RASLAN 2018
Recent Advances in Slavonic Natural Language Processing

A. Horák, P. Rychlý, A. Rambousek (Eds.)

RASLAN 2018

Recent Advances in Slavonic Natural Language Processing

Twelfth Workshop on Recent Advances in Slavonic Natural Language Processing, RASLAN 2018
Karlova Studánka, Czech Republic, December 7–9, 2018
Proceedings

Tribun EU
2018

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

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

ISBN 978-80-263-1517-9
ISSN 2336-4289

Preface

This volume contains the Proceedings of the Twelfth Workshop on Recent Advances in Slavonic Natural Language Processing (RASLAN 2018) held on December 7th–9th 2018 in Karlova Studánka, Sporthotel Kurzovní, Jeseníky, 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 discussion 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.

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 parsing
* 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 2018 offers a rich program of presentations, short talks, technical papers and mainly discussions. A total of 15 papers were accepted, contributed altogether by 23 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 Marie Stará. 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 a printer of these proceedings is gratefully acknowledged.

Brno, December 2018 Karel Pala

Table of Contents

IV Text Corpora

Part I

Morphology and Syntax

Towards the New Czech Grammar-checker

Vojtěch Mrkývka

Faculty of Arts, Masaryk University
Arne Nováka 1, 602 00 Brno, Czech Republic
mrkyvka@phil.muni.cz

Abstract. I created a basis for the new grammar-checker of Czech. This was positively accepted by the committee and I was allowed to continue its development in my further study. In this paper, I want to describe the proximate issues of its active development.

Keywords: spell checker; grammar checker; text analysis; correction

1 Introduction

In September 2018 I've published the first version of the new grammar-checker of Czech (see Figure 1). I was motivated by the fact that the presumably best current option is part of the proprietary software. Additionally, it doesn't provide satisfactory results. Limitations of it can be discovered by comparing functionality described in the article *Kontrola české gramatiky (český grammar checker)* by Vladimír Petkevič and the status quo provided by the most recent versions of the same software, where some of the described features are no longer available. [1]

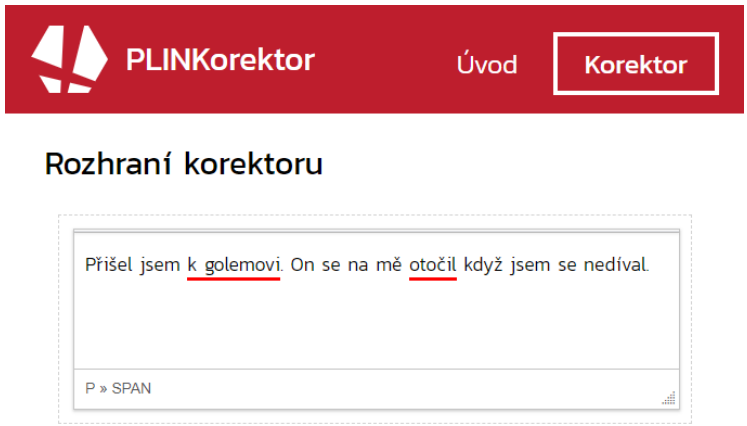


Fig. 1: Current version of the new grammar-checker. The red underlines depict errors.

Aleš Horák, Pavel Rychlý, Adam Rambousek (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2018, pp. 3–8, 2018.

2 The Current Version

2.1 The Corrector Interface

For the corrector to be truly open to use, I developed an online application using widely spread text processor TinyMCE v4 and provided different correction mechanisms as separate modules. [2] Due to its JavaScript nature, I decided to make it asynchronous as faster modules wouldn't need to wait for the slower ones. Obviously, in some cases, there were necessary dependencies. It was likely that tasks such as tokenisation or lemmatisation would be required by multiple correction modules, so it was more than convenient to perform these tasks separately (see Figure 2). The approach could be seen on the final render, where corrections made by faster modules are also displayed sooner in the text processor window. To push the speed even further, I separated the processing to individual paragraphs, where the process of grammar-checking is repeated only on one(s), which were modified.

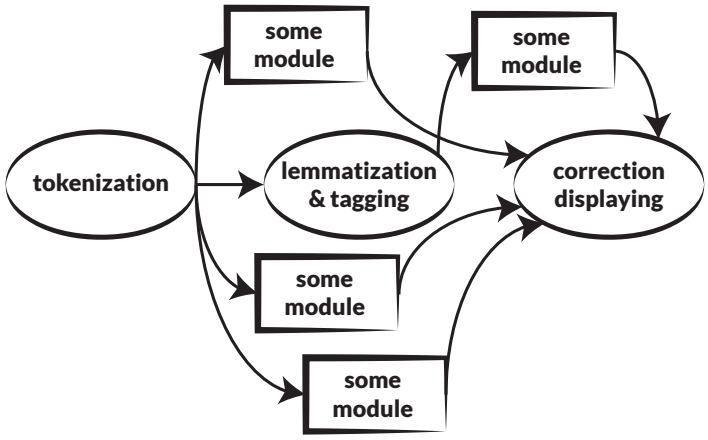


Fig. 2: Visualisation of the new corrector's asynchronous nature.

2.2 Implemented modules

Although the current version does not achieve qualities of its competitor, as a relatively open software it does have a potential to do so by including a multitude of correctly working submodules. At the time of writing this article, there were six focused on four different types of mistakes – spelling, syntactic, morphological and typographical errors. These modules stood fairly well in testing, where I've provided text with artificial mistakes (see table 1; for further information see the thesis [3]). The modules vary in success, but that is mainly because the less successful modules deal with more complex issues.

Table 1: Number of true and false positive/negative values (TP/FP/TN/FN) for the individual modules and corresponding precision (pre) and recall (rec) rates.

Correction	TP	FP	TN	FN	pre	rec
Misspellings (excl. proper nouns)	24	0	487	16	1,000	0,600
Misspellings (incl. proper nouns)	7	17	497	6	0,292	0,538
Vocalisation of prepositions	4	0	8	0	1,000	1,000
Multiple whitespaces	4	0	515	0	1,000	1,000
Whitespace in the interpunction proximity	7	0	119	0	1,000	1,000
Conditionals	2	0	1	0	1,000	1,000
Commas in a sentence	3	0	0	4	1,000	0,429

3 The proximate issues

As I suggested before, the corrector is far from being perfect. There are many issues, which have different difficulty as well as different priority. In the long run, the development will consist of adding new modules, which will improve corrector as a whole. In the short run, there are tasks, which should improve further development and user acceptance alike.

3.1 Genuine testing

The quality evaluation of modules' success is crucial part of the development. The approach I used, however, is far from ideal. The problem is twofold. The text I used was artificial, eg. it didn't reflect real distribution of users' mistakes, and its length was insufficient. Because of this, the results could be hardly accepted as generally valid. The solution lies in the existence of a collection of genuine texts with correctly labelled errors. On Masaryk University corpus *Chyby* was created, containing various annotated types of errors in texts made by students. [4] Unfortunately, access to this corpus is limited, so I cannot assess its suitability and I did not find any other evidence of the Czech corpus of such quality.

Additionally, the current version of the corrector lacks any kind of interface, which would allow evaluating results automatically. This is necessary as the manual inspection would cost excessive amount of time, which could be used for the further development.

3.2 Error reporting

Next issue covers incorrectly found (non-)errors and users' feedback. Though it isn't implemented in the current version, the reporting itself should be relatively easy. The problem unfolds with keeping these cases hidden even after the minor change of text. Although the current approach is based on tokens, it is not bound to token's content, as there could be a mismatch, but to its position (token

number, see Fig. 3). One option is to extend the binding to different criteria, but due to the various nature of different modules, there would be problem to find any general approach and this issue will have to be addressed by the different correction modules separately. In some cases, I believe the solution will be fairly easy, such as building ignore list for the spell-checker, but for other, it could provide a significant challenge.

Fig. 3: Example of correction, word *runing* would be highlighted and user would be suggested with the correct form.

3.3 Spell-checking

Spell-checking is a distinct chapter of the corrector and such as it has specific issues. They are often described as *non-word error detection*, *isolated-word error correction*, and *context-dependent error correction*. [5] The context-dependent error correction is an advanced task, so in this phase of development, I do not plan to focus on it. The other two, however, are very important as they would be among the first things required by the end users. The non-word error detection aims solely on distinguishing words from non-words. Its precision is closely linked to the quality and size of the lexicon used. Although morphological analyser Majka, whose lexicon is used in the current version of the grammar-checker, provides a fairly big number of word forms, there are contained substandard word forms beside the standard ones. If it would be used in the future, there is the necessity of filtering these forms. Additionally, even though Majka as an analyser, unlike its predecessor, is not dependent on its lexicon, the standard lexicon is not often updated. [6] This raises the question of whether to use it in the future or if would be better to switch to different, expandable resource such as *hunspell*. [7] Either way, there should be created service (if it doesn't already exist with given system), such as a programme or internet application, which would allow the moderator to easily add new words, as they can come in large numbers, which should be then automatically included in the right format.

The related issue is an isolated-word correction. This topic covers automatic corrections as well as suggestions. Unlike the competition, this is not yet covered by the corrector in any way. Some tools, such as previously mentioned hunspell, have their own methods implemented. Possible custom implementation could use algorithms based on *Damerau-Levenshtein distance* measures or other noisy channel approaches such as *Brill and Moore's model*. [8]

4 Conclusion

This paper summarises the development history of the new Czech grammar-checker and uncovers the proximate issues in future development. These issues aren't the only ones to be solved before the corrector could be considered as satisfactory. The success of the final product rests on the success rate of the added modules. There are multiple works on partial topics, which are in some stage of development or relatively freshly finished. This can provide me with quality resources for future module development. Apart from correction itself, the interface has to provide a sufficient range of metatext options, such as headers, bold and italic text, cut and others for users to start using it on daily basis. This is already implemented in the TinyMCE editor, but momentarily disabled.

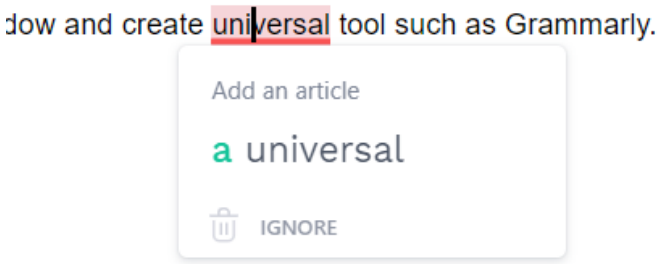


Fig. 4: Correction information in Grammarly.

Although one of the steps is to enable users these options in the editor's window, the ultimate goal is to make the corrector independent. The model for this can be seen in the American *Grammarly* [9], which provides grammar checking of English text on the internet (see Figure 4). As it provides quality and understandable results for its language domain I hope I will be able to get at least close to its successes from the Czech language point of view.

Acknowledgements. This work was supported by the project of specific research *Čeština v jednotě synchronie a diachronie* (Czech language in unity of synchrony and diachrony; project no. MUNI/A/0862/2017).

References

1. Petkevič, V.: Kontrola české gramatiky (český grammar checker). Studie z aplikované lingvistiky-Studies in Applied Linguistics [online] **5**(2) (2014 [2018-10-31]) 48–66
2. TinyMCE: Create a plugin for tinymce [online] (2018 [2018-10-31])
3. Mrkývka, V.: Webové rozhraní pro automatický jazykový korektor češtiny [online]. Diplomová práce, Masarykova univerzita, Filozofická fakulta, Brno (2018 [2018-10-31])

4. Pala, K., Rychlý, P., Smrž, P.: Text corpus with errors. In Matoušek, V., Mautner, P., eds.: Text, Speech and Dialogue [online], Berlin, Heidelberg, Springer Berlin Heidelberg (2003 [2018-10-30]) 90–97
5. Kukich, K.: Techniques for automatically correcting words in text. ACM Comput. Surv. [online] **24**(4) (December 1992 [2018-10-31]) 377–439
6. Šmerk, P., Rychlý, P.: Majka – rychlý morfologický analyzátor [online]. Technical report, Masarykova univerzita (2009 [2018-10-31])
7. Németh, L.: Hunspell: About [online] (2003 [2018-10-31])
8. Brill, E., Moore, R.C.: An improved error model for noisy channel spelling correction. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics [online]. ACL '00, Stroudsburg, PA, USA, Association for Computational Linguistics (2000 [2018-10-31]) 286–293
9. Grammarly: About | grammarly [online] (2018 [2018-10-31])

Using Syntax Analyser SET as a Grammar Checker for Czech

Marie Novotná, Markéta Masopustová

Faculty of Arts, Masaryk University
Arne Nováka 1, 602 00 Brno, Czech Republic
{428801,415295}@mail.muni.cz

Abstract. Checking the grammaticality of written text is one of the essential and highly desirable tasks of natural language processing. One of the very common mistakes in Czech texts are errors in agreement or using colloquial expressions in written texts. Based on the analysis, we created new rules for the grammar of the SET syntax analyser to use it not only as an analyser but also as a grammar checker. Then we tested their functionality. The side effect of the work was also the identification of possible complications, deficiencies of the tools and partly also suggestions for their solution.

Keywords: syntactic analysis; SET; spell checker; grammar checker; grammatical agreement; subject-predicate agreement; compound subjects; attributive adjective-noun agreement; colloquial expressions

1 Introduction

One of the basic tasks of natural language processing is checking the grammaticality of texts. Grammar checkers check the formal and grammatical accuracy of text written in a natural language based on the rules of the language in question. Since the complexity of spelling, grammatical and stylistic features varies in different languages, the level of grammar checkers differs. While the spell check is already relatively well solved, the problem is more complicated at other levels of language.

For our work, we have chosen areas of language in which users of language often fail. One of them is grammatical agreement, which is often written ungrammatically, since it is not noticeable in standard spoken Czech (i.e. the difference in writing *i/y*), or it uses different endings in colloquial form, which are informal for written texts. Since the subject-predicate agreement has already been partly solved earlier (see Chapter 2), we focused only on sentences in which the subject was multiple (consisting of two names), and the attributive adjective-noun agreement when attribute stands before the name. We also worked on selected common mistakes in the area of word formation, stylistics and syntax. These are often found in the commonly spoken language, however, in a written language, they are considered as faulty constructions. Based on this analysis, we created rules for the grammar of the syntax analyser SET [1], which makes it possible to identify the mistakes in the texts.

Aleš Horák, Pavel Rychlý, Adam Rambousek (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2018, pp. 9–14, 2018.

2 Related Work

For the Czech language, there are several different grammar checkers. The problem of spelling is solved well, whether in stand-alone programs or web applications or as part of text editors (e.g. Microsoft Office). In case of grammar checkers, only two commercial products are known, namely Grammaticon from Lingea [2] and Kontrola české gramatiky, developed at Czech Language Institute of the Czech Academy of Sciences as part of the Microsoft Word editor [3]. There is not much information about these grammar checkers, mostly only advertising posts. Nevertheless, Grammaticon is no longer supported today (the support ended in 2014), and Kontrola českého gramatiky has, according to the author's words, limited functionality since its launch. The only known stylistic checker for Czech was part of Grammaticon and had only minimal functionality [4].

As mentioned in the previous chapter, attempt to use the SET analyser as a grammar checker already exists [5]. There was a simple rule for detecting an error in a subject-predicate agreement. This subject was referred to as subject-bad. The rule was added to the existing one, and then the new grammar was tested on 26 sentences with 11 mistakes in subject-predicate agreement, which came from the tests of pupils of the first grade of elementary schools and were manually identified and classified [6]. This rule, however, has been able to work only with a simple subject, so in our work, we have continued with the rules for a multiple subject. Also, we extended the coverage of the grammar checker to include other mistakes, which we will introduce in Chapters 4-6.

3 The SET parser

The Syntactic Engineering Tool, introduced in [1], is based on partial segmentation of sentence. It works with implemented grammar, which is made out of rules for searching for the relations between tokens, or sentence members. These rules are then compared with the input sentence, and all relevant records are counted, their weight is evaluated, and the "heaviest" rules are applied. The result of such an analysis is the sentence with the labelled parts of speech and the relationships between them. Examples of rules are given in [1].

Our task was to create rules to correctly identify the part of speech in which the error lies and to mark it appropriately.

4 Attributive adjective-noun agreement

In the case of the attributive adjective-noun agreement, we limited the experiments to a simple adjective attribute standing before the noun. A new label has been introduced to indicate the wrong attribute *modifier-bad*. Primarily, we have searched for attributes that are widely used in spoken or informal language formations but do not belong to the written text.

In some cases, finding the error attribute was a simple task, because the information about the colloquial expressions was already mentioned in the

morphological tag on which the syntactic analysis is based (e.g. *malýmu klukovi*). In many cases, it was the format of an adjective that is formal for another case than that it was used (e.g. *o obědový pauze*). These mistakes were due to a case divergence. A common mistake in Czech is also in the use of the dual ending of the attribute if it is plural (non-zero) number (e.g. *barevnýma pastelkama*). However, due to the current limitations of the tools, we have not been able to solve this problem. In total, four new rules have been created to detect an error in the attributive adjective-noun agreement.

Following example shows one of the rules. This rule marks adjective as modifier-bad if the adjective was marked as colloquial in the morphologic analysis.

```
TMPL:   $MALYHOMU $...* noun MARK 0DEP 2 PROB 6001
        LABEL modifier-bad

$MALYHOMU(tag): k2.*wH
```

The rules were tested with 230 sentences with the attributive adjective-noun agreement selected from the Skript2012 [7] corpora, either in the correct or wrong form, approximately in a ratio of 1:1. It was correctly marked as *modifier-bad* 92 of the 136 wrong sentences, and a false positive appeared only once. The results are shown in Table 1, further comments can be found in [8].

In an analysis of the results, we found that a relatively large part of the attributes in which the error was not revealed was pronouns that we did not focus on (e.g. type *v kterým příkladu*), so the actual coverage inadequate cases could have been more successful.

5 Multiple subject-predicate agreements

Since the subject-predicate agreement has been dealt with earlier [5], we have focused on the multiple subjects, regarding the limitations of the tool used to the subject consisting of two nouns. SET allows to create coordination, but failed to allocate its morphological tag. In the syntactic analysis, it usually found a coordination rule, and each component of the multiple subjects was joined to it, but it evaluated the subject-predicate agreement for each name separately and subsequently assigned the coordination a label which had the highest weight according to the rules. The simplest solution to this problem would be to implement the SET rules for assigning a morphological tag to all coordination

Table 1: Results of attributive adjective-noun agreement.

TP	FN	FP	recall	precision
92	44	1	0,68	0,99

(for example, if there is at least one member of the coordination family, consider coordination as the gender of male animated).

Because SET does not allow coordination for the above mentioned morphological tag assignment, we had to deal with the problem differently. We have created entirely new rules for creating coordination directly linked to the predicate. New rules increased the number of different combinations, so it was necessary to create a relatively large number of rules for different gender and numbers of subjects and predicate. Given the complexity of this problem, the rules were set relatively "roughly". In the future, however, we expect the analyser to be adjusted, and when the coordination can be managed more efficiently, so we expect to change our rules in order to work more reliably. As a result of this part of the work, there were 24 rules for identifying the error in subject-predicate agreement and 35 rules to indicate the correctness of such compliance.

Following example shows one of the rules which are made for detecting multiple subject-predicate agreements. In this agreement on the first position is noun masculine animated and on the second position is any noun in nominative followed by predicate with ending *-i* (masculine animated in plural). If this pattern is found, multiple subject-predicate is marked as <*cood-s*> during syntactic analysis.

```
TMPL:   $SUBJ_M $...* $AND $...* $SUBJ $...* $PREDi MARK 0 2 4
        HEAD 2 DEP 6 PROB 10000

$PREDi(tag): k5.*gM.*nP
$SUBJ(tag): k1.*c1 $SUBJ_M(tag): k1.*gM.*c1
$AND(word): a i nebo ani $...*(tag not): k5 k8
```

We tested these rules on selected sentences of the Skript2012 [7] and CzeSL-SGT [9] corpora, which contained multiple subjects. Of the 39 errors, 24 were revealed, a false positive appeared 18 times in 131 correct sentences. The results are shown in Table 2, further comments can be found in [8].

When we tried analysis on the random sentences from the Internet, the results were even worse, but the exact numbers are unknown. The grammar checker's unreliability in this regard was mainly due to erroneous morphological disambiguation, but also to other factors (for example, we have failed to limit the rules to verbs in past tense that work on most of the mistakes in agreement). We continue to work on these issues to make the rules applicable in practice.

Table 2: Results of multiple subject-predicate agreements.

TP	FN	FP	recall	precision
24	15	18	0,62	0,57

6 Colloquial expressions in the written texts

Within the next module, we focused on the occurrence of spoken language elements in the written texts, because colloquial expressions are formal only in its spoken version. The first problem was the word order, namely the order of enclitics and prepositions. In Czech, the sentences are beginning with the enclitics, e.g. **Si pořídím nové kolo.*, rated as informal, whereas the sentences with two prepositions behind each other are considered as confusing for the participant of the speech, because the bond is broken, e.g. *Dospěl k pro něj těžké otázce*. We also tried to solve the excess of the demonstrative pronouns in the sentence and repeating the same expressions within one sentence. Also pleonasm, i.e. redundant repetition of the same (for example, *dárek zadarmo*), absurd superlatives (e.g. *nejzákladnější*), wrong use of the word *jakýkoli*, forgetting a double conjunctions (e.g. *buď-(a)nebo*), wrong usage of pronoun *který/jenž* and finally, the occurrence of words or forms of words rated as spoken lexicons in written form. The last subcategory were so-called other mistakes where we included mistakes such as the bad writing of words *výjimka* and *permanentka*, addressing another case than the fifth, incorrect form of the word *datum*, hypercorrection in the nominative of life masculine pattern muž (e.g. **reprezentanté*) and misuse of conjunction *mimo*.

The result is an extensive set of rules. For our evaluation, we built corpus of 1200 sentences without error and 400 sentences with an error. Generally speaking, the simple rules have had great success, and the more complex rules were worse, which is the result that we expected. The results are shown in Table 3 and discussed in more detail in [10].

Table 3: Results of colloquial expressions in the written texts.

rule	TP	FN	FP	precision	recall
enclitics correct sentences	136	0	0	1	1
enclitics bad senteces	41	0	309	0,117	1
prepositions	35	3	5	0,875	0,921
demonstrative nouns	59	1	1	0,983	0,983
repetition of words	46	0	0	1	1
pleonasms	131	17	0	1	0,885
superlatives	11	0	0	1	1
pronoun *jakýkoli*	20	0	0	1	1
double conjunctions	54	7	19	0,740	0,885
gender *který* correct sentences	151	0	1	0,993	1
gender *který* bad sentences	37	3	133	0,218	0,925
colloquial expressions	16	1	8	0,667	0,941
other	117	0	0	1	1

7 Conclusion

Our project aimed to create new rules for the SET grammar checker, which extends its functionality not only as a syntactic analyser but also as a Czech grammar checker. We have tried to partially cover the area of the attributive adjective-noun agreement, multiple subject-predicate agreements and other selected language issues. We are aware that the range of errors covered by our work has been considerably limited and that it needs to be expanded even more for the needs of the grammar checker. Similarly, it is possible to work on improving the precision and recall.

Creating a grammar checker for such a grammatically demanding language as the Czech language is not an easy task. However, we are convinced that if enough attention is paid to the problem and the tools are continually improved; we can make it to the ideal result slowly and in small steps.

Acknowledgements. This work has been partly supported by the project of specific research *Čeština v jednotě synchronie a diachronie* (Czech language in unity of synchrony and diachrony; project no. MUNI/A/0862/2017).

References

1. Kovář, V., Horák, A., Jakubíček, M. In: Syntactic Analysis Using Finite Patterns: A New Parsing System for Czech. Springer (2011) 161–171
2. Behún, D.: Lingea grammaticon – přísný strážce jazyka českého
3. Petkevič, V. In: Kontrola české gramatiky: český grammar checker. Univerzita Karlova v Praze, Filozofická fakulta (2014) 48–86
4. Pala, K.: Stylistický korektor (2017)
5. Kovář, V. In: Partial Grammar Checking for Czech Using the SET Parser. Springer (2014) 308–314
6. Trifanová, B.: Analýza chyb v diktátech žáků po absolvování 1. stupně ZŠ [online] (2013 [cit. 2018-10-30])
7. Šebesta, K.: Skript2012: akviziční korpus psané češtiny – přepisy písemných prací žáků základních a středních škol v ČR (2013)
8. Novotná, M.: Automatická detekce chyb v gramatické shodě v češtině [online]. Master's thesis, Masaryk University, Faculty of Arts, Brno (2018 [cit. 2018-10-30])
9. Šebesta, K.: CzeSL-SGT: korpus češtiny nerodilých mluvčích s automaticky provedenou anotací, verze 2 (2014)
10. Masopustová, M.: Automatická analýza srozumitelnosti textu [online]. Master's thesis, Masaryk University, Faculty of Arts, Brno (2018 [cit. 2018-10-30])

Comments on Czech Morphological Tagset

Karel Pala

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

Abstract. In the area of natural language processing the appropriate morphological annotation is necessary. In this paper we offer some comments on the Czech morphological tagset as used in the analyzer Majka that has been developed in NLP Centre (CZPJ) both for academic and commercial purposes relatively recently. We try to argue that the existing approach to the morphological annotation is not in agreement with the language reality and that the used solutions are motivated rather technically than theoretically. We think that it makes sense to consider some changes of the presently applied annotation principles that might, if applied, to improve the annotation accuracy.

Keywords: part-of-speech tagging; morphological analysis

1 Introduction

Morphology is as a rule the base for a number NLP applications and it is obvious that its descriptive adequacy is heavily determined by the annotation principles and consequently by the tagset depending on them.

