic Expression

Jan Bušta

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
busta@fi.muni.cz
http://nlp.fi.muni.cz/

Abstract. This paper presents evaluation of different approaches to extract verb-noun idiomatic expressions in Czech. These approaches are based on the structure of the idiom and its behavior in language. PMI and syntactic and lexical fixedness modified using VerbaLex and generated thesaurus provide useful tool for choosing best idiomatic candidates for manual annotation and evaluation. Moreover we focused on general adapting the algorithms for Czech.

Key words: idioms, idiomatic candidates, syntactic fixedness, lexical fixedness, transitive verbs, thesaurus

1 Introduction

In any language there are always multi word expressions which are about to breake the Frege's Principle of compositionality. Let's call them idioms. But, it is not so easy to determine, if the compound word expression breaks this principle or not.

The need of determining such language segments is obvious: Every time we deal with machine translation from one language to another, we deal with this non word-by-word translation. The words of idioms cannot be translated by this basic approach, we have to mark them and handle separately. Lexisting lexicons of idiomatic phrases are always old and does not allow to search the today's language. Computer processing helps to build this kind of lexicons very fastly with the minimum amount of time needed by the lexicographers.

There are many approaches, how to define the idiom better, but there will be always the problem with the decision which is individual for every language user (or language user group). We based this work on the weakest presumption of Principle of compositionality and provide the language user more reliable data sources to determine the border line, therefore any time we speak about idiom, we just mean idiomatic candidates.

In further chapters we will present the basics of approaches to automatic idiom search, modified algorithms, the differences to approaches to English and evaluation of the methods.

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2013, pp. 93–96, 2013.

2 Type-based: Verb-noun

In this article we present the approaches to type-based idioms. It defines our working area as the idiomatic phrases which consist from transitive verb and noun in accusative case (the transitive verb requires direct subject in accusative case). We can be sure, there is an object (in accusative case) for all transitive verbs we will handle.

Let's define the idiomatic phrase as a tuple $< v, n >$, where v is a transitive verb and n is the object.

3 The base: Fixedness

Talking about the Frege's principle we assume, it could represented by fixedness. The more fixed the phrase is, the more idiomatic behaviour we expect, so we are convinced that the idiomacity and the language fixedness are related and correlating.

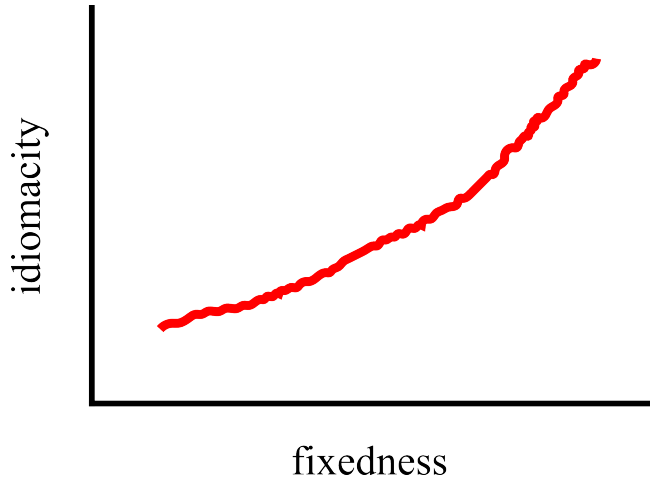


Fig. 1: Fixedness and idiomacity correlation.

The fixedness is much easier to implement as an algorithm for computer processing than the Frege's principle itself. There are multiple metrics available to show the fixedness of the phrase.

First we build $\mathcal{V}$, the set of transitive verbs acquired from *VerbaLex*. For all these verb we search the corpus for concordances in the form of $< verb, noun >$ or $< verb, intersegment, noun >$ with preserving the accusative case restriction. The intersegment can be adjectives or pronouns.

So we have all the candidates phrases and further algorithms show us the fixedness of each pair found in corpus.

We construct distributional thesaurus based on the word sketches for every noun n, let the set be $\mathcal{T}_n$, and let the $f(v, n_j)$ be frequency, where v is the transitive verb of candidate pair $< v, n >$ and n_j is element of $\mathcal{T}_n$. Let $f(v, *) = \sum_{n_j \in \mathcal{T}_n} f(v, n_j)$ and $f(*, n_j) = \sum_{v \in \mathcal{V}_n} f(v, n_j)$. $\mathcal{N}$ is set of all nouns found in corpus and being object of any transitive verb.