For Czech, two tagsets have been available since the 90's, developed by two leading NLP groups: one produced in the Institute of Formal and Applied Linguistics at the Charles University in Prague [1] and another one in the NLP Centre at the Masaryk University in Brno [5].

In this paper we refer to comments on the version of the second tagset together with the underlying morphological database used by the analyzer Majka [7,8]. The tagset can be found in the Appendix B of the paper by Jakubíček et al. [2]

2 Annotation principles

The question that is essential: should the principle on which the used morphological annotation is based be holistic or partial? The question is not touched in Jakubíček's (et al) paper but it can be seen that linguistically complex expressions not behaving compositionally as e.g. *Karlovy Vary*, *vzhledem k (with regard to)*, *jestliže, ... pak (if ... then)*, *a to (and this)* or *budu číst (I will read)* are simply taken apart and later put together again, thus being analyzed twice. The approach

Aleš Horák, Pavel Rychlý, Adam Rambousek (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2018, pp. 15–19, 2018.

does not reflect the language reality and consequently generates a significant number of disambiguating errors. The main argument used for it is based on technical aspects of the annotation only and in our view is evading the essence of the annotation problem in Czech as such.

There is a convincing frequence evidence proving that so far used partial approach to the morphological annotation can hardly lead to an essential improvement of the present situation with annotation accuracy.

In our view the indicated problems with the disambiguation accuracy certainly include the following parts of speech:

PoS	tag	CzTenTen	Desam
adverbs	k6	278,172,710	49,469
conjunctions	k7	447,920,261	99,432
particles	k8	324,980,597	52,951
verbs	k5	694,012,081	126,067

As a data we use corpora CzTenTen12 with 4,175,089,441 tokens and also Desam with 874,354 tokens which show that the mentioned four parts of speech display really high frequences. This means that they (k5, k6, k7, k8) represent a large source of the disambiguation difficulties because of their strong mutual polyfunctionality. Apart from the high frequencies there is another relevant cause of the disambiguation errors (possibly up to 10 %) – many of these parts of speech are MWES as the examples above show, e.g. frequency of *a to* is 3,014,697 in CzTenTen12. The indicated simple facts can be generalized to support the thoughts about the possible changes of the so far used partial disambiguation strategy.

2.1 An example with MWE *a to* (and this)

We are well aware that proposing changes to an existing and established tagset can be understood as an unpopular step that implicates compatibility issues with the older tagset version(s). However, we would like to show an example taken from the corpus Desam that indicated holistic approach to MWEs with regard to disambiguation is worth of consideration.

We have chosen Czech MWE conjuction *a to (and this)* and found that it is tagged in manually disambiguated corpus Desam in a rather conflicting way as the Table 1 shows. We can see that 100 tokens of *a to* is tagged in different ways, particularly, we can see that *a (and)* is tagged as k8, k9 or not tagged at all. Similarly, the tagging of the second part of MWE *to (this)* varies considerably too as k3, k8 and k9, or not tagged at all.

Notice that the indicated conflicts would disappear if we decide to treat *a to (and this)* (and similar expressions) as one MWE unit which should be in this case tagged as a conjunction (k8).

Table 1: Various tags for *a to* and their frequency in Desam corpus.

not tagged	13
k8 + k3	60
k9 + k3	19
k8 + k9	3
k9 + k9	5

2.2 A Comment on Verbs (k5)

Few words should be said about verbs. They display the highest frequency among the mentioned parts of speech (in CzTenTen12 694,012,081 tokens, in Desam 126 067 tokens), thus they will be affected by the holistic approach relevantly.

In Czech this includes all analytical verb forms, i. e. future tense forms of imperfective verbs *budu číst (I will read/will be reading)*, past tense forms *četl jsem (I read/have read/was reading)*, conditional forms *četl bych (I would read/)*, which should be tagged as single units. It is obvious that it would require the massive re-tagging though we have to admit that the current tagging of the analytical verb forms does not generate too many disambiguation errors. However, if we want to be in concordance with the language reality we have to think seriously about the re-tagging of verbs. On the other hand, we realize that this re-tagging will considerably influence the structure of the existing parsers for Czech such as Synt or Set [3].

The question is if anybody can be found who would be able to try to undertake such demanding task. The language reality speaks for treating analytical verb forms consistently as single units, however, the technical consequences for a respective syntactic analysis would be extensive. It is, however, possible to go for a compromise and deal with the verb analytical forms in their current shape.

2.3 A Remark on Adverbs (k6), Prepositions (k7), Conjuctions (k8) and Particles (k9)

These frequent parts of speech are not inflected and as a rule they are difficult to disambiguate thanks to their high mutual ambiguity. They are not easy to classify because they often pass over between the individual categories and they can be disambiguated relativel reliably only in the particular contexts, which is sometimes difficult even for humans. As we could see above, it often happens that the conjunctions are recognized as particles, see *a to* above or adverbs as complex prepositions, e.g. *pokud (unless)*. It would be desirable to go through these ambiguities manually and disambiguate them, however, it is obviously not real because of their high frequency. Perhaps it would be helpful to collect

Table 2: Mapping of the Czech tagset to the Google Universal Tagset

universal tag	description	attributive tags
VERB	verbs (all tenses and modes)	k5.*
NOUN	nouns (common and proper)	k1.*
PRON	pronouns	k3.*
ADJ	adjectives	k2.*, k4.*xO, k4.*xR
ADV	adverbs	k6.*
ADP	adpositions (prepositions and postpositions)	k7.*
CONJ	conjunctions	k8.*
DET	determiners	(none)
NUM	cardinal numbers	k4.*xC
PRT	particles or other function words	k9.*
X	other: foreign words, typos, abbreviations	k0
.	punctuation	kI

all these PoSs and keep them as the particular lists making them a part of the database of Majka analyzer.

In existing Czech grammars, e.g. Karlík et al. [4], we can find subclassifications of the mentioned parts of speech – k6, k7, k8, k9, which are quite detailed but partly a bit overlapping. It would be useful to compare it with the corresponding subclassifications in Jakubíček's paper but this would be topic for a separate paper.

We would like to point out that in Jakubíček's paper some PoS' and their details are left aside, particularly prepositions (k7) and conjunctions (k8). It has to be stressed that they also include a number of MWES that should be re-tagged as single units. An attempt in this direction can be found in [9].

2.4 Mapping to the Google Universal Tagset

We decided to refer here to a mapping to the universal tagset created by joint effort of Google Research and Carnegie Mellon University [6]. The mapping is given in Table 2 and we take it over from Jakubíček's paper.

To remark: we assume that the comparison of the tags above should not be much influenced by the considered change of the annotation principles. In this respect we do not expect any relevant modification of the presented lists of tags.

3 Conclusions

We have offered some comments dealing with theoretically motivated changes to the attributive tagset for Czech language. Implicitly, we react to the paper by Jakubíček et al, however, we think that the revision proposed in it is not sufficient and that there is a time to consider more essential, holistic revision of the tagging principles and consequent re-tagging of existing corpora. We are

well aware that in fact such revision would be a painful and expensive new project as well but the challenge is here. This should lead to essential changes in tagging results and not only for Czech language.

The question remains whether the techniques exploiting neural networks will be able to deal with MWES in a holistic and descriptively adequate way. This too represents a topic for a separate paper.

Acknowledgements This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin infrastructure LM2015071 and OP VVV project CZ.02.1.01/0.0/0.0/16_013/0001781.

References

1. Hana, J., Zeman, D., Hajič, J., Hanová, H., Hladká, B., Jeřábek, E.: Manual for Morphological Annotation PDT. Tech. Rep. 27, Institute of Formal and Applied Linguistics, MFF UK, Prague, Czech Republic (2005)
2. Jakubíček, M., Kovář, V., Šmerk, P.: Czech morphological tagset revisited. Proceedings of Recent Advances in Slavonic Natural Language Processing pp. 29–42 (2011)
3. 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_15
4. Nekula, M., Rusínová, Z., Karlík, P.: Příruční mluvnice češtiny. Nakladatelství Lidové noviny (1995)
5. Pala, K., Rychlý, P., Smrž, P.: DESAM – Annotated Corpus for Czech. In: Proceedings of SOFSEM '97. pp. 523–530. Springer-Verlag (1997)
6. Petrov, S., Das, D., McDonald, R.: A universal part-of-speech tagset. Arxiv preprint ArXiv:1104.2086 (2011)
7. Šmerk, P.: Fast Morphological Analysis of Czech. In: Proceedings of the RASLAN Workshop 2009. Brno (2009)
8. Šmerk, P.: Towards Computational Morphological Analysis of Czech. Ph.D. thesis, Faculty of Informatics, Masaryk University, Brno (August 2010)
9. Žižková, H.: Improving Compound Adverb Tagging. In: Proceedings of the RASLAN Workshop 2018. Brno (2018)

Similarity between the Association Measures: a Case Study of Noun Phrases

Maria Khokhlova

Department of Mathematical Linguistics, St. Petersburg State University,
Universitetskaya emb. 11, 199034 St. Petersburg, Russia
m.khokhlova@spbu.ru

Abstract. Collocation extraction has gained much attention in natural language processing, its results are important in various areas of applied linguistics. The research focuses on a comparison between over a dozen of association measures based on a subset of the Russian Web corpus. The paper studies the automatically extracted Adj-Noun collocations. The aim of the experiments is two-fold. First, to examine the difference between statistical measures and second to find the most effective one for the Russian data. The former assumes the calculation of the Spearman's rank correlation coefficient and the latter implies the evaluation of the extracted lists against a Russian dictionary, i.e. identifying automatically extracted and manually collected collocations. The results are not such straightforward, one can distinguish between groups of measures that demonstrate a relative interchangeability. Also the produced bigrams can be considered as collocations by experts and thus may enrich dictionaries.

Keywords: collocability; collocations; corpora; statistics; statistical measures; gold standard

1 Introduction

Statistical tools play an active role in corpus linguistics and allow the researchers to extract data from texts supplying them with quantitative evaluation of the represented results. Collocation extraction is a task of natural language processing that is primarily based on statistical methods. Nowadays there are 82 statistical measures that are used for collocation extraction [1]. Usually they are called association measures and involve different principles. However, only a few of them were evaluated on linguistic data and even less applied to Russian. The aim of our experiments is to apply both well-known and not widespread association measures to the Russian data and to analyze possible similarity in the results.

The paper is structured as follows. In the next section we give a brief outline of the experiments. Then, we describe results of the analysis paying attention to the difference between the measures. Finally, the last section concludes the paper with discussion and gives suggestions for future work.

Aleš Horák, Pavel Rychlý, Adam Rambousek (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2018, pp. 21–27, 2018.

2 Experiments

By a collocation we understand a recurrent word combination and analyze bigrams involving Adj-Noun model. The experiments were based on a subset of the Russian web corpus (ruWaC) [2] that comprises 9.5 mln tokens. At first we extracted over 200,000 Adj-Noun combinations from the corpus and then cleared the list leaving 197,343 bigrams. Among other phrases we deleted those with punctuation marks, other parts-of-speech (due to the errors in lemmatization and morphological annotation) etc. There was a number of Belorusian examples that were also excluded (only those that use letters lacking in the Russian alphabet, e.g. ў and і). The following noun phrases can be seen as the examples after processing the list: *gumanitarnaya aktsiya* "humanitarian action", *legendarny geroy* "legendary hero", *professional'naya konsul'tatsiya* "professional consultation", *natsional'naya osobennost'* "national feature", *prikladnaya sistema* "applied system" etc.

We tested the following thirteen association measures implemented in the UCS tool [3]: mutual information (MI), MI2, MI3, t-score, z-score, minimum sensitivity (MS), Dice, Jaccard and geometric mean (gmean) coefficients, Fisher, Poisson and chi-squared tests, and logarithmic odds ratio. The examined coefficients belong to different categories, e.g. exact and asymptotic hypothesis tests, point estimates of association strength, and heuristic measures. The comprehensive survey of the measures was made in [3,4]. To the best of our knowledge there is no comparison of these methods applied to the Russian language, however several measures were applied on Russian texts [5,6].

The experiments involved the comparison between each pair of measures in order to determine to what extent they produce the same results. Also we evaluated the extracted list across the dictionary [7] that can be seen to a certain degree as a source of true collocations. Dictionary data were automatically lemmatized by MyStem [8].

3 Results

3.1 Spearman's Rank Correlation Coefficient

We analyzed the lists of all bigrams extracted by the measures and calculated the Spearman's rank correlation coefficient (r_s) in order to assess the similarity between the measures. The coefficient can take values from -1 up to +1 indicating a perfect negative or a perfect positive correlation respectively. Zero value indicates there is no correlation between the ranks. Also we used co-occurrence frequency to demonstrate how the measures rank the extracted collocations compared to the simple metrics. The point of our work was also to study if the frequency can be applied instead of statistical measures or can be used as a baseline for further work on improvement of collocation extraction methods. The Table 1 presents the correlation between the pairs of measures.

Several pairs of the measures have the highest correlation that equals to 1 and thus show the same rankings. They are as follows: 1) Jaccard and Dice coefficients; 2) Poisson and Fisher tests; 3) chi-squared test, gmean coefficient,

Table 1: Values of Spearman correlation coefficient

	freq.	Dice	Fisher	Jaccard	MI	MI2	MI3	MS	Poisson	chi.sq	gmean	odds	t-score	z-score
freq.		0.21	0.56	0.21	-0.13	0.08	0.26	0.22	0.56	0.08	0.08	-0.13	0.77	0.08
Dice	0.21		0.74	1.00	0.72	0.77	0.79	0.99	0.74	0.77	0.77	0.64	0.60	0.77
Fisher	0.56	0.74		0.74	0.71	0.85	0.93	0.69	1.00	0.85	0.85	0.70	0.91	0.85
Jaccard	0.21	1.00	0.74		0.72	0.77	0.79	0.99	0.74	0.77	0.77	0.64	0.60	0.77
MI	-0.13	0.72	0.71	0.72		0.97	0.90	0.65	0.71	0.97	0.97	0.99	0.47	0.97
MI2	0.08	0.77	0.85	0.77	0.97		0.98	0.71	0.85	1.00	1.00	0.96	0.64	1.00
MI3	0.26	0.79	0.93	0.79	0.90	0.98		0.73	0.93	0.98	0.98	0.89	0.77	0.98
MS	0.22	0.99	0.69	0.99	0.65	0.71	0.73		0.69	0.71	0.71	0.57	0.57	0.71
Poisson	0.56	0.74	1.00	0.74	0.71	0.85	0.93	0.69		0.85	0.85	0.70	0.91	0.85
chi.sq	0.08	0.77	0.85	0.77	0.97	1.00	0.98	0.71	0.85		1.00	0.96	0.64	1.00
gmean	0.08	0.77	0.85	0.77	0.97	1.00	0.98	0.71	0.85	1.00		0.96	0.64	1.00
odds	-0.13	0.64	0.70	0.64	0.99	0.96	0.89	0.57	0.70	0.96	0.96		0.46	0.96
t-score	0.77	0.60	0.91	0.60	0.47	0.64	0.77	0.57	0.91	0.64	0.64	0.46		0.64
z-score	0.08	0.77	0.85	0.77	0.97	1.00	0.98	0.71	0.85	1.00	1.00	0.96	0.64	

z-score and MI2. Analyzing the data one can suggest that the coefficients within one group share some features and thus behaviour in common.

Jaccard and Dice coefficients are often viewed as similar statistics due to their formulae, the obtained results prove the fact showing that they are full equivalents when it comes to rankings. We find it peculiar that the value of r_s between Poisson and Fisher tests is so high (1.0).

In our experiments another four measures showed a strong correlation between them. The chi-squared test and z-score belong to asymptotic hypothesis tests, gmean coefficient exemplifies the degree of association group while MI2 is a pure heuristic statistic. Squares of z-score values equal to those of chi-squared and hence they rank collocations in the same way. As it was mentioned in [3] the gmean measure uses the geometric mean and is similar to MI. This statement holds true and here we find that the coefficient has even more in common with MI2.

As it was expected the co-occurrence frequency showed the lowest correlation with other measures with the exception of t-score. The t-test statistic can be seen as closely linked to the observed co-occurrence frequency (usually labelled as O11) and hence the high value of r_s (0.77) supports the statement indicating that the pair has strong positive correlation and produces similar ranking to a certain degree. The same negative value was obtained by the co-occurrence frequency in the pairs with MI and odds ratio (-0.13). That fact can suggest that the ranking produced by the measures do not coincide with the one made by co-occurrence frequency and moreover can be slightly the opposite. MI is often referred to as an association measure that is sensitive to low frequencies and

tends to overestimate them ranking on top rare word combinations. Also we find other values of r_s. The ones produces by the co-occurrence frequency with MI2, chi-squared test, gmean and z-score are extremely low (0.08) and can be interpreted as no correlation. Fisher and Poisson behave in between and indicate a middle correlation with the co-occurrence frequency.

MI achieved fairly strong correlation with almost all the measures. The largest values ranging from 0.90 up to 0.99 were shown by the pairs with MI2, MI3, chi-squared test, gmean coefficient, odds ratio and z-score. It is no wonder that MI correlates considerably with its heuristic variants (namely MI2 and MI3) that give greater weight in the numerator to O11. However r_s is lower for the pair MI and MI3 (0.90). One could not anticipate that r_s between MI and t-score will be relative high (0.47) as they are usually described as statistics placing opposite collocations on top.

Values of r_s for the pairs of t-score with other measures vary from 0.46 up to 0.91. This leads us to the conclusion that the coefficient behaves something in between but also provides the rankings that are similar to those produced by other statistics.

3.2 Top Bigrams Analysis

As next step we aimed to evaluate the extracted collocations ranked by different measures across the dictionary [7]. We analyzed true collocations (true positives) from top 100, i.e. collocations found in the dictionaries. The results showed the lists do not only contain true collocations but also meaningless combinations and the phrases that can be viewed as collocations but were not described in the dictionary. We put more emphasis on the third group expecting that such word combinations can be useful for lexicographic needs. We calculated true positives in top 100 across the dictionary and expert evaluation, Table 2 presents the results.

Co-occurrence frequency. The co-occurrence frequency produces positive results placing on top the following high frequent collocations that present in the dictionaries and are also marked by the expert: *tsennaya bumaga* "negotiable paper", *uchebnoye zavedeniye* "educational institution", *soyedinennye shtaty* "United States", *domennoye imya* "domain name", *meditsinskaya pomoshch* "medical care". Its precision for top 100 is high (0.75).

MI. MI proves the results found in other works retrieving the largest number of typos, mistakes in annotation and foreign lexis. Even though the Adj-Noun pairs were initially preprocessed the measure ranged top bigrams with Belorussian words and hapax legomena. The expert analysis marked the following collocations extracted by MI as true ones: *snezhnaya baba* "snowman", *Rizhsky balzam* "Riga balsam" (name of a Latvian herbal liqueur), *parnikovy gaz* "greenhouse gas", *finansovy defolt* "financial default", *pitschevye dobavki* "food additive". Also a large number of low-frequency proper names were found the list.

Table 2: Effectiveness of the association measures

Association measure	Precision (dictionary, top 100)	Precision (expert, top 100)	Precision (dictionary, all set)
frequency	**0.25**	0.75	**0.05**
Dice	0.00	0.28	0.01
Fisher	**0.25**	**0.87**	**0.05**
Jaccard	0.00	0.28	0.01
MI	0.00	0.30	0.01
MI2	0.00	0.28	0.01
MI3	0.22	0.82	0.03
MS	0.00	0.28	0.01
Poisson	**0.25**	0.86	**0.05**
chi.sq	0.00	0.28	0.01
gmean	0.00	0.28	0.01
odds	0.00	0.10	0.01
t-score	**0.25**	0.79	**0.05**
z-score	0.00	0.25	0.01

MI3. MI3 places on top more common collocations than two previously mentioned measures and extract noun compounds: *krugly stol* "round table", *mobil'my telefon* "mobile phone", *detsky sad* "kindergarden", *zapisnaya knizhka* "notebook", *zdravy smysl* "common sense". The precision of the coefficient is higher evaluated both against the dictionary and expert data.

MI2, MS, chi-squared test, gmean, Dice and Jaccard coefficients. As it was stated above the Dice and Jaccard coefficients give different values for bigrams but provide the same ranking due to their nature. They place on top combinations that are not listed in the dictionary and thus show low precision even against expert evaluation (0.28). The same score for precision was achieved by other statistics, as they produced the same lists being extremely sensitive to low frequencies of either nodes, collocates or their combinations.

Fisher and Poisson tests. The results show that the best precision was obtained by Poisson and Fisher coefficients and it holds true both for the dictionary and expert evaluation. Top 100 lists a high number of collocations, e.g. *krayn'aya mera* "extreme measure", *molodoy chelovek* "young man", *aktsionernoye obshchestvo* "joint-stock company", *postsovetskoye prostranstvo* "post-Soviet space", *tamozhennaya poshlina* "customs duty". It should be also noted that two measures rank collocations slightly differently but however 99% represent the same word combinations.

T-score. T-score has extracted the majority of high-frequent word combinations, e.g. *rossiyskaya federatsiya* "Russian Federation", *okruzhayutschaya sreda* "environment", *general'ny direktor* "general director", *vneshnyaya politika* "external politics", *intellektual'naya sobstvennost'* "intellectual property". The precision is also high and equals to 0.79.

Z-score. Being a close relative of t-score, z-score however does not yield the same results extracting several other collocations. The produced list of bigrams for top 100 coincides with the one by MI.

Logarithmic odds ratio. The coefficient overestimates low-frequency bigrams, however it also ranks top word combinations in which either a node or a collocate has extremely high values compared to other word. For example, we find *velikaya armada* "Great (Spanish) Armada", where the collocate "velikiy" (in masculine) has an absolute frequency 1,967 and the node "armada" only 1, hence the collocation occurs once.

3.3 Discussion

As we can see the precision of the obtained results is extremely low evaluated against the lexicographic data. This poor result can be explained by the structure of the dictionary entries that influenced the list of the collocations we used as a gold standard. The word combinations were automatically extracted from the dictionary part intended for idioms and phrases. Also compared to the corpus size the number of true collocations was quite low and it also made its impact.

According to the Zipf's law a vast number of lexis has low frequency and hence in a large corpus there is a certain number of words that occur only once. In case of association measures it can be the case that the 1st rank produced by a coefficient will correspond to several word combinations. The results reveal that MI2, MS, chi-squared test, gmean, Dice and Jaccard coefficients rank hapax legomena on top 100 and thus the bigrams coincide.

Values shown in the second and third columns correlate to a certain degree, i.e. for non-zero scores against the dictionary we find also better results given by the expert evaluation.

Top 100 bigrams extracted by the co-occurrence frequency, MI3, t-score, Fisher and Poisson coefficients proved to include much more true collocations than other measures. And that leads us to the conclusion that collocations described in dictionaries are frequent ones and thus they can be retrieved only by the measures that have strong correlation with frequency.

4 Conclusion and Future Work

In summary, the research shows that co-occurrence frequency, MI3, t-score, Fisher and Poisson tests yield significant collocates that occur relatively frequently. In

most cases, they are the most reliable measures. Our approach has its limitations as every dictionary is personalized and does not provide a comprehensive description of collocability. We believe it is important to study the coefficients on other large corpora and compare between them. Also corpus data should be cleaned up as the majority of measures in the experiments were sensitive to typos and errors. The size of the gold standard used for the evaluation should be increased, for now it is not sufficient enough.

The results demonstrate a relative interchangeability between the association measures and can be used in future work on quantitative methods and their evaluation. The possible solution for the improvement of collocation extraction techniques is to combine the measures, e.g. use more complex rankings involving values of different measures or add other models (syntactic or vector).

In future work we plan to make experiments with other languages and offer a wider scale comparison between them implying also more association measures and other types of phrases.

Statistical measures for evaluating the strength between the items can be used for word sense disambiguation, in translation studies and CAT systems, for identification of synonyms and antonyms etc.

Acknowledgements. This work was supported by the grant of the President of Russian Federation for state support of scholarly research by young scholars (Project No. MK-2513.2018.6).

References

1. Pecina P. Lexical Association Measures. Collocation Extraction. Prague: Institute of Formal and Applied Linguistics, 2009.
2. ruWaC: Russian web corpus, `https://www.sketchengine.eu/russian-web-corpus`
3. Evert S. The Statistics of Word Cooccurrences: Word Pairs and Collocations. Dissertation, Institut für maschinelle Sprachverarbeitung, University of Stuttgart, `http://purl.org/stefan.evert/PUB/Evert2004phd.pdf`
4. Evert S. Association measures, `http://collocations.de`
5. Khokhlova M. V. Eksperimental'naja proverka metodov vydelelnija kollokacij [Evaluation of Methods for Collocation Extraction]. In Slavica Helsingiensia 34. Instrumentarij rusistiki: Korpusnye podhody. Eds. A. Mustajoki, M.V. Kopotev, L.A.Birjulin, J.J. Protasova. Helsinki, 2008, pp. 343-357.
6. Pivovarova L., Kormacheva D., Kopotev M. Evaluation of collocation extraction methods for the Russian language. In Quantitative Approaches to the Russian Language (ed. by M. Kopotev, O. Lyashevskaya, A. Mustajoki). London, New York: Routledge, 2018. P. 137–157.
7. Yevgen'yeva A.P. (ed.-in-chief). Slovar' russkogo jazyka [A Dictionary of the Russian Language] vol. 1-4, 2nd edition, revised and supplemented. Moscow: Russkij jazyk, 1981-1984.
8. MyStem, `https://tech.yandex.ru/mystem/`

Part II

Semantics and Language Modelling

Weighting of Passages in Question Answering

Vít Novotný and Petr Sojka

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

Abstract. Modern text retrieval systems employ text segmentation during the indexing of documents. We show that, rather than returning the passages to the user, significant improvements are achieved on the semantic text similarity task on question answering (QA) datasets by combining all passages from a document into a single result with an aggregate similarity score. Following an analysis of the SemEval-2016 and 2017 task 3 datasets, we develop a weighted averaging operator that achieves state-of-the-art results on subtask B and can be implemented into existing search engines. Segmentation in information retrieval matters. Our results show that paying attention to important passages by using a task-specific weighting method leads to the best results on these question answering domain retrieval tasks.

Keywords: passage retrieval; question answering; Godwin's law

1 Introduction

The standard bag-of-words vector space model [17] (VSM) represents documents in terms of word frequencies as vectors in high-dimensional real inner-product spaces. The model disregards word order, which immediately limits its ability to capture the meaning of a document. Nevertheless, the inner product provides a notion of document similarity that is well-understood and scales to large datasets. As a result, the VSM forms the basis of popular inverted-index-based search engines such as Apache Lucene [2], and improvements to the VSM will have an immediate impact on the performance of many text retrieval systems.

Long documents that cover a range of different topics provide a significant challenge for the VSM, since they are difficult to statically summarize, and deemed irrelevant to most queries. For that reason, Hearst and Plaunt [7] suggested "motivated segments", segmentation that reflects the text's true underlying subtopic structure, which often spans paragraph boundaries. The method for passage retrieval that requires a NLP-parser and a semantic representation in Roget-based vectors was suggested by Prince and Labadié [16]. Keikha et al. [8] evaluated passage retrieval methods and showed that the existing methods are not effective for the passage retrieval task, and also observe that the relative performance of these methods in retrieving answers does not correspond to their performance in retrieving relevant documents. Carmel et al. [3] developed contextualisation approach for passage retrieval.

Aleš Horák, Pavel Rychlý, Adam Rambousek (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2018, pp. 31–40, 2018.

Recently, we suggested several notable improvements. Based on machine learned word vector space semantic models the indexed documents are segmented into *semantically coherent* passages, to retrieve these passages instead of the original documents. In this paper, we focus on the frequent case, when the search engine is expected to retrieve full documents rather than just the passages relevant to a query. It would seem that, in this scenario, passages are useful for the summarization of results at best. Contrary to this intuition, we show that for question answering (QA) datasets, combining the evidence of similarity provided by the retrieved passages yields significant improvements on the text similarity task compared to the VSM on unsegmented documents. Our results are fully reproducible.[1] [14]

The paper is structured as follows: In Section 2, we review the related work. In Section 3, we give an overview of our system without delving into the specifics of our datasets. In Section 4, we describe our datasets and the experimental setup. Section 5 reports and interprets the results. We conclude in Section 6 with a summary of our results, and suggestions for future research.

2 Related work

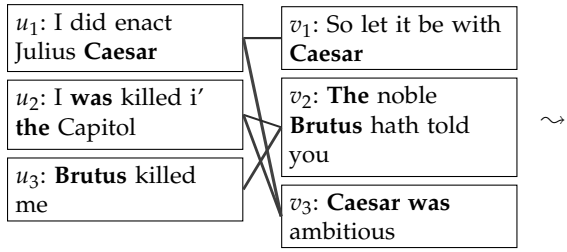
The notion of representing a document as a vector of weighted term frequencies, and estimating the similarity between two documents by the inner product was perhaps first researched by Salton and Buckley [17] during their work on the SMART information retrieval system. Several competing methods for assigning term weights and normalizing document vectors were proposed in literature. [5] In this paper, we consider those originally presented by Salton and Buckley [17].

The task of retrieving only the portions of a document that are relevant to a particular query is known as the passage retrieval and was perhaps first researched by O'Connor [15], who suggested retrieving document titles, abstracts, and figure captions in the absence of full texts. In the context of full-text retrieval, Khalid and Verberne [9] divide passage retrieval systems to those that index each passage as a separate document, which are the kind of retrieval systems that we target in this paper, and systems that first retrieve relevant documents and then retrieve passages from the retrieved documents, which is the inverse of our technique where we first retrieve passages and then aggregate the retrieved passages into documents. Beside disjoint passages, which we consider in this paper, Khalid and Verberne [9] also recognize *sliding passages* that can overlap arbitrarily.

Assessing the similarity of two structured documents by combining the evidence of similarity provided by their *structural elements* (i.e. passages) has already been explored in the context of XML document retrieval. In this paper, we draw inspiration from IBM Haifa's JuruXML system described by Mass et al. [11]. However, whereas XML documents have a tree structure, which makes it possible to compare passages based on structural similarity, our system makes no assumptions about the structure of passages.

[1] `https://github.com/witiko-masters-thesis/segmentation`

The removal and the weighting of *document zones* (i.e. passages) has been of interest to researchers in the fields of text summarization, feature selection, and text classification. In this paper, we consider the approach of Kołcz et al. [10] to reduce the number of considered passages.



$$\rightsquigarrow \quad \mathbf{M}_{uv} = \begin{pmatrix} 0.18 & 0 & 0.26 \\ 0 & 0.16 & 0.24 \\ 0 & 0.24 & 0 \end{pmatrix} \quad \rightsquigarrow \quad \oplus_k \ominus_l m_{kl} = 0.205$$

Fig. 1: Given query and result documents u and v consisting of passages u_1, u_2, u_3, v_1, v_2, and v_3, we compute a similarity matrix $\mathbf{M}_{uv}$ using the `bnc.bnc` tf-idf weighting scheme [17]. Using the operators $\oplus = \ominus = \text{wavg}_{\text{length}}$, we compute the aggregate score $S'(u, v)$.

3 System description

Our system takes a list of passages that form a single document and preprocesses them. If a passage k comes from a result document, then we store k in the database. If a passage k comes from a query document u, then we search the database for candidate passages l that have at least one term in common with k and we compute the similarity score $S(k, l)$. With the VSM, we first convert the passages k and l to the orthonormal coordinates of the passage vectors $\mathbf{v}_i$ and $\mathbf{v}_j$. In this paper, we perform the conversion using the `bfx.tfx` tf-idf weighting scheme suggested by Salton and Buckley [17] for short and homogeneous passages, which fits well with our QA datasets. The similarity score S between the two passages then corresponds to the inner product between $\mathbf{v}_k$ and $\mathbf{v}_l$, i.e. $S(k, l) = \langle \mathbf{v}_k, \mathbf{v}_l \rangle = \mathbf{v}_k^\mathsf{T} \mathbf{v}_l$.

If we performed no segmentation, then a passage corresponds directly to a document. In this scenario, we return to the user a list of candidate passages l ordered in the decreasing order of $S(k, l)$, where k is the single query passage. If we performed segmentation, then for each document v (result document) containing at least one candidate passage l, we compute a similarity matrix $\mathbf{M}_{uv}$, where every row contains the similarity scores between a single query passage from u and all passages from v (result passages) and every column contains the

scores between all query passages from u and a single result passage from v. We seek an aggregate scoring function $S'(u, v)$ defined in terms of the elements of $\mathbf{M}_{uv}$, such that the ordering of result documents induced by S' correlates with the relevance of the result documents v to the information need behind the query document u.

Let $\oplus$ and $\ominus$ be weighted averaging operators on $\mathbb{R}$ and let m_{kl} denote the value of a matrix $\mathbf{M}_{uv}$ in the row and column corresponding to the query and result passages k and l. Then we can express our aggregate scoring function S' as $\oplus_{\text{query passage } k \in u} \ominus_{\text{result passage } l \in v} m_{kl}$ (see Fig. 1). In our experiments, we evaluated two operators, namely $\text{wavg}_{\text{length}}$, which assigns weights proportional to the number of tokens in a passage, and $\text{wavg}_{\text{Godwin}}$ that we will develop as a part of our dataset analysis (see Section 4.3).

When the number of passages in a document is large, the computation of $\mathbf{M}_{uv}$ can be prohibitively slow. One possible approach to speeding up the retrieval is to avoid the segmentation of query documents and to segment only the result documents instead. This is the standard approach in semi-structured XML retrieval [11], where the query constitutes only a single branch of an XML document tree. An alternative approach would be to assume that the similarity score between the query passages and the non-candidate result passages is close to zero. Instead of retrieving all results passages from v, we would fill the columns corresponding to non-candidate result passages with zeros.

4 Experimental setup

In this section, we will describe the datasets that we used for our experiments. We will then describe how we preprocessed, and analyzed the datasets.

4.1 Datasets

We evaluated our system on the SemEval-2016 and 2017 task 3 subtask B QA datasets. These datasets consist of discussion threads from the Qatar Living[2] internet forum. Given an original question, and a set of ten candidate threads, the task is to rank the candidate threads by their relevance to the original question. A candidate thread contains a related question, and the first ten comments in the thread. The performance of a system is evaluated by its mean average precision (MAP) according to the relevance judgements from the datasets. [13,12]

The SemEval-2016 task 3 subtask B datasets consist of a training dataset (267 original questions, 1,790 threads), a dev dataset (50 original questions, 244 unique threads), and a test dataset (70 original questions, 327 unique threads). The winning SemEval-2016 task 3 subtask B submission was from UH-PRHLT-*primary* [6] with a MAP score of 76.70 who ranked threads using support vector machines (SVMs), and crafted features. The SemEval-2016 task 3 subtask B information retrieval (IR) baseline had a MAP score of 74.75.

[2] `http://www.qatarliving.com/forum`

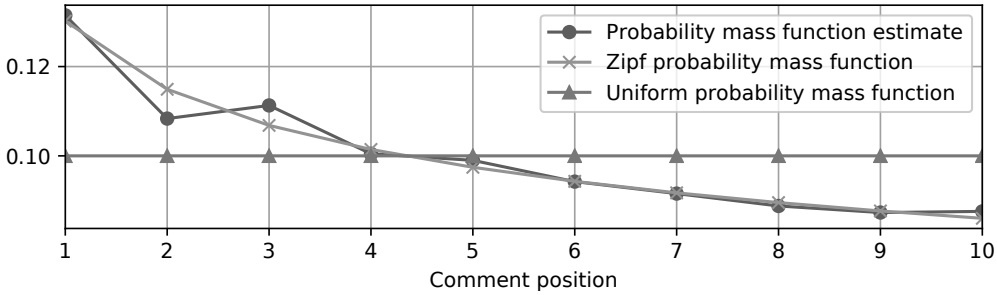


Fig. 2: Probability mass function (PMF) estimate $\hat{P}(\text{at position } i \mid \text{relevant})$ plotted along the PMF of the Zipf distribution with parameters $n = 10$ and $s = 0.18$. If the position of a comment and its relevance were independent, we would expect the PMF estimate to be uniformly distributed. Since $P(\text{at position } i)$ is uniformly distributed, $P(\text{at position } i \mid \text{relevant})$ is proportional to $P(\text{relevant} \mid \text{at position } i)$.

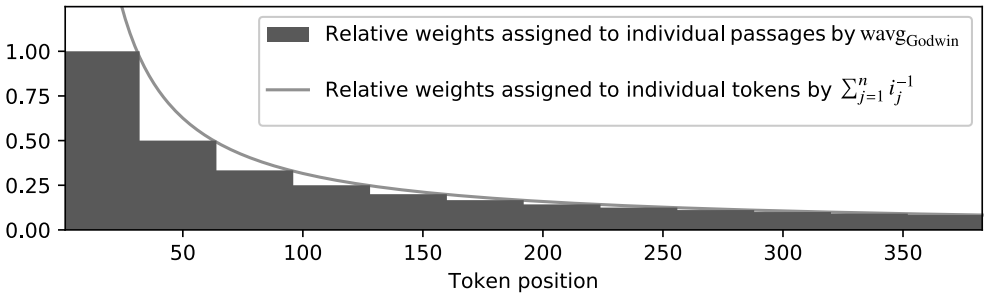


Fig. 3: The relative weights assigned to the individual passages by the $\text{wavg}_{\text{Godwin}}$ weighted averaging operator and the relative weights assigned to the individual tokens by the Godwin term weighting method. The figure assumes the mean number of tokens per a thread in the subtask A unannotated datasets (383 tokens), a uniform number of tokens in a passage, and the `txx.txx` tf-idf weighting scheme.

SemEval-2017 task 3 subtask B uses the same training and dev datasets as SemEval-2016 with the provision that the SemEval-2016 test dataset can be used for training. A new test dataset (88 original questions, 293 unique threads) has also been added. The SemEval-2017 task 3 subtask B winning configuration was *SimBow-primary* [4] with a MAP score of 47.22 who ranked threads using logistic regression and unsupervised similarity measures. The SemEval-2016 task 3 subtask B IR baseline had a MAP score of 41.85.

For statistical analysis, we used the SemEval-2016, and 2017 task 3 subtask A datasets. These datasets contain equivalent data as the subtask B training datasets (2,654 questions), but now the relevance judgements assess how relevant a comment is to a question. For language modeling, we used the unannotated

SemEval-2016 and 2017 task 3 subtask A datasets (189,941 questions, 1,894,456 comments).

4.2 Language modeling, and segmentation

Texts in datasets were lower-cased, stripped of images and URLs, and tokenized on white spaces and punctuation. Tokens shorter than two characters or longer than 15 characters were removed to cope with the problem of missing and extra whitespaces in questions, and comments. Using the existing structure of the datasets, every original question was split into two passages corresponding to the question subject and text, and every candidate thread was split into twelve passages corresponding to the related question subject and text, and the initial ten comments.

Since the annotated datasets did not contain enough text to build a proper language model, we used the unannotated subtask A datasets to obtain the collection-wide statistics required to compute the scoring function S described in Section 3.

4.3 Dataset analysis

In 1991, the American attorney and author Mike Godwin formulated[3] a rule that "as a Usenet discussion grows longer, the probability of a comparison involving Nazis or Hitler approaches one." An immediate corollary would be that as an online discussion grows longer, the probability of a relevant contribution approaches zero. We were curious whether the datasets would confirm these observations. We used the subtask A relevance judgements to estimate the probability mass function (PMF) $P(\text{at position } i \mid \text{relevant})$ for $i = 1, 2, \ldots, 10$. Since there is a uniform number of comments at each position i, i.e. $P(\text{at position } i) = 0.1$, we would expect $P(\text{at position } i \mid \text{relevant})$ to be also uniform if the position of a comment and its relevance are independent. We show in Fig. 2 that this expectation is implausible, and that there appears to be an inverse relationship between the position of a comment and its relevance.

To see if this relationship was statistically significant, we modeled the number of relevant comments at each position i as a binomial random variable $X_i \sim \text{Bi}(n, \theta_i)$ with a known number of trials $n = 2{,}410$, and an unknown probability of success θ_i. We then used the one-tailed Fisher's exact test to reject the following system of null hypotheses at 5% significance:

$$H_0^{(ij)} : \theta_i = \theta_j, \text{ where } i, j = 1, 2, \ldots, 10, \;\; i < j$$

We rejected $H_0^{(ij)}$ for any $j - i > 3$. We failed to reject $H_0^{(ij)}$ for $(i, j) = (2, 3)$, $(4, 5)$, $(5, 6)$, $(6, 7)$, $(7, 8)$, $(7, 9)$, $(7, 10)$, $(8, 9)$, $(8, 10)$, and $(9, 10)$. We used the procedure of Benjamini and Hochberg [1] to control the false discovery rate due to multiple testing.

[3] `news:1991Aug18.215029.19421@eff.org`

Table 1: Results for the four evaluated configurations (one primary, and three contrastive) on the SemEval-2016 task 3 subtask B test dataset. The primary configuration is highlighted in bold, whereas the winning SemEval-2016 task 3 subtask B submission and the IR baseline are highlighted in italics.

Configuration	Segm.	Text summ.	S. f. S	Aggregate s. f. S'	MAP
Primary	**Yes**		`bfx.tfx`	$\oplus = \mathbf{wavg_{length}}$, $\ominus = \mathbf{wavg_{Godwin}}$	**76.77**
SemEval-2016 task 3 subtask B winner (UH-PRHLT-primary)					*76.70*
Third contrastive	No	FirstTwoPara	`bfx.tfx`		75.21
SemEval-2016 task 3 subtask B IR baseline					*74.75*
First contrastive	No		`bfx.tfx`		73.94
Second contrastive	No		`bfx.tfx`, Godwin		70.28

This discovery led us to develop the $\text{wavg}_{\text{Godwin}}$ weighted averaging operator, which assigns a weight proportional to i^{-1} to a passage at position i in accordance to *Zipf's law*. This decreases the effect of comments that are likely to be irrelevant. Under the hypothesis that relevant comments are more likely to contain important terms that describe the meaning of a document, this operator pays attention to scores between those passages that are likely to contain important terms.

Since term weighting is conceptually and computationally simpler than segmentation and result aggregation, we wanted to verify that the segmentation is meaningful and that the relevance loss occurs at passage boundaries rather than at term boundaries. For that reason, we developed the *Godwin* term weighting method for the VSM scoring function S. For each term t at positions $i_1, i_2, \ldots, i_n$ in a document, the method multiplies the term frequency term-weighting component [17] with a weight proportional to $\sum_{j=1}^{n} i_j^{-1}$. It is easy to show that, given the right choice of the term frequency component (`t`) and the collection frequency component (`x`), the scoring function S induces the same ordering on unsegmented threads as the aggregate scoring function S' with $\oplus = \text{wavg}_{\text{length}}, \ominus = \text{wavg}_{\text{Godwin}}$ would if the threads were segmented to one passage per a token (see Fig. 3).

5 Results

The results for the four evaluated configurations are shown in Table 1 and Table 2. The primary configuration performs segmentation with the $\oplus = \text{wavg}_{\text{length}}, \ominus = \text{wavg}_{\text{Godwin}}$ operators and consistently outperforms the winning SemEval task 3 subtask B submissions. This shows that the $\text{wavg}_{\text{Godwin}}$ weighted averaging operator works well with our datasets and hopefully with QA datasets in general.

Table 2: Results for the four evaluated configurations (one primary, and three contrastive) on the SemEval-2017 task 3 subtask B test dataset. The primary configuration is highlighted in bold, whereas the winning SemEval-2017 task 3 subtask B submission and the IR baseline are highlighted in italics.

Configuration	Segm.	Text summ.	S. f. S	Aggregate s. f. S'	MAP
Primary	**Yes**		`bfx.tfx`	$\oplus = \mathbf{wavg}_{\mathbf{length'}}$, $\ominus = \mathbf{wavg}_{\mathbf{Godwin}}$	**47.45**
SemEval-2017 task 3 subtask B winner (SimBow-primary)					*47.22*
Third contrastive	No	FirstTwoPara	`bfx.tfx`		44.67
SemEval-2017 task 3 subtask B IR baseline					*41.85*
Second contrastive	No		`bfx.tfx`, Godwin		37.18
First contrastive	No		`bfx.tfx`		36.82

The three contrastive configurations do not perform segmentation. The first configuration corresponds to the base system with no extra preprocessing or weighting and is consistently outperformed by the remaining configurations as well as by the SemEval task 3 subtask B IR baselines. The second configuration uses the Godwin term weighting method developed in Section 4.3 and performs on-par with the first contrastive configuration. This shows that the segmentation to semantically coherent passages is meaningful and cannot be replaced with simple term weighting. The third configuration uses the *FirstTwoPara* text summarization technique [10], which reduces a thread to the question subject, the question text, and the first comment, and outperforms all the remaining contrastive configurations as well as the SemEval task 3 subtask B IR baselines. This shows that removing all but the first comment improves the signal-to-noise ratio, but at the cost of losing important terms.

6 Conclusion and future work

Segmentation matters and so does careful weighting. By combining both, we were able to achieve state-of-the-art results on the SemEval-2016 and 2017 task 3 subtask B QA datasets using the standard bag-of-words vector space model without any semantic modeling. Our technique can be readily implemented into existing inverted-index-based search engines.

We have shown that there exists a statistically significant relationship between the position of a comment and its relevance in the SemEval-2016 and 2017 subtask A datasets. Investigating whether such a relationship exists in other QA datasets and other datasets in general will provide us with new insights to the dynamics of online discourse and lead to more effective retrieval systems.

In this paper, we assumed that passages were disjoint. This is not true in general and future research should extend our technique to sliding passages [9] that can overlap arbitrarily.

Acknowledgments. TAČR Omega grant TD03000295 is gratefully acknowledged.

References

1. Benjamini, Y., Hochberg, Y.: Controlling the False Discovery Rate: a Practical and Powerful Approach to Multiple Testing. Journal of the Royal Statistical Society. Series B (Methodological) pp. 289–300 (1995), https://www.jstor.org/stable/2346101
2. Białecki, A., Muir, R., Ingersoll, G., Imagination, L.: Apache Lucene 4. In: SIGIR 2012 workshop on open source information retrieval. p. 17 (2012)
3. Carmel, D., Shtok, A., Kurland, O.: Position-based Contextualization for Passage Retrieval. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management. pp. 1241–1244. CIKM '13, ACM, New York, NY, USA (2013), https://doi.acm.org/10.1145/2505515.2507865
4. Charlet, D., Damnati, G.: SimBow at SemEval-2017 Task 3: Soft-Cosine Semantic Similarity between Questions for Community Question Answering. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). pp. 315–319. Association for Computational Linguistics (2017), https://www.aclweb.org/anthology/S17-2051
5. Chisholm, E., Kolda, T.G.: New Term Weighting Formulas for the Vector Space Method in Information Retrieval. Tech. rep., Computer Science and Mathematics Division, Oak Ridge National Laboratory, Tennessee, United States (Mar 1999), https://doi.acm.org/10.2172/5698
6. Franco-Salvador, M., Kar, S., Solorio, T., Rosso, P.: UH-PRHLT at SemEval-2016 Task 3: Combining Lexical and Semantic-based Features for Community Question Answering. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). pp. 814–821. Association for Computational Linguistics (2016), https://www.aclweb.org/anthology/S16-1126
7. Hearst, M.A., Plaunt, C.: Subtopic Structuring for Full-length Document Access. In: Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 59–68. SIGIR '93, ACM, New York, NY, USA (1993), https://doi.acm.org/10.1145/160688.160695
8. Keikha, M., Park, J.H., Croft, W.B., Sanderson, M.: Retrieving Passages and Finding Answers. In: Proceedings of the 2014 Australasian Document Computing Symposium. pp. 81:81–81:84. ADCS '14, ACM, New York, NY, USA (2014), https://doi.acm.org/10.1145/2682862.2682877
9. Khalid, M.A., Verberne, S.: Passage Retrieval for Question Answering using Sliding Windows. In: COLING 2008: Proceedings of the 2nd Workshop on Information Retrieval for Question Answering. pp. 26–33. Association for Computational Linguistics (2008), https://aclweb.org/anthology/W08-1804
10. Kołcz, A., Prabakarmurthi, V., Kalita, J.: Summarization as feature selection for text categorization. In: Proceedings of the ACM CIKM Conference. pp. 365–370. ACM (2000)
11. Mass, Y., Mandelbrod, M., Amitay, E., Carmel, D., Maarek, Y.S., Soffer, A.: JuruXML – an XML retrieval system at INEX'02. In: INEX Workshop. pp. 73–80 (2002)

12. Nakov, P., Hoogeveen, D., Màrquez, L., Moschitti, A., Mubarak, H., Baldwin, T., Verspoor, K.: SemEval-2017 task 3: Community question answering. In: Proceedings of the 11th International Workshop on Semantic Evaluation. pp. 27–48. SemEval '17, ACL, Vancouver, Canada (Aug 2017)
13. Nakov, P., Màrquez, L., Moschitti, A., Magdy, W., Mubarak, H., Freihat, A.A., Glass, J., Randeree, B.: SemEval-2016 task 3: Community question answering. In: Proceedings of the 10th International Workshop on Semantic Evaluation. SemEval '16, ACL, San Diego, USA (Jun 2016)
14. Novotný, V.: Vector Space Representations in Information Retrieval. Master's thesis supervised by Petr Sojka, Faculty of Informatics, Masaryk University, Brno, Czech Republic (2018), https://github.com/witiko-masters-thesis/thesis
15. O'Connor, J.: Retrieval of Answer-Sentences and Answer-Figures from Papers by Text Searching. Information Processing & Management 11(5–7), 155–164 (1975)
16. Prince, V., Labadié, A.: Text Segmentation Based on Document Understanding for Information Retrieval. In: Kedad, Z., Lammari, N., Métais, E., Meziane, F., Rezgui, Y. (eds.) Natural Language Processing and Information Systems. pp. 295–304. Springer, Berlin, Heidelberg (2007), https://doi.acm.org/10.1007/978-3-540-73351-5_26
17. Salton, G., Buckley, C.: Term-Weighting Approaches in Automatic Text Retrieval. Information Processing and Management 24, 513–523 (1988)

Towards Czech Answer Type Analysis

Daša Kušniráková and Marek Medveď

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

Abstract. In this paper, we introduce two answer type detection systems for Czech language. Based on the input question, the goal of these tools is to recognise the question type and extract an appropriate answer type. Except for the same goal, these systems are completely different. The first one is a rule based system utilising Czech Wordnet for hypernym detection. The second one uses a machine learning approach in form of a neural network. We present architectures of these two systems and offer a detailed evaluation on more than 8,500 question-answer pairs using the SQAD v2.1 benchmark dataset.

Keywords: question answering; question classification; answer classification; Czech; Simple Question Answering Database; SQAD

1 Introduction

Open domain question answering (QA) systems have seen a great progress in recent years. Using neural networks models [1,2] and large datasets, e.g. SQuAD [3], the systems have become more and more usable.

The majority of QA tools consists of several modules that contribute to the final system performance. In this paper, we present answer type detection module that usually appears at the beginning of the processing pipeline mostly on the pre-processing level, whose main task is to determine the answer type according the input question. We introduce two implementations of such answer type detection tool. The first one is represented by a system based on rules enriched by a hypernymic dictionary, whereas the second one utilises a recurrent neural network model. Both systems will be tested inside the AQA system [4,5] pipeline and the answer type detection is expected to improve the decision process in the Answer extraction module of AQA (see Figure 1).