One of the metrics we used is PMI, which is defined for all pairs as:

$$PMI(v, n_j) = log \frac{|\mathcal{V} \times \mathcal{N}| f(v, n_j)}{f(v, *) f(*, n_j)}$$

The lexical fixedness is based on PMI and defined as

$$Fix_{lex}(v, n) = \frac{PMI(v, n) - \overline{PMI}}{s}$$

The $\overline{PMI}$ is the mean of all $PMI(v, n_j)$ and s is the standard deviation. This approach is motivated by the fact, that idiomatic phrases consist of such a word collocation which is significantly different from the others. Fazly and Stevenson define the set $\mathcal{T}_n$ not as thesaurus, but as a set of synonyms to the noun n. We assume, that if the noun from idiom can be replaced by the noun from thesaurus, the fixedness is lower than by using the set of synonyms.

The last algorithm we present is measuring the syntactic fixedness. Fazly and Stevenson approach is based on syntactic changes of the phrase: passivation, pluralisation and type of article, but those features are not feasible for Czech. Handling the passivation in Czech was not selected because the corpus searching queries were not prepared for this purpose. The free word order makes this task difficult if one wants to be precise. The type of article is not significant even, the Czech language do not use it. The pluralisation is the only one of the syntactic fixedness computation parameters which could be used in Czech, but according to the fact, that this is the only one, another approach is presented: To measure the syntactic fixedness we observed the changes in intersegment length. Let the $f_i(v, n) = f(v, n)$ where i is the length of intersegment. The syntactic fixedness of the pair as follows:

$$Fix_{syn}(v, n) = \frac{max(f_0(v, n), f_1(v, n), f_2(v, n))}{\sum_{i=0}^{2} f_i(v, n)}$$

This shows us that the phrase is more fixed (idiomatic) if most of the occurrences are in just one form. The length of the intersegment is fixed and language does not allow the flexibility by changing the internal structure of the idiom.

4 Conclusions

All three approaches produce lot of candidate phrases, but it is much easier for human evaluation to read the candidate phrases instead of searching the whole text for the idioms.

Modifying the algorithms for Czech was beneficial so that we can use it also for other Slavic languages and generate the candidate dictionary.

The evaluation showed that with comparing to the existing lexicons of idiomatic phrases, there were the verb-noun pairs on the top which were not in the lexicon (this has been marked by annotators).

Distinguishing of idiomatic phrases is a complex task and our work shows that we can easily modify the current approaches and make them better for other languages.

Acknowledgements This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin project LM2010013.

References

1. Bannard, Colin. *A measure of syntactic flexibility for automatically identifying multiword expressions in corpora* In *Proceedings of the Workshop on a Broader Perspective on Multiword Expressions*. Association for Computational Linguistics, 2007.
2. Bušta, Jan. *Výpočet četnosti výskytů hesel SČFI v korpusu*. Bakalářská práce, Fakulta informatiky, Masarykova univerzita, Brno, 2009.
3. Čermák, František. *Frazeologie a idiomatika česká a obecná*. Praha: Nakladatelství Karolinum, 2007.
4. Čermák, F., Hronek, J. – editors. *Slovník české frazeologie a idiomatiky*. Praha: Academia, 1994.
5. Čermák, F. a kol. *Slovník české frazeologie a idiomatiky 3, Výrazy slovesné*. Praha: Leda, 2009.
6. Fazly, A. a Stevenson, S. *Automatically constructing a lexicon of verb phrase idiomatic combinations*. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*. 2006.
7. Fellbaum, Christiane. *The determiner in English idioms*. In *Idioms: Processing, structure, and interpretation*. Hillsdale: Lawrence Erlbaum Assiciates, 1993.
8. Hlaváčková, D. a Horák, A. *VerbaLex – New Comprehensive Lexicon of Verb Valencies for Czech*. In *Proceedings of the Computer Treatment of Slavic and East European Languages 2005*. Bratislava, 2005.
9. Horák, A., Rychlý, P., Kilgarriff, A. *Czech word sketch relations with full syntax parser*. In *After Half a Century of Slavonic Natural Language Processing*. Brno: Masaryk University, 2009.
10. Karlík, P., Nekula, M., Rusínová, Z. – editors. *Příruční mluvnice češtiny*. Praha: Nakladatelství Lidové noviny, 2008.
11. Kilgarriff, A., Rychly, P., Smrz, P. a Tugwell, D. *The Sketch Engine*. In *EURALEX Proceedings 2004*. Lorient, 2004.
12. Lin, D. *Automatic identification of non-compositional phrases*. In *ACL '99 Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Stroudsburg: Association for Computational Linguistics, 1999.
13. Rychlý, Pavel. *Korpusové manažery a jejich efektivní implementace*. Doktorská práce, Fakulta informatiky, Masarykova univerzita, Brno, 2000.

Author Index

RASLAN 2013

Seventh Workshop on Recent Advances in Slavonic Natural Language Processing

Editors: Aleš Horák, Pavel Rychlý
Typesetting: Adam Rambousek
Cover design: Petr Sojka

Printed and published by Tribun EU s. r. o.
Cejl 32, 602 00 Brno, Czech Republic

First edition at Tribun EU
Brno 2013

ISBN 978-80-263-0520-0

www.librix.eu