The following chapters provide a detailed specification of the rule based as well as the machine learning based system. In the last section, we offer a thorough evaluation of both systems, for which the benchmarking dataset SQAD v2.1 [6] database consisting of 8,566 questions-answer pairs has been used.

Aleš Horák, Pavel Rychlý, Adam Rambousek (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2018, pp. 41–51, 2018.

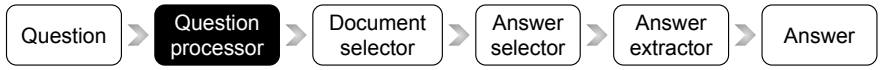


Fig. 1: AQA system visualisation

2 Question and Answer Type Detection

Same as for other classification issues, different approaches can be applied even when dealing with question and answer type detection. While some systems have been developed using rule based approach [7,8], machine learning based approach [9,10] has become more popular over the years. Because of the need to improve the overall performance of the AQA system, two systems have been developed, while each is based on a different approach. The rule based as well as the machine learning based system are described below.

2.1 Rule Based System

The rule based approach has been used from the early beginnings of dealing with QA type detection – e.g. a tool for question classification introduced in [11] capable of distinguishing between three classes. Even though the approach is on the decline these days due to more effective methods, rule based systems can still achieve satisfactory results. The systems introduced in [7,8] are able to classify required answer types with precision up to 83%.

The core of the developed rule based system introduced in this paper is formed by a set of hand-written rules approaching different features extracted from the question during the preprocessing phase. Such features include lexical features, POS tags, the *question keyword* and its hypernyms. The keyword is represented by the main (head) question meaning noun.

The keyword extraction algorithm is based on the following three rules:

- The question keyword candidate is the first noun after the relative pronoun *"který"* (which) or *"jaký"* (what), if such relative pronoun is present in the question and is not part of a relative sentence.
- Otherwise the first noun after the first verb in question is selected as the candidate for question keyword.
- The candidate becomes the final keyword unless it is one of words *"název"* (title), *"pojem"* (concept), *"termín"* (term), *"typ"* (type), *"část"* (part), or *"větev"* (branch). Otherwise the first following noun after the selected candidate is returned as the final keyword.

Keyword hypernyms are obtained by means of the Czech Wordnet API [12] in a two-step process. The Czech Wordnet is queried for the first time to find all

possible senses of the keyword extracted from the question and subsequently, the Wordnet API is queried again to create a list of hypernyms for three[1] most common word senses.

After all features have been obtained during the preprocessing phase, type detection rules are applied step by step to the question. If the rule's conditions are met by the question which is being classified, the appropriate labels representing question and answer types are returned. A schematic description of the QA type detection process is presented in Figure 2.

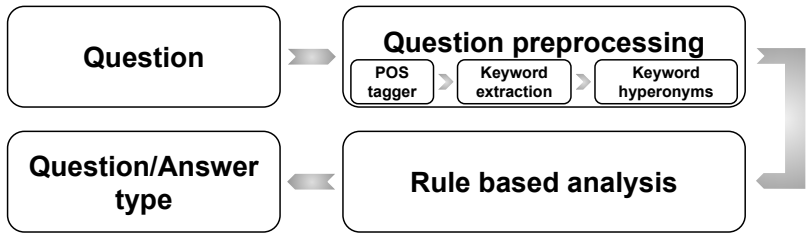


Fig. 2: Rule based question/answer type detection schema

The rules themselves are formed by any combination of the features recognized during the preprocessing phase. These include:

- keyword hypernym match:
  Example: `"<word>" in keyword.hypernym`
- important word recognition:
  Example: `"<word>" == words.lemma_at_index(0)`
  -> the first word in the sentence is the specified word
- question structure match:
  Example: `"k2" in words.tag_at_index(1)`
  -> the second word in the sentence is an adjective[2]

All pieces of information gained during the whole process of QA type detection (including the preprocessing as well as rule application phase) for a particular question can be seen in Figure 3.

[1] The number has been determined by testing of the overall performance of the system. Creating a list of hypernyms for both lower and higher number of word senses affects the performance in a negative way as the list becomes either too narrow or too broad, respectively.

[2] see [13] for more information about the POS tagset.

question: 'Jak se jmenovala první manželka Miloše Formana?'
(What was the name of the first wife of Miloš Forman?)
keyword: 'manželka' (wife)
hypernyms: ['manželka', 'jednotlivec', 'osoba', 'bytost', 'organismus']
(wife, individual, person, being, organism)
rule: `(PERSON; PERSON) -> "osoba" in keyword.hypernym`

Fig. 3: A question/answer type rule example: *if* `"osoba"` *(person) is one of the question's keyword hypernyms, then the question type is* `PERSON` *and the answer type is also* `PERSON`.

2.2 Machine Learning Based System

In comparison to the rule based approach, machine learning makes the process of question analysis and classification more automatic. Apart from that, these systems are able to achieve results comparable or even outperforming with other approaches. In the systems introduced in [9] or [10], the accuracy reaching for fine-grained classes around 90%, for coarse-grained classes even up to 95%.

In addition to the rule based system described in the previous section, a system for question and answer type detection based on machine learning – a Long Short-Term Memory (LSTM) network has been developed, too. Recurrent neural network has been chosen due to its ability to handle sequential input data of any length, while the LSTM unit is capable of dealing with the exploding and vanishing gradient problem.

Our proposed LSTM model consists of four layers. It contains two stacked LSTM layers with a dropout layer applied in between, and a linear layer. A visual representation of the model's architecture is shown in Figure 4. The two-layered LSTM architecture has outperformed both single and a three stacked LSTM layers by more than 80% and 5% in experimental attempts, respectively. The model has been trained with the usage of cross-entropy as loss function and with 40 epochs, batch size of 64, dropout rate of 0.5 and learning rate of 0.001 as its hyperparameters.
The process of question and answer classification is performed in the following steps:

- The input question is split into individual words, which are subsequently converted into dense, 100-dimensional vectors. The vectors are obtained from pre-trained Fasttext word embeddings trained on Czech corpora of more than 10 milliard words.
- Words in the form of 100D vectors represent the input to the LSTM model. They are processed one-by-one by LSTM layers. For each sequence, only the most recent timestep (affected by the previous ones) of the LSTM network is passed to the Linear layer.
- The Linear layer transforms the LSTM output to a vector of scores for each question and answer type combination, which is created by the cartesian

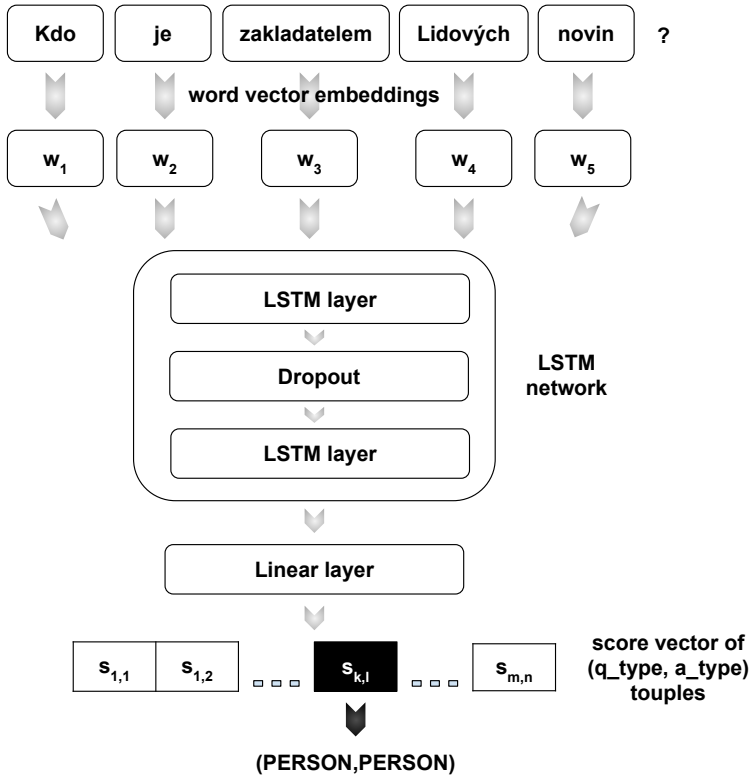


Fig. 4: LSTM network visualisation

product of all question and answer types. Considering there are possible 10 question and 10 answer classes, such vector created in the Linear layer is then 100-dimensional.

– The searched question and answer type is then determined from the position of the maximal score found in the vector returned from the Linear layer. The higher is the score, with the higher certainty is the particular combination of classes predicted by the model.

3 SQAD Database

The Czech Simple Question Answering Database, or SQAD [14,15], is a QA benchmarking dataset resource consisting of manually processed and manually annotated question-answer pairs. SQAD, originally created from Czech Wikipedia articles, now represents a consistent and representative data source for any model training and tool evaluation needs.

The SQAD v2.1 database currently contains 8,566 question-answer pairs which are related to the content of 3,149 Czech Wikipedia articles. The SQAD database is organised in structured records (one QA pair corresponds to one record) consisting of 6 items:

- the *question*,
- the *correct answer* (as can be extracted from the document),
- *answer selection* – the context of the correct answer, one or two sentences,
- the full *article text*
- the *source URL* in Wikipedia
- *question-answer metadata* containing *types of the question and the correct answer*.

All texts have been manually corrected and enriched by base word forms (lemma) and Part-Of-Speech (POS) annotation (DEsamb [16,13]).

The dataset contains annotation with classification of each record into categories for the question type and the actual correct answer type. The sets of possible types [15] took inspiration from the large benchmark dataset for English, the Stanford Question Answering Dataset [3].

The distribution of question classes over answer classes is displayed in Table 1.

Table 1: SQAD v2.1 distribution matrix of question and answer types

Q type /A type	PER.	DENOT.	ENT.	OTHER	ORG.	D./T.	LOC.	NUM.	ABB.	Y/N
PERSON	1,016	0	2	0	3	0	2	0	0	0
ENTITY	20	101	1,031	378	204	1	7	1	2	0
ADJ_P.	7	0	8	216	0	0	0	2	0	0
D./T.	0	0	1	2	0	1,844	0	4	0	0
LOC.	1	0	14	5	3	0	1,501	0	0	0
NUM.	1	0	0	2	0	0	0	910	0	0
ABB.	0	1	0	0	0	0	0	0	80	0
CLAUSE	1	0	27	205	6	0	1	1	0	0
VERB_P.	2	0	1	0	0	0	0	0	0	937
OTHER	2	0	1	11	0	0	0	0	0	1

4 Evaluation

This section offers an evaluation of the question-answer types detection on SQAD v2.1 database for both rule based and machine learing based systems. For correct evaluation, the database has been divided into parts. For the rule based system, the dataset has been split into training and testing set, for the LSTM network into training, evaluation and testing set. All parts are properly

balanced to maintain each question/answer type present in each division. The exact numbers of records in training, evaluation and testing sets for each system are listed in Table 2.

Table 2: Number of records after dataset splitting

	training	evaluation	testing
Rule based system	4,279	-	4,287
LSTM network	7,011	735	820

The final evaluation of the rule based system as well as the LSTM network is present in Table 3 and Table 4, respectively. The evaluation is calculated by weighted average process that is more suitable for multiclass classification setting.

Table 3: QA types detection evaluation – rule-based system

	precision	recall	F1
question t.	88.77%	87.79%	88.28%
answer t.	85.05%	84.52%	84.78%
both types	82.43%	82.93%	82.68%

Table 4: QA types detection evaluation – LSTM network

	precision	recall	F1
question t.	91.59%	90.73%	91.16%
answer t.	89.76%	89.14%	89.45%
both types	86.15%	87.07%	86.61%

4.1 Rule Based System

The recall of both types detection is **82.93%**, while the combined precision is 82.43% with question type precision of 88.77% and the answer type precision of 85.05%. The question type detection achieves recall of 87.79% and F1 measure going up to 88.28%. A detailed confusion matrix of all the expected and predicted question types is presented in Figure 5. According to the results, it can be seen that ENTITY class is among the most complex ones as entities can be expressed in several ways. A detailed evaluation of the answer type detection is present in Figure 6, where the most confusing classes for the system are ENTITY, OTHER and PERSON. This may call for further specification of the members of the OTHER class.

Table 5: Question type confusion matrix – rule-based system

	expected									
predicted	AB	APHR	CL	D/T	ENT	LOC	NUM	OTH	PER	VPHR
ABBR.	37	1	1	0	19	3	1	0	0	0
ADJ_P.	1	52	4	0	49	6	6	0	4	0
CLAUSE	1	0	35	0	14	4	0	0	5	0
D/T	0	0	1	916	16	0	2	0	1	1
ENTITY	0	44	71	3	685	41	13	2	40	8
LOC.	0	6	1	0	22	695	3	0	3	1
NUM.	1	4	1	4	8	0	422	0	0	0
OTHER	0	1	3	2	25	7	7	5	3	6
PERSON	0	8	3	0	33	6	2	0	455	0
V_PHR.	0	0	0	0	0	0	0	0	0	454

Table 6: Answer type confusion matrix – rule-based system

	expected									
predicted	AB	D/T	ENT	LOC	NUM	ORG	OTH	PER	DEN	Y/N
ABBR.	37	0	9	3	1	1	9	2	0	0
D/T	0	915	7	0	2	1	8	1	2	1
ENTITY	0	2	405	32	14	19	191	40	10	5
LOC.	0	0	7	693	3	9	15	3	0	1
NUM.	1	3	3	0	423	0	9	0	1	0
ORG.	1	0	30	5	0	61	24	6	0	0
OTHER	2	2	46	16	14	10	138	19	3	7
PERSON	0	0	12	7	2	13	18	452	3	0
DENOT.	0	0	1	1	1	1	3	0	38	0
YES_NO	0	0	0	0	0	0	0	0	0	454

4.2 LSTM Network

In the machine learning based system, the recall of both types is **87.07%** and the combined precision is 86.15%. The answer type precision is going up to 89.76%, while the question type detection achieves high precision going up to 91.59%. In general, it can be stated that the LSTM outperforms the rule based system by 2.7-5 points in each score according to the results. A detailed evaluation of question type detection is provided by Table 7. The deviation is most apparent for OTHER class, whose results have been affected by misclassifying the only record of this class. Table 8 presents the answer type detection results, where the most remarkable deficiencies can be seen namely in ENTITY, OTHER, and PERSON classes.

The LSTM network outperforms the rule based system according to the most recent results presented above even though no changes in hyperparameters of

the LSTM network have not been properly tested yet. The introduced LSTM system represents our first prototype so the architecture of the network may change in the near future to even better serve the classification task.

Table 7: Question type confusion matrix – LSTM network

	expected									
predicted	AB	APHR	CL	D/T	ENT	LOC	NUM	OTH	PER	VPHR
ABBR.	7	0	0	0	2	0	0	0	0	0
ADJ_P.	0	12	0	0	9	2	0	0	1	0
CLAUSE	0	0	9	0	12	0	0	0	3	0
D/T	0	0	0	175	0	0	0	0	0	0
ENTITY	0	5	6	0	129	3	1	0	9	1
LOC.	0	1	0	0	7	141	0	0	1	0
NUM.	1	1	0	1	0	0	87	0	0	0
OTHER	0	0	0	0	1	0	0	0	0	1
PERSON	0	0	0	0	6	0	0	0	95	0
V_PHR.	0	0	0	0	1	0	0	1	0	89

Table 8: Answer type confusion matrix – LSTM network

	expected									
predicted	AB	D/T	ENT	LOC	NUM	ORG	OTH	PER	DEN	Y/N
ABBR.	7	0	0	0	0	1	1	0	0	0
D/T	0	175	0	0	0	0	0	0	0	0
ENTITY	0	1	72	4	0	2	13	3	1	0
LOC.	0	0	2	140	0	2	2	1	0	0
NUM.	1	1	0	0	87	0	1	0	0	0
ORG.	0	0	1	0	0	12	1	3	0	0
OTHER	0	3	20	2	1	0	44	7	2	2
PERSON	0	0	3	0	0	2	2	96	0	0
DENOT.	0	0	2	0	0	0	1	0	9	0
YES_NO	0	0	0	0	0	0	1	0	0	89

5 Conclusion and Future Work

In this paper, we have introduced two different tools for question and expected answer type detection used in the Question processor and Answer extraction

modules of the question answering system AQA – a rule based system and a Long Short-Term Memory (LSTM) network.

The detection of the rule based system is based on a set of hand-written rules which determine QA types according to lexical, syntactic and semantic features obtained by the question processing. The module was trained on a balanced half of the SQAD questions and evaluated with the testing set of comparatively the same size. The resulting precision was 88.77% for question and 85.05% for answer types with the respective recall of 87.79% and 84.52%. The combined overall F1 measure was 82.68%.

The LSTM network is machine learning based and utilises a recurrent neural network model using Fasttext word embedding vectors. The model has been trained on 50% of the SQAD questions while next 10% have been used for model evaluation during the training phase and 40% for testing. The combined recall of both types is 87.07% with the question and answer type precision going up to 91.59% and 89.76% respectively. The results show that the LSTM system outperforms the rule based model by 2.7-5 points in each score.

The introduced question and answer type detection tools have been developed in order to improve the performance of the question answering system AQA. Because of the fact machine learning based systems have better presumptions for the future, it is planed to continue in the development of the LSTM model, which includes experimenting with its architecture and setting of hyperparameters.

Acknowledgements. This work has been partly supported by the Czech Science Foundation under the project GA18-23891S.

References

1. Wang, W., Yan, M., Wu, C.: Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Volume 1. (2018) 1705–1714
2. Hu, M., Peng, Y., Huang, Z., Yang, N., Zhou, M., et al.: Read + Verify: Machine reading comprehension with unanswerable questions. arXiv preprint arXiv:1808.05759 (2018)
3. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100,000+ questions for machine comprehension of text. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Association for Computational Linguistics (2016) 2383–2392
4. Medveď, M., Horák, A.: AQA: Automatic Question Answering System for Czech. In Sojka, P., et al., eds.: Text, Speech, and Dialogue 19th International Conference, TSD 2016 Brno, Czech Republic, September 12–16, 2016 Proceedings, Switzerland, Springer International Publishing (2016) 270–278
5. Medveď, M., Horák, A.: Sentence and Word Embedding Employed in Open Question-Answering. In: Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018), Setúbal, Portugal, SCITEPRESS - Science and Technology Publications (2018) 486–492

6. Šulganová, T., Medveď, M., Horák, A.: Enlargement of the Czech Question-Answering Dataset to SQAD v2.0. In Aleš Horák, Pavel Rychlý, A.R., ed.: Proceedings of the Eleventh Workshop on Recent Advances in Slavonic Natural Language Processing, RASLAN 2017, Brno, Tribun EU (2017) 79–84
7. Przybyła, P.: Question Analysis for Polish Question Answering. In: 51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop, Association for Computational Linguistics (2013) 96–102
8. Laokulrat, N.: A survey on question classification techniques for question answering. KMITL Information Technology Journal **2**(1) (2013)
9. Silva, J., Coheur, L., Mendes, A.C., Wichert, A.: From symbolic to sub-symbolic information in question classification. Artificial Intelligence Review **35**(2) (2011) 137–154
10. Krishnan, V., Das, S., Chakrabarti, S.: Enhanced Answer Type Inference from Questions Using Sequential Models. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. HLT '05, Association for Computational Linguistics (2005) 315–322
11. Kwok, C., Etzioni, O., Weld, D.S.: Scaling question answering to the web. arXiv preprint arXiv:1808.05759 **19**(3) (2001) 2383–2392
12. Rambousek, A., Pala, K., Tukačová, S.: Overview and Future of Czech Wordnet. In McCrae, J.P., Bond, F., Buitelaar, P., Cimiano, P., 4, T.D., Gracia, J., Kernerman, I., Ponsoda, E.M., Ordan, N., Piasecki, M., eds.: LDK Workshops: OntoLex, TIAD and Challenges for Wordnets, Galway, Ireland, CEUR-WS.org (2017) 146–151
13. Šmerk, P.: Fast Morphological Analysis of Czech. In: Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2009. (2009) 13–16
14. Horák, A., Medveď, M.: SQAD: Simple Question Answering Database. In: Eighth Workshop on Recent Advances in Slavonic Natural Language Processing, RASLAN 2014, Brno, Tribun EU (2014) 121–128
15. Šulganová, T., Medveď, M., Horák, A.: Enlargement of the Czech Question-Answering Dataset to SQAD v2.0. In: Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2017. (2017) 79–84
16. Šmerk, Pavel: K počítačové morfologické analýze češtiny (in Czech, Towards Computational Morphological Analysis of Czech). PhD thesis, Faculty of Informatics, Masaryk University (2010)

Recurrent Networks in AQA Answer Selection

Radoslav Sabol, Marek Medveď, and Aleš Horák

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

Abstract. Unlimited, or open domain, question answering system AQA is being developed and tested with the Simple Question Answering Database (SQAD) for the Czech language. AQA is optimized for work with morphologically rich languages and makes use of syntactic cues provided by the morphosyntactic analysis.
In this paper, we introduce a new answer selection module being developed for the AQA system. The module is based on recurrent neural networks processing the question and answer sentences to derive the most probable answer sentence.
We present the details of the module architecture and offer a detailed evaluation of various hyperparameter setups. The module is trained and tested with 8,500 question-answer pairs using the SQAD v2.1 benchmark dataset.

Keywords: question answering; answer selection; QA dataset; SQAD; AQA

1 Introduction

Open-domain question answering techniques that look for the answer in unstructured textual data often start with identifying the most relevant parts of text, i.e. they select the relevant documents and/or relevant sentences. A success in this answer (sentence) selection procedure is a key predeterminer of the overall accuracy of the technique.

Specifically, given a question and a (large) set of candidate answer sentences (the unstructured textual knowledge base), the task lies in ordering the sentences by a score which reflects the probability of that the correct answer can be extracted from this particular sentence.

The early approaches to answer selection were based on direct exploitation of either syntactic features of the texts [1] or by identifying discourse entities and semantic relations to support the sentence selection process [2]. The recent results are mostly based on machine learning approaches. Starting with the bag-of-words models based on the Textual Entailment problem [3] up to the current state-of-the-art deep neural networks methods [4,5].

In the following text, a new answer selection module of the open domain question answering system AQA [6,5] is presented. Section 2 briefly recaps the

Aleš Horák, Pavel Rychlý, Adam Rambousek (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2018, pp. 53–62, 2018.

structure of the AQA system. In the next section, the architecture of the new answer selection module exploiting the recurrent neural network approach is detailed with a thorough evaluation in Section 4.

2 The Automatic Question Answering Tool

The Automatic Question Answering tool (AQA [6,5]) represents an open-domain QA system which concentrates on inflected languages that are guided by rich morphological and syntactic systems. AQA takes as a particular example the Czech language.

The AQA system architecture follows a pipeline model which consists of four main parts:

- question processor module
- document selection module
- answer selection module
- answer extraction module

After the input question is asked by a user, the AQA system preprocesses the question and extracts several pieces of information: the question type, the possible answer type, base forms of all words in the question, all phrases contained in the question, and the word tree distances. All these information are used in the following processing levels.

According to the extracted information, the next level selects the most probable document from the AQA textual knowledge base that should contain the answer to the input question. This extraction is based on TF-IDF scoring.

In the next step, the answer selection module goes through each sentence in the selected document and evaluates their similarity score with respect to the input question. This score is computed from multiple similarity features that are implemented in the system such as the tree distance, entity match, or phrase similarity based on word embeddings.

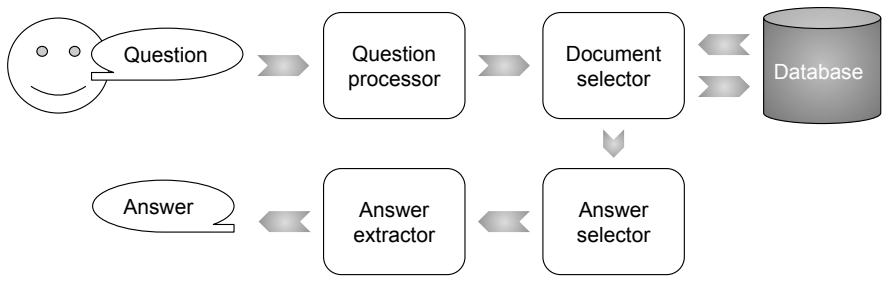


Fig. 1: AQA system work-flow visualization

The last module of answer extraction takes the most probable answer sentence and finds the shortest answer as a sub-phrase of the sentence and provides it as the final answer to the user.

The schema of the AQA system is presented in Figure 1.

The new module introduced in this paper will be employed in the answer selection level as a one of the features that create the scores of candidate sentences.

3 The Answer Selection Architecture

In this section, the architecture of the new answer selection module is detailed. The current implementation is based on the recurrent neural networks (RNN) approach. The general schema takes inspiration from [8], where the authors have introduced a general network schema which jointly learns a similarity measure of two parallel inputs. The schema can be applied to different neural network types, in [8] the convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are evaluated.

The specific RNN architecture of the new answer selection module is presented in Figure 2. Given the question q and pool of candidate answers A, the goal of this network is to rank each input pair with a similarity score $s_\theta(q, a), \forall a \in A$, where the pair with highest ranking is the most probable to be the correct answer.

As an input, the neural network model receives two sequences of word embedding vectors representing the words of the question $q = (q_1, ..., q_L), q_i \in \mathbb{R}^E$ and the words of the candidate answer $a = (a_1, ..., a_M), a_j \in \mathbb{R}^E$ with E being the word embedding dimension. In the first step, the word embedding vectors are independently passed through a bidirectional Gated Recurrent Unit (Bi-GRU) layer consisting of one forward and one backward oriented layer. The Bi-GRU layer creates new hidden representations $Q \in \mathbb{R}^{H \times L}$ and $A \in \mathbb{R}^{H \times M}$ (where H is the hidden output dimension of the Bi-GRU layer), adding contextual information about each token into the new matrix.

The hidden representations of the question and the candidate answer are then combined in the matrix $G \in \mathbb{R}^{L \times M}$ computed as follows:

$$G = \tanh(Q^\intercal \cdot W \cdot A) \tag{1}$$

The central matrix $W \in \mathbb{R}^{H \times H}$ contains learnable weights connecting all elements of the Q and A matrices. The resulting matrix G thus contains soft alignment scores between each token of the question and the answer Bi-GRU outputs. As a projection of this combined information back to the original question and the original candidate answer, the column-wise and row-wise max-pooling followed by the softmax non-linearity is applied to G to obtain attention vectors $g_q \in \mathbb{R}^L$ and $g_a \in \mathbb{R}^M$. The content of g_q can be interpreted as an importance score for each word in q with regard to the candidate answer a and vice versa for g_a.

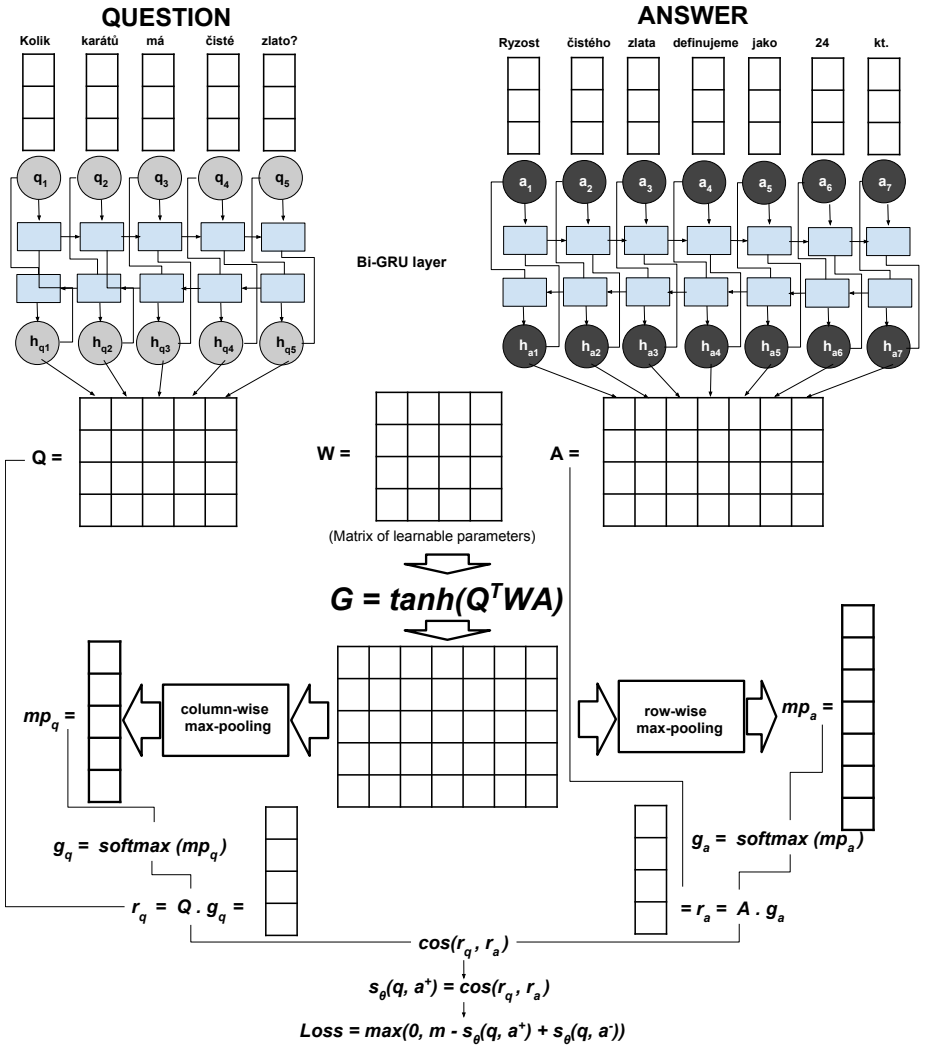
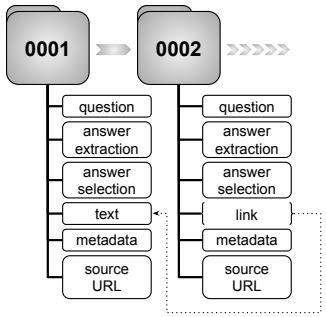


Fig. 2: The RNN architecture of the new answer selection module.

The final representations used for the similarity score computation, $r_q \in \mathbb{R}^H$ and $r_a \in \mathbb{R}^H$ are derived as matrix product of the Bi-GRU hidden outputs Q and A with the corresponding attention vectors. The vectors r_q and r_a thus contain the results of the hidden word representations of the question and the candidate answer weighted by the attentive scores of their mutual relationship:

$$\begin{aligned} r_q &= Q \cdot g_q \\ r_a &= A \cdot g_a \end{aligned} \tag{2}$$



No. of records	8,566
No. of questions	8,566
No. of answer selection	8,566
No. of answer extractions	8,566
No. of unique documents (know.base)	5,417
No. of sentences in KB	433,495
No. of tokens in KB	9,615,966
No. of tokens in questions	69,016
No. of tokens in answer selection	201,069
No. of tokens in answer extraction	19,752

Fig. 3: The SQAD v2.1 file organization and statistics.

The final score of the input pair (q, a) is obtained using the cosine similarity between r_q and q_a.

4 Evaluation

4.1 The Dataset Characteristics

The new answer selection module has been evaluated with the SQAD v2.1 dataset [7]. SQAD is an open source Czech question answering dataset consisting of more than 8,500 question-answer pairs[1] enriched by manually added metadata such as:

- question/answer type labels
- exact answer (answer extraction)
- answer sentence (answer selection)

The exact answer (or answer extraction) is formed by a sub-phrase of the answer selection text. The answer selection sentence comes from identified knowledge base document, but the sentence is too broad to answer the question. Therefore, the answer extraction process is triggered to find the smallest part that can be used as the correct answer to the given question. The database consists of 10 categories of question types and 10 categories of answer types. The distribution matrix of these types is present in Table 1.

To evaluate the module introduced in this paper, the module is trained on a subset of the SQAD database with taking the question and randomly selected sentences from the question source document as negative candidate answers and the answer sentence as the positive candidate answer.

For evaluation purposes the SQAD database has been divided into three parts: training (50% of the dataset), validation (10% of the dataset), and testing

[1] See Figure 3 for the schema of the database content and the current statistics of the SQAD database.

Table 1: SQAD v2.1 distribution matrix of question and answer types

A type / Q type	PER.	DENOT.	ENT.	OTHER	ORG.	DATE /TIME	LOC.	NUM.	ABB.	YES /NO
PERSON	1,016	0	2	0	3	0	2	0	0	0
ENTITY	20	101	1,031	378	204	1	7	1	2	0
ADJ_P.	7	0	8	216	0	0	0	2	0	0
DT./TM.	0	0	1	2	0	1,844	0	4	0	0
LOC.	1	0	14	5	3	0	1,501	0	0	0
NUM.	1	0	0	2	0	0	0	910	0	0
ABBR.	0	1	0	0	0	0	0	0	80	0
CLAUSE	1	0	27	205	6	0	1	1	0	0
VERB_P.	2	0	1	0	0	0	0	0	0	937
OTHER	2	0	1	11	0	0	0	0	0	1

(the remaining 40%). These part are balanced across question/answer type tuples to achieve correct model training and evaluation.

The input word embedding vectors for the question and the candidate answers have been pre-computed by fastText [9] which was trained with large Czech corpus of cleaned web texts (csTenTen17, a corpus of 10 billion words [10]).

4.2 Neural Network Configuration and Training

The Bi-GRU architecture of the new answer selection module was trained and evaluated with the SQAD v2.1 dataset divided into the three subsets – the training set, the validation set, and the testing set. The training set with 4271 question-answer pairs is used to learn the weights and the biases of the model. The validation set of 889 questions provides an unbiased evaluation of the current model parameters after each training epoch. The testing set of 3406 QA pairs is used to evaluate the model after the training is complete.

For each learning epoch, the training set data is shuffled in random order. Each drawn question and the positive answer are randomly supplemented with 50 negative answers sampled from the same text document in the dataset. The input question and each candidate answer are converted to a list of 100-dimensional word2vec embedding vectors. Before each step of the training process, a specific dropout rate is applied on the network input layer.

The final training objective of the answer selection network is defined as a hinge loss [11,12]:

$$Loss = \max\{0,\, m - s_\theta(q, a^+) + s_\theta(q, a^-)\}, \tag{3}$$

where m is a constant margin (0.2 is used as suggested in [8]), s_θ is the cosine similarity as computed by the network with parameters θ, q is the input question and a^+/a^- are the positive/negative answers.

a) one model b) two models at the same time

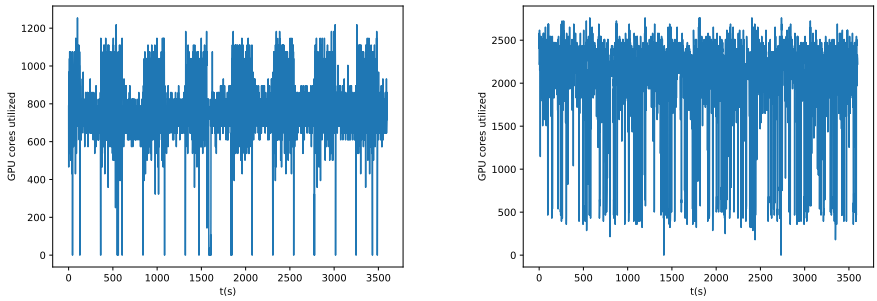


Fig. 4: GPU core utilization for one hour of training. Maximum number of working GPU cores is 3584.

The Stochastic Gradient Descent (SGD) with adaptive learning rate is used as an optimizer. Instead of fixed learning rate, the learning rate λ_t is updated in each epoch t as follows [13]:

$$\lambda_t = \frac{\lambda}{t}, \tag{4}$$

where λ is initial learning rate. The *Loss* is computed for each input of (q, a^*) related to the sampled question, but the network weights are updated only once using the negative answer with the highest score.

The model is trained on 25 epochs, the input data is loaded using multiple workers (on CPU), the training process itself exploits an NVIDIA GeForce GTX 1080 Ti GPU. Figure 4 displays the GPU core utilization for 8 epochs that were executed within 1 hour. The GPU usage noticeably decreases during validation – the reason behind this is the fact that the backpropagation learning algorithm is more demanding than the computations when passing through the validation set. The GPU utilization was 22% on average for training one model, and it raised to 58% (Figure 4 b)) when training 2 models at the same time. The running times for one training epoch were approximately 380 seconds for iterating through the training set, and 260 seconds for the validation set. Although the validation set is smaller in size, all possible answers (sentences in the related document) need to pass through the network for validation, while in the training set only a random sample of 50 candidate answers are used.

4.3 The Results

In this section, the results of models trained with different hyperparameters are presented as shown in Table 2. 27 different models were trained, using 3 values for each of the 3 most influential hyperparameters – the output dimension, the dropout rate, and the initial learning rate.

The output dimension H is the size of the output from the Bi-GRU layer. Note that $H = 2 \cdot h$, where h is the dimension of the hidden vectors of the forward

Table 2: The results for combinations of hyperparameter values

Output size	Dropout	Learning rate λ	Training set (in %)	Validation set (in %)	Test set (in %)
240	0.2	0.05	71.25	54.44	61.77
		0.1	80.05	58.04	65.85
		0.2	85.85	60.29	68.26
	0.5	0.05	67.92	51.74	57.55
		0.1	71.01	53.20	59.95
		0.2	77.68	55.23	63.76
	0.7	0.05	62.83	49.67	54.9
		0.1	47.92	43.31	42.31
		0.2	42.85	36.78	37.17
260	**0.2**	0.05	73.84	54.22	61.89
		0.1	80.84	58.38	65.47
		0.2	**86.09**	61.08	**68.29**
	0.5	0.05	68.15	51.52	58.49
		0.1	69.60	53.99	58.95
		0.2	77.49	55.01	61.13
	0.7	0.05	61.77	50.96	54.9
		0.1	48.74	43.76	43.1
		0.2	44.36	38.69	38.63
280	0.2	0.05	74.59	55.46	62.92
		0.1	80.68	58.72	65.77
		0.2	83.25	**61.52**	68.13
	0.5	0.05	68.83	51.29	58.75
		0.1	69.55	54.11	58.19
		0.2	80.37	56.46	66.18
	0.7	0.05	63.06	50.61	54.67
		0.1	48.68	42.74	42.45
		0.2	43.93	38.92	37.93

or backward GRU layers. These vectors are concatenated to form the output vectors $h_{q_1}, ..., h_{q_L}$ as the columns of the output matrix Q. The output dimension of $H = 260$ has produced the best results, but the other output dimension values of $H = 240$ and $H = 280$ did not substantially degrade the accuracy.

As for the dropout rate, the best results were produced by using the values of 0.2 and 0.5, while 0.7 has decreased the accuracy considerably.

The initial learning rate affects the accuracy in conjunction with the dropout rate: for the dropout of 0.2 and 0.5 increasing the initial learning rate improves the results considerably. On the other hand, the dropout value of 0.7 drops the accuracy by a huge amount independently on the initial learning rate.

For this experiment, the best combination of hyperparameters has been able to find the correct answer in **68.29%**. For a comparison with the previous answer selection module, a different data setup was also evaluated. As the previous module was tested with SQAD v1.0 only, the new module has been here trained with 5265 question-answer pairs from SQAD v2.1 not present in v1.0 and then

tested on the same 3301 QA pairs as the previous module. The new module has reached the accuracy of **66.03%**, which is an improvement of 9.53%.

5 Conclusions and Future Work

We have presented the results of an implementation of a new answer selection module based on the recurrent neural networks approach. The model was trained and evaluated using the SQAD v2.1 question-answering benchmark dataset that consists of 8566 question answer pairs with detailed structured information.

We have provided a thorough evaluation of possible settings of the module hyperparameters with the best attained accuracy of 68.29% when trained on 50% of the dataset and evaluated with 40%, i.e. 3406 questions. In comparison with the previous implementation, the module has reached the accuracy of 66.03% with the SQAD v1.0 data, achieving an increase of almost 10%.

In the next step, we plan to evaluate the whole AQA pipeline accuracy using the newly implemented modules for answer selection and question-answer type detection with the SQAD v2.1 dataset. In an experimental setup, the RNN module will be also tested with sub-sentence phrases instead of full sentences, which would also allow to improve the answer extraction process.

Acknowledgements. This work has been partly supported by the Czech Science Foundation under the project GA18-23891S.

References

1. Litkowski, K.C.: Question-answering using semantic relation triples. In: Proc. of the 8th Text Retrieval Conference (TREC-8). (1999) 349–356
2. Katz, B., Lin, J.: Selectively using relations to improve precision in question answering. In: Proceedings of the workshop on Natural Language Processing for Question Answering (EACL 2003). (2003) 43–50
3. Jijkoun, V., de Rijke, M., et al.: Recognizing textual entailment using lexical similarity. In: Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, Citeseer (2005) 73–76
4. Tay, Y., Tuan, L.A., Hui, S.C.: Multi-cast attention networks for retrieval-based question answering and response prediction (2018)
5. Madabushi, H.T., Lee, M., Barnden, J.: Integrating question classification and deep learning for improved answer selection. In: Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics (2018) 3283–3294
6. Medveď, M., Horák, A.: AQA: Automatic Question Answering System for Czech. In: International Conference on Text, Speech, and Dialogue, TSD 2016, Springer (2016) 270–278
7. Medveď, M., Horák, A.: Sentence and word embedding employed in open question-answering. In: Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018), Setúbal, Portugal, SCITEPRESS - Science and Technology Publications (2018) 486–492

8. dos Santos, C.N., Tan, M., Xiang, B., Zhou, B.: Attentive pooling networks. CoRR **abs/1602.03609** (2016)
9. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. CoRR **abs/1607.04606** (2016)
10. Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlỳ, P., Suchomel, V.: The TenTen corpus family. In: 7th International Corpus Linguistics Conference CL2013. (2013) 125–127
11. Weston, J., Chopra, S., Adams, K.: # tagspace: Semantic embeddings from hashtags. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). (2014) 1822–1827
12. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional neural network architectures for matching natural language sentences. In: Advances in neural information processing systems. (2014) 2042–2050
13. Dos Santos, C.N., Zadrozny, B.: Learning character-level representations for part-of-speech tagging. In: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32. ICML'14, JMLR.org (2014) II–1818–II–1826

Automatically Created Noun Definitions for Czech

Marie Stará

Faculty of Arts, Masaryk University
Arna Nováka 1, 602 00 Brno, Czech Republic
413827@mail.muni.cz

Abstract. This paper comments on the automatically created noun definitions. The definitions were created using the results of the Sketch Grammar developed with the aim to gather data for them. These data (Word Sketches) are combined using Python script to form the definition.

Keywords: dictionary definition; corpora; word sketch

1 Introduction

When a monolingual dictionary is created, one of the most complicated tasks to be dealt with is the creation of definitions. Though corpora are used in lexicography since the 1980s as a source of examples (and there were efforts to automatically find definitions), there has been (at least to my knowledge) no attempt to create a tool that would make it possible to *create* definitions automatically.

The purpose of this paper is to show automatically created definitions of Czech nouns and evaluate whether they can be used in a dictionary as they are, or if they can only serve as a basis for human-made definition.

2 Construction of definitions

As is stated below, it is not possible to create such definitions as those we can find in human-made dictionaries. I am using the word definition even though it is more of a set of hints for understanding a word. The definitions are created by composing Word Sketches of the given word together; I am using existing Word Sketches and adapting them to suit the needs of the definition creation, thus making my own Sketch Grammar. This Sketch Grammar was used with the 5-billion-token czTenTen12 corpus.

To construct a definition, I download Word Sketches of the given word in JSON format and use a script in Python to form the pieces of information together. The script takes first three words with highes score for each relation and merges them together in groupes described below.

Each definition consists of several parts, each of which is formed by one or more Word Sketch relation.

Aleš Horák, Pavel Rychlý, Adam Rambousek (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2018, pp. 63–68, 2018.

The most common definitions of nouns consist of genus proximum (hypernym of the given word) and differentia specifica (what distinguishes the word from its synonyms). [1,2] I am using the hypernym relation as well as the relation of synonymy; since it is not possible to reliably establish the hypernymy and/or synonymy for every word in the corpus, I am using the relation of very loose synonymy in my definitions. It is formed by combining results of Word Sketches *coord* (coordination; searching for words connected either by the conjunctions *a* (and) or *nebo* (or), or *ani–ani* (nor–nor) or *buď–nebo* (either–or), *a_jine* (and similar), *například* (for example), and *hypo_hypero*[1] (hyponymy—hypernymy). That means almost every noun is allocated with one or more words of similar meaning.

Below, the relevant part of definition of *pes* (dog) is showed.

> *podobný význam má zvíře, mazlíček, zvířectvo, dítě, člověk, plemeno, jezevčík*
> (similar meaning has animal, pet, fauna, child, human, breed, Dachshund)

Quite similar is the combination of *adj_modif* (adjective modifiers) of the given word and *slovo_je* (the word is).

> *pes může být hlídací, zakopaný, lovecký, pes, zvíře, přítel*
> (a dog can be a watchdog, burried, hunting, a dog, an animal, a friend)

Different but still quite frequent in definitions is the “part of" relation (partitive) [2]. For finding parts I'm using the *slovo_má* (a word has) and *skládá_se_z* (is consisting of) relations. Related is the *skládá_se_z_2* (is consisting of 2) relation which finds the words that consist of the given word; the skládá_se_z and skládá_se_z_2 relations are symmetrical. Similar to skládá_se_z_2 is the *kdo_co_má_slovo* (who/what has a word) relation; both relations should, optimally, find holonyms of the given word.

> *pes může mít pes, srst, vodítko*
> (the dog can have a dog, a hair, a string)
> *pes, majitel, soused může mít pes [psa]*
> (a dog, an owner, a neighbor can have a dog)

Another piece of the definition consists of verbs to which the defined word is either a subject(*je_podmět*) or an object in accusative (*je_předmět_4*) or instrumental (*je_předmět_7*). The valency is important for the definition, as it is a stable pattern of usage and therefore helps us understand the meaning of the unknown word. [4]

It could be argued that using only accusative and instrumental is not enough and that the genitive and dative forms should be used as well. There are two reasons for excluding them. Firstly, the genitive and dative objects have a lower frequency than the accusative and instrumental ones. Secondly, the cases are

[1] This relation was introduced by Baisa and Suchomel in [3].

often not appropriately recognised by the tagger, probably because of many cases of case syncretism in Czech.[2]

pes (se) může štěkat, štěknout, chcípnout; je možné jej/ji venčit, vyvenčit, pořídit a jím/jí vrtět, nakrýt, venčit
(dog can bark, make one bark, die; it can be being walked, walked, acquired; you can wag it, it can be used at stud (breeded))

There are two more relations I use in the definitions. These are *gen* (genitive following the given noun) and *instr* (instrumental following the given noun). It is useful only in some cases (mostly due to many wrong PoS tags), but I aim for good recall more than for good precision. (The reason is that I try to find definitions even for words with low frequency, which results in a lot of garbage data with the frequented ones.)

pes čeho: stář[3]*, demokracie, plemeno*
(dog of an age, a democracy, a breed)
pes (s) kým/čím: mikročip, povel, psovod
(dog with a microchip, a command, a dog handler)

3 Evaluation of definitions

I evaluated the definitions on a set of 78 nouns. The words were chosen based on various criteria. The most apparent criterion is frequency: words with both high and low frequency are included. There are words which seem to be easy to define (e.g. *pes – zvíře, které štěká*, dog – an animal which barks) and those which are harder to explain. The complexity of the explanation is connected to whether the word being defined is an abstractum or a concretum (abstract words being more complicated to define). In the set, there are words with one and more meanings. There are also synonyms included as well as words creating a scale – diminutives and augmentatives. Some words were picked ad hoc to ensure the test set is differentiated enough.

3.1 Examples

There are few words, for which the Word Sketches do not yield sufficient data. *Cestující* (someone who is travelling) is not recognised as a noun, but only as an adjective, therefore it does not contain data for the definition. *Barabizna* (augmentative expression for a house) is assigned only adjective modifiers and verbs to which it is either subject or object, due to its low frequency. Some other words with low frequency are not possible to define using my approach, for example, *barik* (oak [barrel for winemaking]) or *exposé* (an expose), the only word in the set for which I found only irrelevant data.

[2] *nominative–accusative*: inanimate masculine in both singular and plural; plural of feminine and neuter
genitive–accusative: singular of animate masculine
dative–locative: plural of masculine (both animate and inanimate), feminine, neuter

[3] this is an example of wrong lemma, it should be spelled as *stáří*

barik
podobný význam má sičák[4]
(similar meaning has a crook)
barik může být zatříděný, ovocitý, zapracovaný, víno
(oak can be classed, fruity, strong, wine)
barik (se) může ochutnat, potlačit, slušet; jím/jí překvapit, ovlivnit
(oak can be tasted, suppressed, matching; you can surprise with it, you can influence it)
barik čeho: aroma, vůně, chuť
(oak of aroma, smell, taste)
exposé
podobný význam má spacesa, spaces, dashboard, najetí, program
(similar meaning has spaces, dashboard, pointing (the cursor), programme)
exposé může být chvaličský, peterssonův, a-l, ročenka, kratochvíle, inovace
(expose can be subservient, peterssons, a-l, a yearbook, an amusement, an innovation)
exposé může mít svazek, roh, omezení
(exposé can have ligament, horn/corner, restriction)
stejskal, senzor, kláves[5] *může mít exposé*
(stejskal (a surname), a sensor, a key(board) can have an expose)
exposé (se) může namapovat, ozřejmit, sjednocovat; je možné jej/ji salariésit, namapovat, chromat a jím/jí napodobit, narušit, zobrazit
(expose can be mapped, explained, unified; it can be *salariesed*, mapped, *chromed*; you can imitate, disturb, show it)
exposé čeho: přescent[6]*, eleanora, zaorálek*
(expose of eleanor, zaorálek (a surname of a politician)
exposé (s) kým/čím: kinematografie, líčení
(expose with a cinematography, make-up)

On the other hand, the meaning of *trdliště* (fish breeding ground) can be deduced from the automatically created definition, even though it is a very uncommon expression.

trdliště
podobný význam má zimoviště, jikra, úkryt, chvojí
(similar meaning has winter quarters, fish egg, hiding, branches)
trdliště může být lipaní, vysbírán, lososí, pach, lov, samice
(it can be of graylings, picked up, of salmons, smell, hunt, female
trdliště může mít orlice, průměr
(it can have eagle, diameter)
makrela, ryba, populace může mít trdliště
(mackarel, fish, population can have fish breeding ground)

[4] misspelled in corpus, should be syčák
[5] misspelled, should be klávesa
[6] a word play

trdliště (se) může vlákat, spásat, hloubit; je možné jej/ji vybagrovat, devastovat, poničit
(it can allure sth, be grazed, be deepened; it can be excaveted, devastated, destroyed)
trdliště čeho: bistrino, lipan, losos
(fish breeding ground of bistrino (a name), a grayling, a salmon)
trdliště (s) kým/čím: sediment, peřej, většina
(fish breeding ground with a sediment, chute, majority)

Another problematic group is abstract expressions, e.g. *vypjatost* (the "being extreme" property), *dobro* (well-being, the good), *léto* (summer, year). On the other hand, other abstract expressions have more or less acceptible definition, for example *nenávist* (hatred) or *nicota* (nothingness).

3.2 Evaluation

The most reliable data result from valency followed by the loose synonymy and word-is relation. The least reliable are the instrumental and genitive of the given word with only 14 and 34 good results, respectively.

The partitive relations (holonymy and meronymy) [5] has comparatively better results for finding parts of the given word than for finding its holonyms. (This difference might be caused by the test set.)

One of the reasons the definitions are not good enough to be used without any editing is considerably high frequency of wrongly identified lemmata (typically recognizing adjectives as a 3rd person of a verb, e.g. *(mainská) mývalí kočka* (Maine Coon), where the adjective *mývalí* (racoon-like) is identified as a verb ([a cat is] racooning). Another reason is the above-mentioned wrong case identification.

It is worth noting that many of the ill-defined words are not included in the most up-to-date Czech monolingual dictionary [6]. Some of the definitions presented there are, moreover, hard to decipher even for native speakers.[7]

All in all, the definitions are not good enough to be presented in a dictionary without any editing. Nevertheless, they could be very well used as a basis for forming new user-friendly definitions.

4 Conclusion

With the corpus data containing mistakes in lemmata as well as tags, it is nearly impossible to automatically create definitions which would not need any editing. It is, however, possible to make a good basis for lexicographers to work on. This approach could be used in other languages significantly simplifying the process of dictionary.

[7] I asked a non-native speaker with C1 level Czech, and he could not understand about 20 % of the presented definitions. I would argue that monolingual dictionary that does not explain is not very good.

Table 1: Percentage of good/bad/no results for each group of relations

	good results (%)	bad results (%)	no result (%)
similar meaning has (the loose synonymy)	91.03	6.41	2.56
word can be	93.59	6.41	0.00
word can have (meronymy)	66.66	23.08	10.26
sth can have *word* (holonymy)	50.00	33.33	16.67
valency	92.31	5.13	2.56
genitive	43.59	50.00	6.41
instrumental	17.95	60.26	21.79

Acknowledgements. This work has been partly supported by the project of specific research Čeština v jednotě synchronie a diachronie (Czech language in unity of synchrony and diachrony), MUNI/A/0862/2017.

References

1. Atkins, S., Rundell, M. The Oxfor Guide to Practical Lexicography. Oxford University Press, New York (2008)
2. Landau, S.: Dictionaries: the art and craft of lexicography. Cambridge University Press, Cambridge (2001)
3. Baisa, V., Suchomel, B.: Corpus Based Extraction of Hypernyms. [online] (accessed August 29, 2018)
4. Hanks, P.: How people use words to make meanings: Semantic types meet valencies. [online] (accessed October 14, 2018)
5. Hladká, Z., Dočekal, M.: MERONYMNĚ-HOLONYMNÍ VZTAH. [online] (accessed November 9, 2018)
6. Filipec, J.: Slovník spisovné češtiny pro školu a veřejnost. 4th edn. Academia, Praha (2005)

Part III

NLP Applications

Multiple Instance Terminological Thesaurus with Central Management

Adam Rambousek

Natural Language Processing Centre
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00, Brno, Czech republic
rambousek@fi.muni.cz

Abstract. This paper describes the design of the new specialized dictionary writing system for the creation and management of terminological thesaurus. To help with information sharing and terminology unification, the system also includes central node that keeps track of all the dictionary instances and synchronize data between them.

Keywords: terminology thesaurus; dictionary writing; DEB platform; Linked Data

1 Introduction

Specialists in any branch inevitably rely on domain-specific vocabulary as a basis for sharing exact terminology amongst professionals. Such detailed domain terminology cannot be included in general language dictionaries, which is why specialized terminology dictionaries are being built and managed. With the need to share information unambiguously in different languages, terminology dictionaries often link original terms to their translations. Taxonomic ordering of the terminology is described by means of term relations such as synonymy or hypernymy/hyponymy. In our new system, information about the terms is presented and visualized in a way that helps the readers (both specialists and the general public) to understand the meaning of the term and its usage in contexts.

As a pilot project, The Natural Language Processing Centre (NLP Centre) at the Faculty of Informatics, Masaryk University in cooperation with the Czech Office for Surveying, Mapping and Cadastre (CUZK) has developed a new system for building and extending a specialized terminology thesaurus for the domain of land surveying and land cadastre. The project consists of several tightly interconnected parts—a web-based application to create, edit, browse and visualize the terminology thesaurus, and a set of tools to build large corpora of domain oriented documents which allows for the detection of newly emerging terms, or terms missing from the thesaurus.

In the follow-up project, the developed application for creation and editing of terminology thesaurus will be updated to be generally usable for any domain. Thus any organization may re-use the same system for terminology dictionary.

Aleš Horák, Pavel Rychlý, Adam Rambousek (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2018, pp. 71–76, 2018.

However, with several applications running in daily use, same term may appear in various thesauri. Also, users might want to inter-link connected terms between dictionaries. To handle the management of thesaurus instances and links between them, new central management system is in development.

1.1 The DEB platform

Both the thesaurus application and the central management system are developed using the universal dictionary writing system developed at the NLP Centre (Faculty of Informatics, Masaryk University). The system is called Dictionary Editor and Browser, or the DEB platform [3,4]. Since 2005, the DEB platform was applied in more than 10 large international research projects. Large-scale applications based on the DEB platform include the lexicographic workstation for the development of the Czech Lexical Database [2] with detailed morpho-syntactic information on more than 213 thousands Czech words, or the complex lexical database Cornetto combining the Dutch wordnet, an ontology, and an elaborate lexicon [5]. Currently ongoing projects include Pattern Dictionary of English Verbs tightly interlinked with the corpus evidence [6], Family names in Britain and Ireland [1] providing detailed investigations for over 45,000 surnames to be published by Oxford University Press, or the dictionary of the Czech Sign Language[1] with an extensive use of video recordings to present the signs [7].

The DEB platform is based on the client-server architecture, which brings along a lot of benefits. All the dictionary and interlinked data are stored on a server and a considerable part of the functionality is also implemented on the server-side, consequently the client application can be very lightweight. This approach provides very good tools for editor team cooperation; data modifications are immediately seen by all involved users. The DEB server also provides authentication and authorization tools.

2 Central management system

2.1 Thesaurus management

Central node keeps track of all the installed instances of thesaurus system. After the installation, local administrator of the thesaurus system fills in the metadata. Following details are stored at the central node:

- thesaurus instance ID,
- URL to access the data,
- name of the organization running this thesaurus,
- administrative contact,
- domain and content description.

[1] `http://www.dictio.info`

Metadata are sent to the central node, where they are stored as unverified data. New thesaurus instance is now available for search and inter-linking, although with the warning about unverified status. After the central node administrator contacts local organization and checks the details, new thesaurus instance may be switched to verified.

After registration, central system will start with periodical downloading of the thesaurus content (term list and entry details). This data serve as a back-up copy of each thesaurus and also are used for term linking and cross-reference checks, as described below.

2.2 Inter-linking terms

When the user of thesaurus application is adding new term, the same term is also checked in all registered instances. Request is sent to the central system and all the downloaded back-ups are queried for the new term. Consequently, central node returns the list of all instances that contain the entry for the given term.

After consulting the list, user may decide to copy details of one of the existing term entries. In such case, local thesaurus instance will request term entry details directly from the remote instance. Entry details are copied to the new entry, together with the link to source term entry. See Figure 1 with the chart explaining the process.

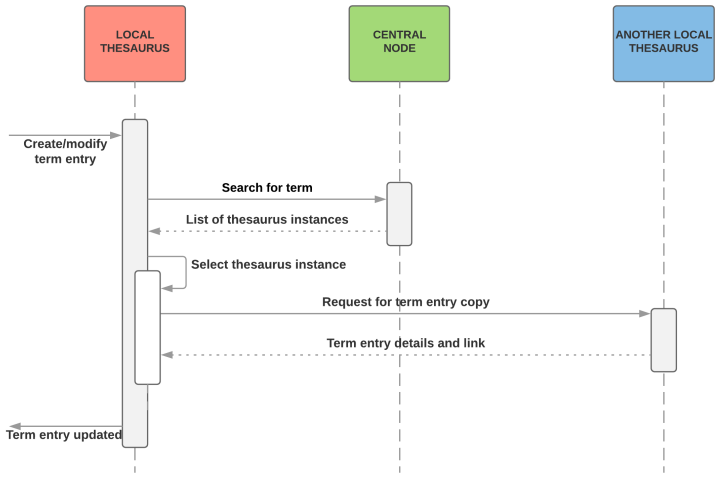


Fig. 1: Process of inter-linking thesaurus instances when creating new term entry.

2.3 Cross-reference checks

During the life cycle of various terminology thesaurus, existing entries are often updated, merged or split. These changes may also break the links between entries. For this reason, central management system periodically checks links between all the term entries, both in the same thesaurus instance, or in different instances.

If the central system detects updates in the link destination, editor-in-chief of the originating thesaurus instance is notified about the change. Consequently, editors will decide about the best action needed to keep term entries synchronized. See Figure 2 for the example of term entry containing link to external thesaurus instance.

```
<entry id="3611">
  <terms>
    <term lang="cs">stavba</term>
    <term lang="en">building</term>
  </terms>
  <refs>
    <ref type="external" system_id="https://terminologie.mvcr.cz"
      entry_id="895">stavba (Geoinfostrategie)</ref>
  </refs>
</entry>
```

Fig. 2: Term entry with link to entry in external thesaurus instance.

2.4 Official reference checks

Many terminology dictionaries are mentioned as the reference data in various official documents (laws, standards, regulations), or are derived from the official documents, e.g. term meaning is defined by the law. To support this kind of link, the thesaurus system provides special format of cross-reference links to official documents. Source data for the documents will be provided by the e-government office of the Ministry of the Interior and the cross-reference format was consulted to conform to future specification.

If the central system is notified by the external service that some official document was updated, all the entries in each thesaurus instance are checked. When an entry is found linking to the given document, editor-in-chief is notified and decides the best action to keep term entries in line with the official reference document. See Figure 3 for the example of term entry linking to the law where the term is defined.

```
<entry id="3761">
  <terms>
    <term lang="cs">stavební pozemek</term>
    <term lang="en">building site</term>
  </terms>
  <refs>
    <ref type="law" nr="183" year="2006" source="Sb">
      zákon č. 183/2006 Sb., o územním plánování a stavebním řádu (stavební zákon)
    </ref>
  </refs>
</entry>
```

Fig. 3: Term entry with link to the law where the term is defined.

2.5 Appearing new terms

As mentioned before, when the users create new term entry, they are provided with the list of thesaurus instances where the same term is existing. However, it may also happen that the same term appears in one of the thesaurus instances at later point.

To detect such case, central system is also periodically checking newly created term entries. If a new entry appears with the same term that is already existing, editors of all affected thesaurus instances are notified and asked to synchronize the term entries.

3 Conclusion

We have described enhancement of the lexicographic system for building and editing terminology thesaurus. The goal of the project currently in development is to inter-connect many thesaurus instances to the central management system. This organization will help to keep terminology synchronized between various domains and also in reference to the official government documents.

Acknowledgements. This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin OP VVV project CZ.02.1.01/0.0/0.0/16_013/0001781.

References

1. Hanks, P., Coates, R., McClure, P.: Methods for Studying the Origins and History of Family Names in Britain. In: Facts and Findings on Personal Names: Some European Examples. pp. 37–58. Acta Academiae Regiae Scientiarum Upsaliensis, Uppsala (2011)

2. Horák, A., Rambousek, A.: PRALED – A New Kind of Lexicographic Workstation. In: Przepiórkowski, A., Piasecki, M., Jassem, K., Fuglewicz, P. (eds.) Computational Linguistics: Applications, pp. 131–141. Springer (2013)
3. Horák, A., Rambousek, A.: DEB Platform Deployment – Current Applications. In: RASLAN 2007: Recent Advances in Slavonic Natural Language Processing. pp. 3–11. Masaryk University, Brno, Czech Republic (2007)
4. Horák, A., Rambousek, A.: Using DEB Services for Knowledge Representation within the KYOTO Project. In: Principles, Construction and Application of Multilingual WordNets, Proceedings of the Fifth Global WordNet Conference. pp. 165–170. Narosa Publishing House, New Delhi, India (2010)
5. Horák, A., Vossen, P., Rambousek, A.: A Distributed Database System for Developing Ontological and Lexical Resources in Harmony. In: Lecture Notes in Computer Science: Computational Linguistics and Intelligent Text Processing. pp. 1–15. Springer-Verlag, Haifa, Israel (2008)
6. Maarouf, I.E., Bradbury, J., Baisa, V., Hanks, P.: Disambiguating verbs by collocation: Corpus lexicography meets natural language processing. In: Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14). European Language Resources Association (ELRA), Reykjavik, Iceland (may 2014)
7. Rambousek, A., Horák, A.: Management and Publishing of Multimedia Dictionary of the Czech Sign Language. In: Biemann, C., Handschuh, S., Freitas, A., Meziane, F., Métais, E. (eds.) Natural Language Processing and Information Systems, NLDB 2015. pp. 399–403. Lecture Notes in Computer Science, Springer (2015). https://doi.org/10.1007/978-3-319-19581-0_37

Detection of Abusive Speech for Mixed Sociolects of Russian and Ukrainian Languages

Bohdan Andrusyak[1], Mykhailo Rimel[2], and Roman Kern[1]

[1] Know Center,
Inffeldgasse 13 8010 Graz AUSTRIA
badnrusyak@know-center.at
rkern@know-center.at
know-center.at
[2] Grenoble INP
46 Avenue Félix Viallet 38031 Grenoble FRANCE
Mykhailo.Rimel@grenoble-inp.org
grenoble-inp.fr

Abstract. Uncontrolled use of abusive language is a problem in modern society. Development of automatic tools for detecting abusive and hate speech has been an active research topic in the past decade. However, very little research has been done on this topic for Russian and Ukrainian languages. To our best knowledge, no research considered surzhyk [3]. We propose to use unsupervised probabilistic technique with a seed dictionary for detecting abusive comments in social media in Russian and Ukrainian languages. We demonstrate that this approach is feasible and is able to detect abusive terms that are not present in the seed dictionary.

Keywords: Russian; Ukrainian; abusive speech

1 Introduction

Abusive language, including different swear words and hate speech is not a new phenomenon on the Internet. There are numerous reasons why such language might be considered undesirable, including but not limited to legal issues, issues related to the ease of searching information and preventing people (especially adolescents) from the negative impact harmful content. Before the rapid growth of the social media and lower volume of user generated content, the effort to manually detect and moderate comments or forum posts containing such language was manageable. Nowadays, however, due to the prevalence of the social media, it becomes economically infeasible to manually moderate all the comments. This has resulted in an active development of techniques of automated detection of abusive language [9] [2]. A lot of published papers are dedicated to the detection of hate speech and/or abusive language in comments

[3] We refer to surzhyk as to a specific mixed sociolect in Ukrainian-Russian bilingualism [5]

Aleš Horák, Pavel Rychlý, Adam Rambousek (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2018, pp. 77–84, 2018.

in English. According to our best knowledge, there are no adequate methods for automatic detection of abusive language and/or hate speech for Russian and Ukrainian language. Development of such methods is a difficult task because several reasons:

- There are no labeled databases or corpora of comments/posts in social media with abusive content in these languages;
- Due to the peculiarities of word formation in Russian and Ukrainian languages, it is practically impossible to define a finite list of abusive words; [6] – people often invent new, obviously obscene words, by linking two common stems by an infix or attaching prefixes and/or suffixes.
- Use of surzhyk creates great variety of abusive words.

Similarly to other languages, people sometimes try to mask the use of the abusive language by using euphemisms, inverting the order of letters in a word, replacing letters with stars or other symbols etc. Due to the fact that often social media comments in Ukrainian social media environment is written in surzhyk, we treat Russian and Ukrainian languages the same. This approach is reasonable, because most of the stems of abusive words are the equal in these languages. The goal of our research is to develop an automated approach for detecting abusive and hateful speech in Russian and Ukrainian languages in social media. Given very limited availability of the labeled data, in this research we focus on the unsupervised probabilistic techniques by using a seed dictionary of abusive terms as input.

2 Related work

Literature describes many approaches for automatic detection of hate speech and abusive language in social media. There are solutions for very specific types of hate speech (for example, Jihadist hate speech [12]) as well as for general offensive speech and hate speech detection (for example, [3]).

For the cases when there is a limited availability of labelled data sets (as with Russian and Ukrainian languages) unsupervised learning methods are often used. Thereby we mainly focus on unsupervised methods in this section.

Unsupervised learning techniques are commonly used for the natural language processing task. For example in [13], authors use these techniques for the sentiment classification of Chinese Text. They showed that the unsupervised techniques produced results that are surprisingly close to the ones obtained by the use of supervised learning. At the same time, unsupervised techniques are not usually susceptible of being domain- or language-specific.

Many approaches for extracting features from the natural text rely on some sort of external knowledge. Seed dictionaries is a form of such an external knowledge, which are relatively cost-effective to be assembled. There are different application areas of such seed dictionaries. For example, they may be used for entity extraction, automatic translation and many other application. It is common to use seed dictionaries in conjunction with unsupervised learning

techniques. For example, [4] used unsupervised learning with a seed dictionary of entities of multiple classes for the task of pattern learning and entity extraction in informal text corpuses. Furthermore, this approach was used for detection of abusive and hate language by [8].

There is little research dedicated to the automated detection of the offensive language in Russian and Ukrainian texts. Existing work for these languages focus on other tasks. For example, [11] developed a system for classifying Russian text into thematic categories. However, their main focus was to detect the content that is illegal in Russia (e.g. related to selling narcotics on the Internet) rather than detecting abusive speech. They also targeted all web sites which usually contain large blocks of texts with a small number of orthographic and grammatical errors rather than social media resources, where most of the comments are short and often contain slang, mistakes and/or obscured offensive speech.

3 Methodology

The core idea of our approach is to use a seed dictionary of abusive terms in combination with unsupervised assignment of labels (abusive or not abusive) to social media comments and then iteratively expand the initial seed dictionary with abusive and obscene words. Our hypothesis is that inappropriate comments contain in many cases more than one abusive word, thus the likelihood of abusive words appearing together in a single comment is sufficiently high to extend the dictionary. An overview of the workflow of our method is depicted in Diagram 1.

Before applying any algorithms to the textual data, we applied the following pre-processing steps: we removed punctuation, numbers, emoticons and other non-alphabetic symbols. We choose to remove those features, even though they can be strong indicator of toxic speech, because we consider words to be the most important features. Moreover, it allows us to manually evaluate the correctness of our method. In order to check the influence of word formation in Ukrainian and Russian languages, we first trained our model with words as they were present in the dataset and thereafter by using the words reduced to their stem. For evaluation of the results, we split the dataset into training, validation and test datasets, where the validation and test datasets were manually labeled by multiple native speakers.

After the dataset was pre-processed, each comment was automatically labelled as abusive or non-abusive by applying the seed dictionary, where a comment was assigned to contain abusive language, if at least a single word from the dictionary was present in the comment. This approach may introduce some false positives in case when abusive word is used in decent context, such as quote, however such occurrences are extremely rare in the context of social media. Then based on these initial labels, the dataset was split into two classes: abusive and non-abusive. Then for every word that appeared in the dataset more times than a threshold, a likelihood of being in the abusive or non-abusive class

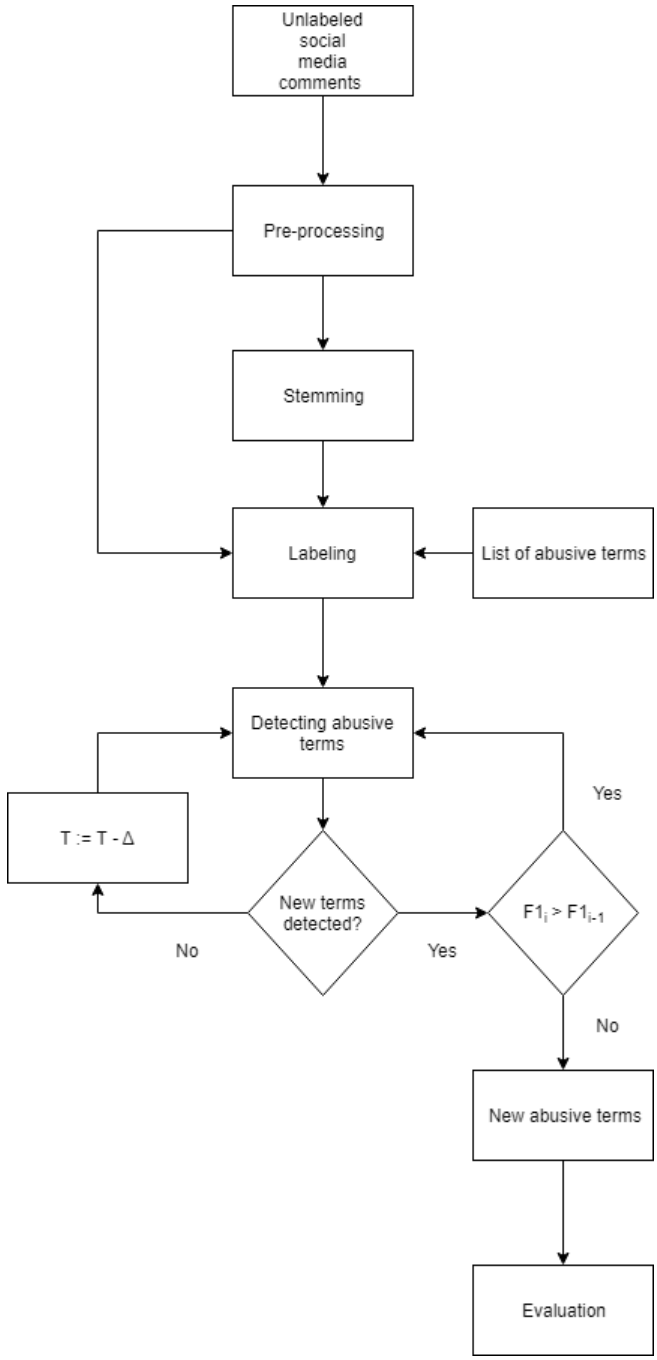


Fig. 1: High-level workflow

was computed. To that end, we used the frequency of the occurrences based on a threshold to filter out rare words that can be falsely classified as abusive.

Having initial likelihoods for each term we estimated its probability to be abusive. For the estimation we used relative distance and log odds ratio metrics. The formula of relative distance and log odds ratio are presented on Eq. (1) and Eq. (2) respectively:

$$P(x) = \frac{p_b(x) - p_g(x)}{max(p_b(x), p_g(x))} \tag{1}$$

$$L(x) = \frac{\frac{p_g(x)}{1-p_g(x)}}{\frac{p_b(x)}{1-p_b(x)}} \tag{2}$$

Where x is selected term, $p_b(x)$ - likelihood of x being in abusive part, $p_g(x)$ - the likelihood of x being in the non-abusive part.

In order to decide if term is abusive or not, we compare its estimation of being abusive to a given threshold T. This threshold was set to 0.95 based on the empirical data. If the estimation metric exceeds T, then the selected term is added to the dictionary of abusive words.

After all newly found abusive terms were added to the list, we evaluated our classifier on validation dataset. For evaluation we use precision, recall and F_1 score, which are calculated as in Eq. (3):

$$\begin{aligned} P &= \frac{tp}{tp+fp} \\ R &= \frac{tp}{tp+fn} \\ F_1 &= \frac{2 \cdot P \cdot R}{P+R} \end{aligned} \tag{3}$$

Where P is precision, R — recall, F_1 — F_1 score, tp — number of true positives, fp — number of false positives and fn — number of false negatives.

If the F_1 score is bigger than the F_1 score calculated on previous step, training dataset is re-labeled with expanded dictionary and estimation for each term are recalculated. If none of estimations exceeds threshold T, the threshold T is lowered by delta. If the F_1 score is lower than on the previous step, the iterative process stops and results are evaluated on the test dataset. The final dictionary then contains all words from the initial seed dictionary together with all newly found abusive words.

4 Experimental setup

In our experiments we assembled a dataset based on YouTube comments. We have chosen YouTube[4] since it was shown by [7] that flaming and abusive

[4] https://www.youtube.com/

language are common on this particular social media platform. We manually selected videos related to the topic of Ukrainian Revolution of Dignity, also know as Euromaidan [10]. Events of this revolution stirred a high controversy in all types of media, thus leading us to believe that YouTube videos on this topic will have high percentage of abusive language.

We collected comments from 329 videos, which were all related to the topic of Euromaidan. During exploratory analysis of collected comments, we noticed that some comments are written in transliteration - using English alphabet to write Russian or Ukrainian words. Such comments were deleted from our dataset. After cleaning and pre-processing our final dataset comprised more than 50,000 comments [5]. From this dataset 2,000 comments were randomly selected for manual labeling by native speakers [6]. The manual annotators found 32.7% of all comments contained abusive language.

For the seed dictionary we used crowdsourced list, which contained over 600 abusive and obscene words [7]. In addition, we also evaluated our approach on a minimal seed dictionary comprising only the 5 most top-used abusive words [8] in order to check whether our approach will still work with such a small seed dictionary.

The pre-processing, e.g. stemming, of the text of the comments were performed using the Natural Language Toolkit [1]. We used the settings for the Russian language for all terms in our dataset.

5 Results

We report the results of our approach for a number of different configurations, in order to assess the influence of various parameters and settings on the final performance. In Table 1 the results of our evaluation of our approach using different probabilistic estimation metrics are presented.

In our evaluation we found that stemming improves recall but at the same time reduces precision and overall slightly improves F_1 score. When comparing metrics of relative distance and log odds ratio, it can be seen that in all cases when using log odds ratio a high recall has been achieved, but the precision did drop considerably. With relative distance we achieved an improvement in recall and slight decrease in precision. When using the micro seed dictionary, we noticed that the use of log odds ratio metric leads to a rapid growth of the seed dictionary such that in the end it contained many non-abusive words. At the same time, the use of relative distance metrics provided more conservative results: the size of initial seed dictionary grew from 6 to 23 terms, where each term was indeed abusive.

We were surprised to find that our algorithm was able to pick up several ethnophaulisms that were not a part of initial seed dictionary but are definitely

[5] https://github.com/bohdan1/AbusiveLanguageDataset/blob/master/data.csv
[6] https://github.com/bohdan1/AbusiveLanguageDataset/blob/master/labled.csv
[7] https://github.com/bohdan1/AbusiveLanguageDataset/blob/master/bad_words.txt
[8] https://github.com/bohdan1/AbusiveLanguageDataset/blob/master/bad_words_seed.txt

Table 1: Results achieved using our approach
SD – seed dictionary
RD – relative distance
LOR – log odds ratio
STM – stemmed dictionary
MSD – micro seed dictionary
Words – number of new abusive terms added to the seed dictionary

Method	P	R	F_1	*Words*
SD	0.875	0.510	0.644	—
SD, RD	0.736	0.629	0.678	8
SD, LOR	0.678	0.629	0.652	11
STM, SD	0.742	0.629	0.681	—
STM, SD, RD	0.667	0.675	0.671	10
STM, SD, LOR	0.474	0.741	0.578	13
MSD	0.857	0.158	0.268	—
MSD, RD	0.588	0.463	0.518	44
MSD, LOR	0.303	1.000	0.465	573
STM, MSD	0.897	0.231	0.368	—
STM, MSD, RD	0.684	0.344	0.458	17
STM, MSD, LOR	0.308	0.920	0.462	145

offensive. Examples include words “хохлы” (khokhly) and “кацапы” (katsapy) which are derogatory names for people of Ukrainian and Russian nationalities respectively.

6 Conclusion and Future work

We demonstrated that unsupervised automatic labeling approach is a feasible choice for automatic detection of abusive speech in social media comments in Russian and Ukrainian languages as well as for surzhyk. As expected, we observed a slight drop in precision for the automatic population of the abusive word dictionary, there was a considerable gain in terms of recall. We found that the available pre-processing tools for the Slavic languages lag behind their counterparts for languages like English. For example, the stemming approach is based on a simple heuristic, which is not fully capable to match the Russian and Ukrainian languages. Considering everything mentioned above, we have outlined the following steps for our future research:

(i) develop a robust tool for lemmatization and stemming for both languages
(ii) develop algorithm for detecting hate speech aimed at nationality.
(iii) expanding dataset of social media comments and training word embedding models.

References

1. BIRD, S., KLEIN, E., AND LOPER, E. *Natural Language Processing with Python*, 1st ed. O'Reilly Media, Inc., 2009.
2. CHEN, Y., ZHOU, Y., ZHU, S., AND XU, H. Detecting offensive language in social media to protect adolescent online safety. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing* (Sept 2012), pp. 71–80.
3. DAVIDSON, T., WARMSLEY, D., MACY, M. W., AND WEBER, I. Automated hate speech detection and the problem of offensive language. *CoRR abs/1703.04009* (2017).
4. GUPTA, S., AND MANNING, C. D. Improved pattern learning for bootstrapped entity extraction. In *CoNLL* (2014).
5. LEWCZUK, P. Socjolingwistyczny status surżyka. *LingVaria*, 21 (2016), 177–189.
6. MOKIENKO, V. M. Russkaya brannaya leksika: tsenzurnoye i nyetsenzurnoye. *Rusistika* (1994).
7. MOOR, P. J., HEUVELMAN, A., AND VERLEUR, R. Flaming on youtube. *Computers in Human Behavior 26*, 6 (2010), 1536 – 1546. Online Interactivity: Role of Technology in Behavior Change.
8. MUBARAK, H., DARWISH, K., AND MAGDY, W. Abusive language detection on arabic social media. In *Proceedings of the First Workshop on Abusive Language Online* (2017), Association for Computational Linguistics, pp. 52–56.
9. PITSILIS, G. K., RAMAMPIARO, H., AND LANGSETH, H. Detecting offensive language in tweets using deep learning. *CoRR abs/1801.04433* (2018).
10. SHVEDA, Y., AND PARK, J. H. Ukraine's revolution of dignity: The dynamics of euromaidan. *Journal of Eurasian Studies 7*, 1 (2016), 85 – 91.
11. SIDOROVA, E., KONONENKO, I., AND ZAGORULKO, Y. An approach to filtering prohibited content on the web. In *DAMDID/RCDL'2017* (2017).
12. SMEDT, T. D., PAUW, G. D., AND OSTAEYEN, P. V. Automatic detection of online jihadist hate speech. *CoRR abs/1803.04596* (2018).
13. ZAGIBALOV, T., AND CARROLL, J. Automatic seed word selection for unsupervised sentiment classification of chinese text. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1* (Stroudsburg, PA, USA, 2008), COLING '08, Association for Computational Linguistics, pp. 1073–1080.

Understanding Search Queries in Natural Language

Zuzana Nevěřilová[1,2] and Matej Kvaššay[1]

[1] Konica Minolta Laboratory Europe, Brno
[2] NLP Centre, Masaryk University, Brno
{zuzana.neverilova,matej.kvassay}@konicaminolta.cz

Abstract. This work is part of a project aiming to provide one single search endpoint for all company data. We present a search query parser that takes a speech-to-text output, i.e. a sentence. The output is a structured representation of the search query from which a SPARQL query is generated. The SPARQL is then applied to an ontology with the company data.
The parsing procedure consists of two steps. First, the search intent is detected, second, the query is parsed based on the search intent. For the intent classification, we use word embeddings with boosting of top 5 words, and support vector machines. For the parsing, we use semantic role labeling, named entity recognition, and external resources such as ConceptNet and DBPedia. The final parsing step is rule-based and related to the ontology structure.
The intent classifier accuracy is 94%. In the subsequent manual evaluation, the resulting structures were complete and correct in 51% cases, in 34.57% of cases it was complete and correct but it also contained irrelevant information.

Keywords: search intent; search query parsing

1 Introduction

Search in corporate data is one of the pain points for many people working in the office. Apart from searching physical objects, they also look for digital objects. This task is considered to be annoying and surprisingly difficult. In our tool, we aggregate all possible data sources the company is working with, such as company wiki, emails, task management tool, employee profiles, or instant messages, so a powerful search engine is a must-have. The user interface allows among other voice inputs. The voice signal is transcribed into text and the system has to interpret this text into a search query. Such inputs are completely different from the common search queries which are mostly keyword-based.

In case of search queries in natural language, we have to parse and interpret a short text, consisting mostly of one or two sentences with many entities and named entities. Often the sentence expresses relationships between the entities. Sometimes, implicit knowledge has to be added to the interpretation.

Aleš Horák, Pavel Rychlý, Adam Rambousek (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2018, pp. 85–93, 2018.

The system has three main components: one detects the main search intent (e.g. a file or a person), the other parses the query into a structure from which a SPARQL query is constructed. In the last component, the search results are ranked and presented to the user from the highest rank.

1.1 Paper Outline

In Section 2, we describe in short the current aspects of search engines and the current search query parser. Section 3 focuses on methods we have used, particularly on . Section 4 discusses the evaluation criteria. Section 5 contains final remarks.

2 Related Work

Search platforms are nothing new. Even in the open source world, one can find search engines combining full text search with search query parsing, faceted search (interactive filtering), synonym expansion, and other features. One such platform is Open Semantic Search [3].

Most search engines are keyword-based fulltext (such as search in the Web) or faceted-based (such as search in a library) or combination of both. In addition, search engine can provide instant feedback or clarification dialogue, and thus, in such cases it becomes something between a search engine and a question answering system. The dialog-like search is also present in personal assistants such as Siri, Cortana, Alexa, and other where the interface is spoken.

The presentation of search results is not a simple list of items anymore. Even web search engines try to guess the user search intent and in some cases provide the direct answer. For example, Google Search provides a calculation if the user enters a mathematical formula, a conversion if the user enters query such as "15 EUR in CZK", or a description if the user enters a named entity (e.g. for "Marylin Monroe" it returns "a film actress"). Many aspects of search query presentation are described in [4].

In the current version of our project, the search engine provides faceted search as shown in Figure 1 and sentence query search intended to work together with voice input. Users can also write search sentences into the search input box but in reality, nobody expects them to do so. For parsing the search sentence, we use Google DialogFlow [4] with predefined dialog intents. The output of sentence parsing by DialogFlow is a structured object such as a Python `dict`, example of such output can be seen in Figure 2.

The granularity of intents is quite high, e.g. searching documents shared with someone is different intent than searching documents from a meeting. DialogFlow is provided by 10–30 example queries and creates a generalization for this particular intent.

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

[4] `http://dialogflow.com`

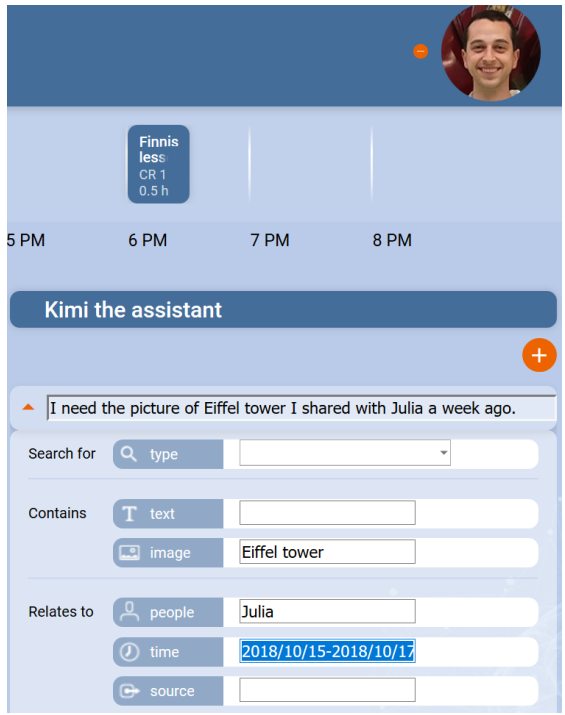


Fig. 1: Search input box and faceted search presented in the frontend

```
{
 ...
 "parameters": {
  "openingPhrase": "find",
  "givenName": "Bob",
  "lastName": "",
  "document": {
    "type": "presentation",
    "topic": "artificial intelligence"
 },
  ...
  "score": 0.9966866513437793
}
```

Fig. 2: Sample output for input “Find the presentation about artificial intelligence that Bob sent to me”.

It seems that DialogFlow extensibility is limited: every time a new intent is added to the current ones, the confidence numbers for all intents decrease.

The aim of this work is to provide a search query parser with at least the same accuracy as DialogFlow on transcribed sentences and higher extensibility.

3 Methods

The design of the search query parser consists of two basic modules: intent classifier and search query parser adapted to a particular intent. For example, the prepositional phrase beginning with "in" means usually a location. In case of digital objects, this location is digital as well (for example, "in office" means "in the Office application"), while in case of physical objects, the location is also physical (for example, "in office" means "in somebody's office").

3.1 Intent Classifier

Problem definition According to an internal survey, people working in office most frequently look for text documents (21% cases), persons (10% cases), multimedia files (7.8% cases), personal belongings, emails (6.4% cases), web pages, locations, tasks, presentations and others with lower frequency. Based on this survey, we identified six classes of search intents, described in Section 4.

Proposed model To classify intents, all tokens of the query are lower-cased and stop-words are removed. Tokens are mapped to 300-dimensional word embeddings using publicly available vocabulary of FastText [2] vectors trained on CommonCrawl dataset. Missing words are ignored. Vectors are then aggregated by averaging in two ways:

1. Average of vectors of first k words of the query
2. Average vectors of all words of the query

Both vectors are then concatenated into single 600-dimensional representation for each sample. Motivation for this **double representation** is our observation that natural language search query often contains most informational words for intent classification at the beginning of the query sentence (e.g. word "document" in "look for a document which contains image of elephant"). By averaging words from beginning of the sentence, we encode information about beginning of the search phrase and let the classifier exploit this positional-specific information. This simple approach enables the classifier to focus on specific parts of the query. The trade-off is the increase in feature vector dimension.

We used SVM [1] classifier with RBF kernel that is also able to evaluate confidence of the prediction.

3.2 Search Query Parser

We assume the search query is composed of one or a few sentences. First, we apply semantic role labeling to each sentence, then we apply rules to parse each of the arguments. The rules depend on the search intent.

Semantic Role Labeling Semantic role labeling (SRL) decomposes each clause of a sentence to predicate-argument structure. Historically, SRL used syntactic parsing, however, the Deep SRL [3] which is based on neural networks outperforms the previous approaches.

We use Deep SRL as a server that for a given sentence outputs separate clauses. For each clause, it outputs the predicate and its arguments. The arguments are the same as in PropBank[5]: numbered arguments `ARG0`–`ARG5`, and predicate and phrasal modifiers (e.g `ARGM-LOC` for locations or `ARGM-TMP` for temporals). We do not consider PropBank links.

Rule-based Argument Parsing Each numbered argument is rule-based parsed. In future, we consider to induce the rules from the ontology scheme but in the current version, the connection with ontology is very limited. We treat predicate and phrasal modifiers, and numbered arguments in different ways.

Predicate and phrasal modifiers For arguments of type `ARGM-LOC` or `ARGM-TMP`, the parsing is straightforward: we consider the whole content of the argument as one unit of the same type as the argument (e.g. location or time).

Argument containing the main intent We parse the argument that contains the main intent in a different than the other numbered arguments. The main intent is always the syntactic head of the argument (if is not, the parsing cannot continue). All dependent components are modifiers of the intent. For example, if the argument contains "pdf file", the main intent is "file" and "pdf" is a constraint to file format.

Arguments not containing the main intent Other numbered arguments are processed together with the predicate since the predicates describe relations (e.g. contain, create, share, ...) between the main intent and other entities. If the extracted entities are recognized as potential objects in the graph database (such as users), they have to have a relationship to the main intent or other objects. In other cases, the entities are identified as keywords. We use SpaCy[6] with large English model for tagging and recognizing named entities.

The overall result of the parsing is a structure. An example can be seen in Figure 3. In the output structure, we consider only autosemantic tokens, however, other part of speech can modify the relation. For example, if the query contains a named entity "Bob", it can be interpreted as the owner or creator

[5] `https://propbank.github.io/`

[6] `https://spacy.io`

```
{
...
 "tokens": [
    {"text": "document",
     "relation": {"value": "intent", "confidence": 0.5},
     "label": {"value": "presentation", "confidence": 1.0}
    },
    {"synonym": [
        {"value": "AI", "confidence": 0.5},
        {"value": "ai", "confidence": 0.5}
     ],
     "informationSource": "conceptnet",
     "text": "artificial intelligence",
     "relation": {"value": "keyword", "confidence": 0.5}
    },
    {"text": "Jane Smart",
     "entity": {"value": "PERSON", "confidence": 0.5},
     "relation": {"value": "sharedWithPERSON", "confidence": 0.5}
    }
  ]
}
```


Fig. 3: Sample output for input "Find the document about artificial intelligence that Jane Smart provided to me.".

of a document (e.g. "find documents by Bob") but it can be also interpreted as a keyword in the document (e.g. "find documents about Bob"). Multiword expressions are identified and treated as a single token. Foreach token, the *relation* is determined. The token text and token relations are necessary, since the resulting structure is later converted into a SPARQL query in the form of triples $(mainintent, relation, label)$.

We process multiword expressions, using syntactic constraints (NOUN-NOUN, ADJ-NOUN, PROPN-PROPN) and external resources. Particularly, we use ConceptNet[7] and DBPedia[8] to confirm that a multiword expression candidate is a single meaning unit. In most cases, the greedy approach (preferring "team building" over "team" and "building" which all three exist in external resources) is the best. In addition, ConceptNet provides synonyms that can later be used to expand the SPARQL query.

4 Evaluation

We evaluated the two parts of the system separately.

[7] `http://conceptnet.io/`
[8] `https://wiki.dbpedia.org/lookup`

4.1 Query intent classification

For evaluation we used internally created data-set of 441 query examples from 6 categories:

- Calendar event (cal)
- Message (msg) – email or instant message
- Multimedia content (mul) – image, video, or audio files
- Personal information (per)
- Task (tsk) – task description optionally with an assignee and a due date
- Text document/spreadsheet (txt)

The data were split 50/50 into category-balanced sets of 220 examples for training and 221 examples for testing due to small available sample size. We evaluated proposed query intent classifier with double representation and ad-hoc selected $k = 5$ against a baseline model, which was the same model without additional average vector for 5 first tokens.

Table 1: Test set performance model with single (word vector average, dim=300) and double (word vector average + first 5 words vector verage,dim=600) representation.

	single representation				double representation			
Category	Precision	Recall	F-1	Support	Precision	Recall	F-1	Support
cal	0.94	0.83	0.88	36	1.00	0.94	0.97	36
msg	0.80	0.75	0.77	32	0.90	0.84	0.87	32
mul	0.80	0.62	0.70	26	1.00	0.96	0.98	26
per	0.93	0.88	0.90	48	0.94	0.96	0.95	48
tsk	1.00	0.90	0.95	10	1.00	1.00	1.00	10
txt	0.74	0.91	0.82	69	0.90	0.96	0.93	69
micro avg	0.83	0.83	0.83	221	0.94	0.94	0.94	221
macro avg	0.87	0.81	0.84	221	0.96	0.94	0.95	221
weighted avg	**0.84**	**0.83**	**0.83**	221	**0.94**	**0.94**	**0.94**	221

In the test set classification performance results (Table 1), we show that the model with double representation achieves high average precision and recall (≥ 0.94) on the test set. We have also shown that adding average vector of first 5 tokens to all word vector average improved the results by at least 10% compared to baseline. The confusion matrices are presented in Figure 4.

4.2 Evaluation of the Search Query Parser

A crucial question for evaluation is whether the parsed structure can be transformed to a SPARQL query that returns correct results. We realized that

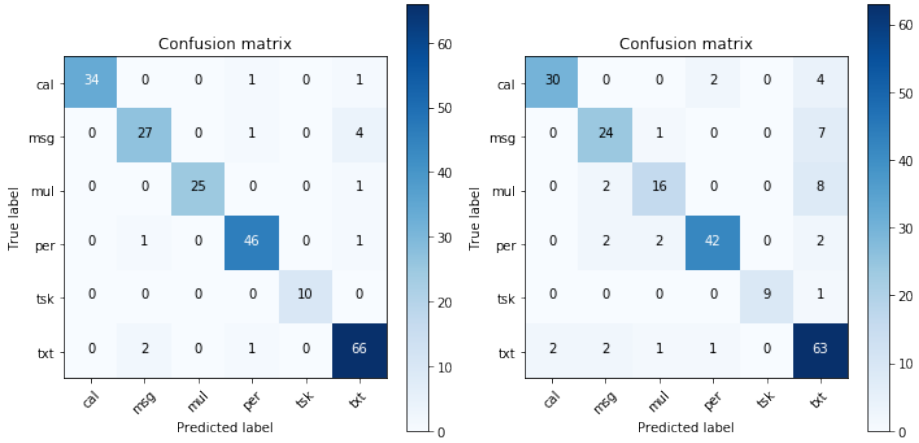


Fig. 4: Test set confusion matrix for model without double representation: word vector average + first 5 words vector average, dim=600 (left) and word vector average, dim=300 (right)

some ambiguity is present even in search queries, e.g. a presentation can be a video, an event, or a PDF/PPTX file. We therefore considered the parsing to be correct if it returned one meaningful interpretation of the sentence. We also wanted that all relevant parts of the sentence were considered in the parsed structure. The relevance was judged using a common sense interpretation of the sentence.

We evaluated manually the parsing on 80 example search sentences. 41 sentences were parsed completely and correctly. In 11 sentences, a relevant token was not recognized. In 3 cases out of these 11, it was a related person, in the remaining cases, it was a keyword. In 28 cases, an irrelevant token was extracted and included in the output structure. This was a case in sentences such as “find an AI expert in the London office” where the word “office” is not relevant for the search. In 6 cases, the relation was detected incorrectly.

5 Conclusion and Future Work

We proposed a natural language search query intent classification model with double representation allowing classifier to focus on specific part of the query and we have shown that this representation can significantly increase the classifier performance in experimental setting.

The sentence parser benefits from the intent classification, and uses semantic role labeling. Parsing of each argument is rule based. Even though we evaluated the parser on a limited number of sentences, we can see that its recall is plausible.

Possible improvements of the query intent classification model include using shorter word embeddings and more granular split of queries to include more

position-specific information into the feature vector and usage of domain-specific word embeddings additionally to embeddings model trained on public data-set.

We also plan to tie the parser more closely to the ontology scheme. The ideal situation would be a parser that can adapt on the ontology scheme modifications.

Acknowledgements. This research was supported by Konica Minolta Laboratory Europe. This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin OP VVV project CZ.02.1.01/0.0/0.0/16_013/0001781.

References

1. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (Sep 1995)
2. Grave, E., Mikolov, T., Joulin, A., Bojanowski, P.: Bag of tricks for efficient text classification. In: In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. pp. 427—-431 (2017), http://arxiv.org/abs/1607.01759
3. He, L., Lee, K., Lewis, M., Zettlemoyer, L.: Deep Semantic Role Labeling: What Works and What's Next. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (2017)
4. Tunkelang, D.: Search Results Presentation (Feb 2018), https://queryunderstanding.com/search-results-presentation-7d6c6c384ec1

Document Functional Type Classification

Kristýna Němcová

[1] Konica Minolta Laboratory Europe, Brno
[2] NLP Centre, Masaryk University, Brno
kristyna.nemcova@konicaminolta.cz

Abstract. This paper presents methods used to classify documents into functional types (e.g. invoices, orders, scientific papers). We analyzed the current solution and we reproduced it with improvement. The problem is divided into classifications based on text and layout, then the results are combined. The work is applicable in office environment e.g. for searching according to a functional type. When appropriately combined with systems designed for a specific functional type, our work can contribute to the system performance.

Keywords: classification; documents; functional type

1 Introduction

In the business world, document processing is crucial. Knowing the functional document type facilitates work with each document. We can for example use different models for named entity extraction for marketing brochures and invoices. We identified thirteen functional types:

- Brochure
- Contract
- Financial Report
- Invoice
- Meeting minute, memo
- NDA
- Order (Purchase order)
- Patent
- Project charter (plan, gantt)
- Project status report
- Questionnaire
- Scientific article
- Technical Specification

Currently, we use the HyDoc functional type classifier. However, it has some issues described in Section 2. Moreover, we wanted to discover whether there are better methods to solve the classification.

Aleš Horák, Pavel Rychlý, Adam Rambousek (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2018, pp. 95–100, 2018.

1.1 Paper Outline

In Section 2, we describe the HyDoc classifier. Section 3 focuses on methods we have used, particularly on the layout-based classifier described in Section 3.2. Section 4 discusses the evaluation criteria. Section 5 contains final remarks.

2 Related Work

2.1 HyDoc Description

HyDoc document classifier is a system which is capable to perform document classification using both text and visual features (page layout). The system relies on two separate classifiers for the text and visual part; in particular, the visual part uses a convolutional neural network to classify a random sample of document pages: single-page classifications are then combined to yield a global estimate for the document class using Bayesian inference. A final ensemble neural network employing two hidden layers combines results from the text and visual part, providing an estimate for the document class. A set of confidences over all the classes are returned. [3]

2.2 Problems with HyDoc

As mentioned above, visual part of HyDoc classification system depends on random page samples. Randomization, that causes program to be non-deterministic, is a major flaw of their solution. Sometimes outcomes are diametrically dissimilar and it is hard to evaluate actual results.

3 Methods

In this section, we describe the process, we built in order to obtain the same or better results as the current HyDoc classifier. Similarly to HyDoc, we set up two classifiers, one taking text features, the other taking visual features.

3.1 Text-based Classifier

The text-based classifier is similar to HyDoc as the text-part is not problematic. We separated text from documents and applied fastText pre-trained word vectors trained on Wikipedia [2]. Simple two layer neural network was trained.

3.2 Layout-based Classifier

The classification based on document layout was a more complicated problem as there is a lot of research done in single page document image classification but almost none in purely multiple pages. This fact caused selection of combined single page classifier of document. The concept of transfer learning was used and we retrained ResNet-152 [1] for single page classification. We classify single pages and then combine the results. In addition, some document types have typically multiple pages, others have typically one page.

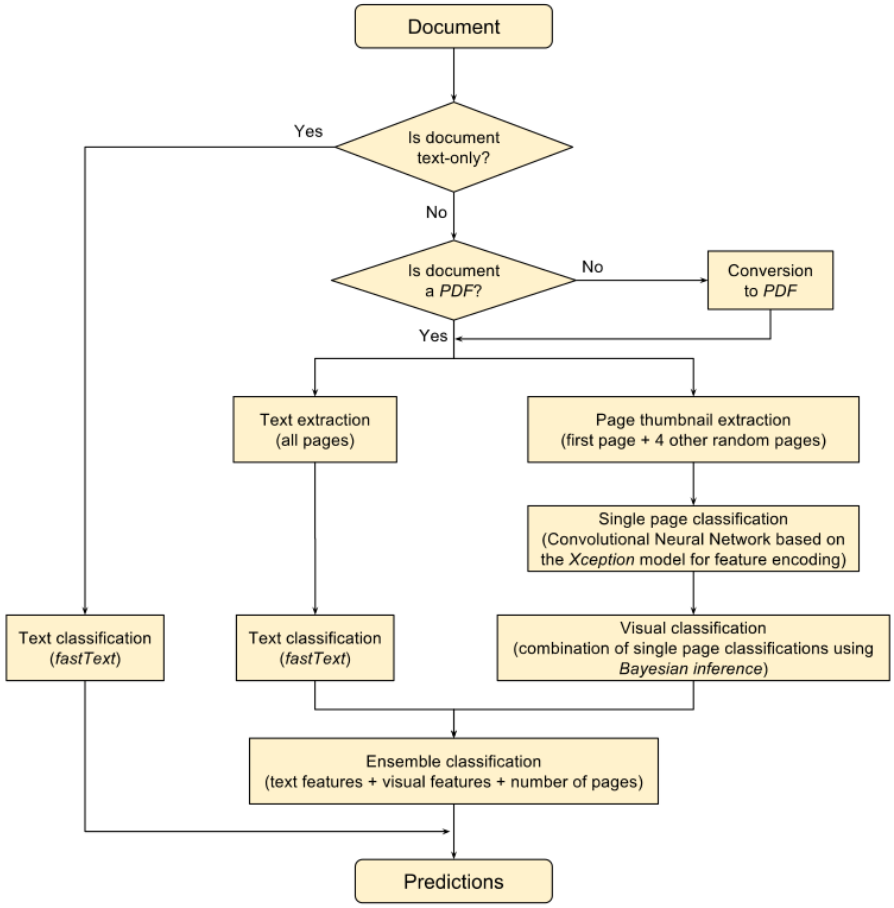


Fig. 1: Flow chart of HyDoc solution as provided by [3]

Possible Approaches for Layout-based Classification

Randomly Selected Pages HyDoc uses the questionable random selection with one classifier. The idea behind this could be the possibility of diverse document where pages are completely different. But that is unlikely to happen and therefore there are more cons than pros as described earlier.

Pages from Predefined Positions of the Document The correct approach seemed to be to take distinct pages in classification. The information about where the page is located in document and training only pages at the same position were vital. The unsophisticated solution would be to create multiple classifiers. The elegant answer was to convert images to different dimensions starting with three

(red, green, blue) and then add more dimensions. Nevertheless, this method has proven to be less promising presumably due to a small dataset.

All Pages of the Document The remaining solutions are based on one classifier. For instance, one can simply take all pages of the document. It works well on the train dataset. However, a problem occurred on validation and test dataset. There was recognizable over-fitting. After a closer look at the data, it is obvious why, because some of the documents have even more than thousand pages. Pages look very similar and therefore the classifier gives more weight to documents with more pages. Moreover it takes a lot of time to process such amount of images.

First Twenty Pages of the Document Experiments with number and position of pages were done (see Table 1) and the most promising approach was to take first twenty pages. It seems to be a compromise between a small dataset and massive over-fitting.

Fig. 2: Visualization of a data batch.

After a single page classification we need to combine the results on one document together as a outcome of layout-based classifier. So far, the used approach is to simply add probability matrices into a new matrix.

3.3 Final Classifier

The final classification is done by a meta-classifier. It takes results from text and layout-based classifiers as inputs of a simple neural network. Nonetheless, the problem with final classifier centers around a small dataset as we used 80 % of our data for training the previous classifiers.

4 Evaluation

The dataset contains at least hundred documents per each of the 13 functional types which means around 80 training documents per class. That is quite a small

amount. In a single image classification, we overcame this problem as we use up to 20 pages therefore the dataset become sufficient. Although as described above, this problem appears especially with combining results from text and layout.

The text data from documents provide reliable results as expected. We experimented with the neural network architecture and finally got the accuracy of 0.910.

As the decision to use only one classifier for all pages in layout-part was done, we needed to decide what pages to take into consideration. The best result came with first twenty pages combined with L2 penalty to prevent over-fitting [4]. The accuracy is 0.548.

Table 1: The results of approaches with one classifier

Pages	Accuracy
first page only	0.478
1st, 2nd, 2 pages from the middle, last page	0.521
all pages	0.532
first twenty pages	0.548

5 Conclusion and Future Work

In this paper, we have shown methods for classifying documents based on their functional type. We described an existing solution and examined its flaws. Our own approach based on two separate classifiers for text and layout was introduced. The text-part is robust and straightforward. In the layout-part, the single page classifier was properly examined and completed.

For the final layout-based classification, single page probability matrices are added. A more complex solution would be to compute confidence of matrix and add only few or let them vote or even to train a meta-classifier. Future experiments will show what is better.

Final prediction of functional types combines text and layout-based classifiers. The dataset for this part is small. The question is whether to include number of pages as a feature. It could provide a valuable insight, but also bias the outcome due to the data.

Given the current result, we can fearlessly say that our final classifier will have accuracy around 90% as the most weight is lying on the text-part. The layout-part will provide higher stability in cases of visually recognizable documents.

Acknowledgements. This research was supported by Konica Minolta Laboratory Europe.

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. ArXiv e-prints (Dec 2015)
2. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. CoRR **abs/1607.01759** (2016), http://arxiv.org/abs/1607.01759
3. Konica Minolta Laboratory Europe: HyDoc: Software Manual. Version 1.0. Unpublished
4. Krogh, A., Hertz, J.A.: A simple weight decay can improve generalization. In: Proceedings of the 4th International Conference on Neural Information Processing Systems. pp. 950–957. NIPS'91, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1991), http://dl.acm.org/citation.cfm?id=2986916.2987033

Part IV

Text Corpora

Improving Compound Adverb Tagging

Hana Žižková

Masaryk University, Faculty of Arts,
Arna Nováka 1, 60200 Brno, Czech Republic
zizkova@phil.muni.cz
http://www.phil.muni.cz

Abstract. This paper describes the corpus probe we made to obtain and analyze data with a focus on improving compound adverb tagging. Thanks to our research we gain large amounts of unrecognized units that resemble to compound adverbs. We manually selected 470 units and we examined whether they are listed in existing Czech dictionaries and how they are tagged in corpus if we respread it into multiword expression. We found out that the compound adverb tagging in Czech National Corpus is inconsistent and unsatisfactory, so we proposed three solutions for improving compound adverb tagging.

Keywords: compound adverb; automatic morphological analysis; tagging

1 Introduction

Compound adverbs represent an interesting issue in terms of automatic morphological analysis (AMA). In Czech, the compound adverbs are always formed from a preposition and a noun or a preposition and an adjective or a preposition and a pronoun or a preposition and a numeral or a preposition and an adverb. Recognition of compound adverbs by AMA is difficult because, Czech compound adverbs are written mostly together as one word, but often there exists a multiword expression and their meaning is the same (*na příklad* – *například*) [1]. For instance Dokulil [2] states that: "compound adverbs are formed by compounding frequently occurring words in a sentence, without any change in their form. It is characteristic for them that you can always divide the compound adverb again." For the purposes of this paper it is essential that we write compound adverbs mostly together as one word, but often in parallel compound adverbs there exists a multiword expression. Additionally, a member of the multiword expression can function independently of this expression as a separate word [3]. Multiword expressions can be "defined as expressions which are made up of at least two words and which can be syntactically and/or semantically idiosyncratic in nature. Moreover, they act as a single unit at some level of linguistic analysis." [4]

There are contexts in which one may hesitate whether to use a one-word adverb or a multiword expression (*Obarvit načerno.* vs. *Obarvit na černo.*). Another

Aleš Horák, Pavel Rychlý, Adam Rambousek (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2018, pp. 103–109, 2018.

important feature of the compound adverbs is that when written as two (or more) words, it is not possible to insert another expression between the two words that could develop the unit (*například – na příklad*, but not **na dobrý příklad*).

It is important for the compound adverb to be recognized by AMA in both cases (as a one-word and also as a multiword expression) regardless of whether the codification determines what is the correct spelling of the compound adverb. The automatic morphological analysis takes place in three steps: the first is a division of word forms (tokenization), the second is an assignment of one, but usually more interpretations from the morphological dictionary and the third step it is the disambiguation, which means assigning an interpretation [5].

The AMA recognizes and correctly identifies such compound adverbs which are written as one-word and are listed in the morphological dictionary.

There are many ways how to examine compound adverbs. We decided to make a corpus probe to identify compound adverbs tagged as an unrecognized part of speech in Czech National Corpus SYN v3 corpus[1] [6]. We have chosen unrecognized compound adverbs because they will likely have the same characteristics as those recognized, and we will thus have the data to add into the morphological dictionary. The obtained data were sorted out manually and grouped by their prefix: *do-*, *k-/ku-*, *mezi-*, *na-*, *nad-*, *o-*, *ob-*, *od-*, *po-*, *pro-*, *před-*, *při-*, *s-/sou-*, *u-*, *v-*, *z-*, *zpod-*, *za-* and, consequently, by their ending, because every prefix (previously preposition) can have more than one word ending (e.g. *poanglicku, pořadě, pokrk, pošesté, poprvní*). Afterwards, we were interested whether the AMA recognizes expressions that we have found as a one-word unit with the tag [tag="X.*"] if we respread them into multiword expressions. And if the AMA recognizes them, what tag will it assign them with. So we searched in the corpus gradually for multiword expressions of one-word compound adverbs that we found while processing the first step.

2 Finding

Thanks to the chosen CQL queries,[2] we have obtained a relatively large set[3] of one-word expressions that have the same initial and ending strings as a possible compound adverb. By hand selection, we have identified 470 units that we thought could be compound adverbs. They were not recognized by AMA because they were not listed in the morphological dictionary. Many of the one-word compound adverbs (e.g. *kpředu, odposledka, zmísta, zšeda, předloni,*

[1] At the time the biggest availible corpus in Czech National Corpus.

[2] [tag="X.*" & lemma="po.*"], [tag="X.*" & lemma="do.*"], [tag="X.*" & lemma="k.*"], [tag="X.*" & lemma="ob.*"], [tag="X.*" & lemma="od.*"], [tag="X.*" & lemma="o[^ db].*"], [tag="X.*" & lemma="mezi.*"], [tag="X.*" & lemma="na.*"], [tag="X.*" & lemma="pro.*"], [tag="X.*" & lemma="před.*"], [tag="X.*" & lemma="při.*"], [tag="X.*" & lemma="s.*"], [tag="X.*" & lemma="u.*"], [tag="X.*" & lemma="v.*"], [tag="X.*" & lemma="za.*"], [tag="X.*" & lemma="z[^ a].*"]

[3] More than 30.000 units.

naven, nablint, ...) are recorded in existing dictionaries, so they are not only occasionalism.

Somewhat more complicated situations have been encountered in the case of a compound adverbs in the form of a multiword expressions. We noticed that most of the compound adverbs are recognized by automatic morphological analysis and, from the point of view of word formation, the multiword expression is tagged as a preposition and part of speech from which the compound adverb is formed. Most often they are nouns (e.g. *na mokro, k dobru, ob den, ...*), but we have also noted adjectives (e.g. *na jisto, do pevna, ...*), adverbs (e.g. *na knap, na krátce, na tajno, k stáru, ...*) or numerals (e.g. *ob dva, na vícekrát, po mnohokrát, ...*), pronouns (e.g. *po svých, ...*) and prepositions (e.g. *na podél, na prostřed, ...*). In rare cases, we have registered the preposition and the verbs (e.g *do leskla, k předu, na zrz, z nenadála*).[4]

We found interesting that most of the obtained expressions were a compound of preposition and nouns (nouns, adjectives, pronouns, numerals) in the singular (e.g. *naskok, dočervena, nadálku, ...*), but we also noticed the compound of the preposition and the noun in plural (e.g. *nahony, sdíky, počertech, odvěků*).

We have found many multiword compound adverbs in Idiomatic and phrasal dictionary (DEBDict) [7,8] (e.g. *na světlo, na slovo, nablint, po krk, po čertech, ...*) and some of the analyzed data have shown strong collocations (e.g. *zbarvit do bíla / dobíla; zaostřovat do blízka / doblízka; holení na mokro / namokro; rozválet / vyválet / nakrájet na tenko / natenko; být natuty / na tuty; ...*).

Tagging of multiword compound adverbs as a preposition and seven different part of speech is inconsistent. Especially when comparing multiword expressions tagging such as *na tvrdo* (POS=R, POS=A), *na žluto* (POS=R, POS=N), *na tajno* (POS=R, POS=D). However, this is understandable with respect to the tagset currently used for the SYN corpora series in Czech National Corpus. The currently used tagset does not contain any tag for the compound adverb or its part. We think this is inappropriate.

By analyzing, we found that not all prepositions taken into account in queries form part of compound adverbs, to four (*u, mezi, o, při*) no expression was found according to established criteria.

3 Suggestions

By analyzing the corpus data, we came up with three proposals that could improve the automatic tagging of compound adverbs. The first proposal is the addition to the morphological dictionary, the second is the change of tag, and the third is the addition of strong collocations into the Multiword Expressions Lexical Database.

[4] In Czech it is not possible to follow the verb after the preposition. In the case of an expression *k předu*, this is an error in the disambiguation, since both the POS=D and POS=V interpretations are attributed to the unit *předu*. In the case of *do leskla, na zrz, z nenadála* only the POS=V interpretation is in the morphological dictionary.

3.1 Addition into the Morphological Dictionary

We believe that one of the ways to improve automatic morphological analysis is to add data to the morphological dictionary. We have selected 470 units from the corpus probe, but not all of them are suitable for the morphological dictionary, for several reasons. Some expressions are not considered adverbial, because adverbialization has not occurred, there is only missing space when writing this expression (e.g *ožívot, narozloučenou, . . .*).

We also recorded expressions that are compound, but we do not consider them as an adverb (e.g. *doboha*: interjection). In one case the obtained form resembled compound adverb structure, but we came to the conclusion that it is a verb form (*zamražena*: verb). We have recorded expressions which we do not consider to be compound adverbs and are listed in existing dictionaries as another part of speech (*mezitímco/mezitím co*: conjunction, *naprostřed*: preposition). On the other hand, these units are not listed in morphological dictionary and may be added there as a different part of speech (not compound adverb).

Some expressions are compound adverbs, but we understand them more as occasionalism and they occur in the order of units (e.g. *narub, pokopě, vnedohlednu, . . .*). For this reason, we have set a minimum frequency of 15 occurrences in the corpus SYN v3 to add the word into the morphological dictionary. Otherwise, because 15 occurrences are no longer 0 i. p. m. but 0.01 i. p. m. The random check in the corpus SYN v6 showed that in many cases the occurrences of analyzed compound adverbs are very similar.

We propose to add into morphological dictionary those expressions that are demonstrably compound adverbs, the process of adverbialization is either completed or ongoing, and the occurrence frequency is greater than 15. Furthermore, we propose to add into the morphological dictionary the expressions we have found in existing dictionaries (DEBDict) [7], regardless of the frequency of occurrence and part of speech. We also propose to add those units with frequency higher than 15 which we identified as a different part of speech than adverb. The proposal for addition in the morphological dictionary always contained lemma and part of speech interpretation.

Altogether, 177 units were proposed for addition in the morphological dictionary, their number and the part of speech interpretation was as follows:

POS=D, SUB=s, compound adverb, 103 units, (e.g. *domodra*)
POS=O, SUB=s, oscillating, compound, 43 units, (e.g. *modro*)
POS=C, numeral, 20 units, (e.g. *našestkrát*)
POS=D, adverb, 4 units, (e.g. *tuty*)
POS=R, preposition, 2 units, (e.g.*naprostřed*)
POS=I, interjection, 1 unit, (*doboha*)
POS=J, conjunction, 1 unit, (*mezitím*)
POS=T, particle, 1 unit, (*naviděnou*)[5]

[5] We do not consider *naviděnou* as a compound particle. We proposed this unit to be add into the morpohological dictionary because its one-word form is common. Similar case is e.g. *nashledanou*, also tagged as POS=T.

POS=N, noun, 1 unit, (*podmíru*, lemma *podmíra*)
POS=V, verb, 1 unit, (*zamražena*, lemma *zamrazit*)
Number of proposed units is 177, number of analyzed units is 470.

3.2 Change of Morphological Tag

There are two facts which led us to suggest to change the morphological tag: First, there is no tag in the tagset of the Czech National Corpus [9] that indicates the compound adverb and second, there are many words, which we consider to be compound adverbs, that are marked inconsistently. Examples of inconsistent tagging:

na tvrdo: preposition, adjective
na žluto: preposition, noun
na tajno: preposition, adverb

We find a satisfactory solution in the concept of the NOVAMORF project [10], which proposes a new part of speech type: POS=O: an oscillating part of speech. For POS=O, we consider those forms that are ambiguous whether they are nouns, adjectives, or adverbs (e.g. *sucho, mokro, modro, ...*). We also propose to add a subset of the SUB=s meaning compound to adverbs and numerals.

We suggest therefore to tag one-word compound adverbs as POS=D with specifying a type compound as SUB=s (e.g. *namodro*: POS=D, SUB=s). We propose to tag multiword expressions of type *na modro* as *na* POS=R, *modro* POS=O, SUB=s.

In connection with the introduction of a new tag for a compound word, the question arises as to whether this addition should be added to all the part of speech in which the compound word can occur. These would be adverbs, numerals, as well as interjections, prepositions or conjunctions. With a view to the consistency of tagging, we think the adding the tag for a compound is useful, but only for adverbs and numerals. Compound interjections (e.g. *proboha, doboha, ...*), prepositions (e.g. *naprostřed*) or conjunction (e.g. *mezitím*) are very few.

3.3 Collocations

By analyzing data, we have found that some compound adverbs are found in collocations, some of which are part of phrases and idioms, and are recorded in the Idiomatic and phrasal dictionary (DEBDict) [7,8]. Nowadays the Multiword Expression Lexical Database (MWELD) is built by Petkevič et al. [11] and we find very useful to enlarge this database with our data. Larger the MWELD is, better results in disambiguation can be reached.

4 Conclusion

We have focused on compound adverbs from the automatic morphological analysis point of view. Compound adverb tagging is a non-trivial problem

because compound adverbs are written mostly together as one word, but often in parallel there exists a multiword expression and their meaning is the same (e.g. *na příklad – například*).

We made a corpus probe on corpus SYN v3 and we searched for unrecognized forms that can be considered as being compound adverbs. Thanks to CQL queries, we have obtained large data. We have sorted it manually according to prefix and then by ending. We selected 470 units we consider as compound adverbs. Afterward, we were interested whether the AMA recognizes these expressions if we respread it into multiword expressions. Both one-word and multiword expressions were checked in existing Czech dictionaries. We also focused on strong collocations of chosen units.

By analyzing the corpus data we suggest three solutions to improve the compound adverbs tagging: First is to enlarge morphological dictionary by adding units which are demonstrably compound adverbs and which frequency of occurrence is more than 15 in corpus SYN v3. Altogether we have found 103 units to be added into the morphological dictionary as a compound adverb and others 74 units as others part of speech. Second, we propose in accordance with NOVAMORF project a new compound adverb tagging. We propound a new type of part of speech such as POS=O, oscillating part of speech, and also new subset SUB=s, means compound. We suggest tagging subset compound type not only with adverbs but also with numerals.

We are aware that the proposed solutions do not cover the complete issue of compound adverb recognition, but we believe that the corpus probe and the proposed solutions can contribute to an at least partial improvement of the AMA in this area.

References

1. Internetová jazyková příručka, `http://prirucka.ujc.cas.cz/?id=130`
2. Dokulil, M. Tvoření slov v češtině, díl 1 Teorie odvozování. pp. 22. Nakladatelství Československé akademie věd, Praha (1962)
3. Cvrček, V. Mluvnice současné češtiny. Karolinum, Praha (2010)
4. Multiword Expressions. In: Wiki of the Association for Computational Linguistics. Association for Computational Linguistics, Stroudsburg PA, USA (2016).
5. Petkevič, V. Problémy automatické morfologické disambiguace češtiny. Naše řeč, 97(4/5), pp. 194–207 (2014)
6. Křen, M., Čermák, F., Hlaváčová, J., Hnátková, M., Jelínek, T., Kocek, J., Kopřivová, M., Novotná, R., Petkevič, V., Procházka, P., Schmiedtová V., Skoumalová, H., Šulc, M. Korpus SYN, verze 3 z 27. 1. 2014. Ústav Českého národního korpusu FF UK, Praha (2014)
7. Horák, A., Pala, K., Rambousek, A., Povolný, M. DEBVisDic – First Version of New Client-Server Wordnet Browsing and Editing Tool. In: Proceedings of the Third International WordNet Conference – GWC 2006. pp. 325–328. Masaryk University, Brno (2006)
8. Čermák, F. Slovník české frazeologie a idiomatiky. Leda, Praha (2009)
9. Hajič, J., Cvrček, V., Chlumská, L. Morfologické značky (tagy) In: Wiki Český národní korpus. FF UK ÚČNK, Praha (2017)

10. Osolsobě, K., et al. Nová automatická morfologická analýza češtiny. Naše řeč, 100(4), pp. 225–234 (2017)
11. Petkevič, V. et al. Lexicon and Multiword expressions. In: Slavicorp2018 Book of abstracts. FF UK, Praha (2018)

csTenTen17, a Recent Czech Web Corpus

Vít Suchomel[1,2]

[1] Lexical Computing
[2] NLP Centre, Masaryk University, Brno
xsuchom2@fi.muni.cz

Abstract. This article introduces a very large Czech text corpus for language research – *csTenTen17* compiled from texts downloaded in 2015, 2016 and 2017. The corpus is consisting of 10.5 billion words reaching double the size of its predecessor from 2012. A brief comparison with other recent Czech corpora follows.

Keywords: Czech corpus; web corpus; text processing

1 Introduction

Algorithms in the field of natural language processing generally benefit from large language models. Many words and phrases occur rarely, therefore there is a need for very large text colletions to research the behaviour of words. [10]. Furthermore, the quality of the data obtained from the web is also important. [13] Linguists studying natural languages, lexicographers compiling dictionaries, sociologists studying the topics moving the society, marketing experts creating brand names, language engineers building language models and many others are turning to the web as a source of language data. Nowadays, the web is the biggest, easily exploitable and the cheapest source of text data.

We decided to support corpora based research of Czech language again by building an up-to-date corpus from web documents in Czech. The aim was to apply text cleaning software, language discrimination tools, and deduplication to a corpus of a ten billion words size. The corpus should be indexed in a corpus manager providing a basic concordance search as well as advanced functions such as a summary of grammatical and collocational behaviour of words.

1.1 Paper Outline

Corpus construction and properties are described in Section 2. The result corpus is compared to other Czech corpora in Section 3. Final remarks are presented in Section 4.

Aleš Horák, Pavel Rychlý, Adam Rambousek (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2018, pp. 111–123, 2018.

2 Corpus Construction And Properties

2.1 Crawling The Czech Web

The corpus consists of texts obtained using crawler SpiderLing [14]. The crawler collected texts from the web in October and November 2015, October and November 2016, and May, October and November 2017. The crawler started from seed web domains and URLs coming from various sources:

- csTenTen12 document sources (the previous Czech web corpus),
- lists of web domains presenting a good quality content such as `dmoz.org` and `urlblacklist.com`[3],
- URLs of Czech documents obtained by querying search engine Bing for Czech words,
- manually selected Czech web news sites (`blisty.cz`, `ihned.cz`, `lidovky.cz`, `novinky.cz`, `reflex.cz`, `seznam.cz`).

The crawler was not restricted to download just from the Czech national top level domain `.cz`. It was set not to crawl web sites not providing Czech text and to slightly prefer web sites yielding more Czech text than other domains.

Data processing tasks important for the crawler to evaluate the yield rate of Czech text of web sites were carried out by tools embedded within the crawler:

- Encoding detection using byte trigram models by Chared[4] [11],
- language identification on the document level using character trigram models,
- HTML boilerplate removal by Justext 1.4[5] [9],
- splitting text to paragraphs by Justext using HTML tags `<p>`, `<div>` and `<br>`,
- language checking on the paragraph level using lists of frequent words by Justext,
- exact duplicate removal on the document level using hashes of HTML data and plain text.

All models necessary for the process were built using samples of Czech text or web pages from the Czech web before starting the crawler.

The following sizes apply just to the 2017 batch: The crawler made 590 million HTTP requests to internet servers. 7.0 TB of raw HTTP response data containing 150 million web pages were collected. Of these, 35 million web pages contained at least one paragraph of Czech text recognised by Justext[6]. The size of the plain text obtained by the crawler before additional filtering described in Section 2.3 was 60 GB.

[3] Both of these web domain catalogues are no longer available on the web in 2018.

[4] `http://corpus.tools/wiki/Chared`

[5] `http://corpus.tools/wiki/Justext`

[6] The following Justext parameters were used to recognise paragraphs of text long enough: `length_low = 70`, `length_high = 140` (200 by default), `stopwords_low = 0.2` (0.3 by default), `stopwords_high = 0.3` (0.32 by default), `max_link_density = 0.4` (0.2 by default). The default values were altered to allow slightly shorter paragraphs to extract more text while keeping the level of strictness high.

2.2 Collecting Texts From Wikipedia

Since Wikipedia pages share the structure of a document and they are coded in MediaWiki markup language[7] which is not straightforward to turn into a plain text, software Wiki2Corpus[8] was used to obtain texts from the Czech Wikipedia for the corpus. The tool ran in November 2017 and aimed for both encyclopedia articles and respective talk pages[9].

642,693 Wikipedia pages or 15 GB of data were downloaded by Wiki2Corpus (including talks, redirections, disambiguation pages) and converted from the MediaWiki markup to documents consisting of plain text. The size of the data after extracting paragraphs using Justext was 1.0 GB. Most stubs or other short articles were discarded because they were lacking nice long paragraphs recognised by Justext.

The plain text was further cleaned and filtered using the same methods as the crawled web pages. The process is described in the following section.

2.3 Postprocessing of Text

Methods of postprocessing of corpus plain text after the crawling applied to all parts of the corpus are described in this section. The sizes however represent only the part of data collected in 2017 since the information about processing the parts from 2015 and 2016 are no longer available.

The plain text was split to tokens using Unitok [15]. The size of the 2017 data at this stage of processing was 7.8 billion tokens.

Despite the character n-gram model based removal of documents in other than the target language, there were still a lot of paragraphs in unwanted languages (i.e. other than Czech, especially English and Slovak). Language seperation based on a method exploiting large lists of word forms with relative corpus frequency[10] in large monolingual web corpora[11] described in [3] was applied to paragraphs and documents of the tokenised text.

Czech, Czech without diacritics, Slovak, Slovak without diacritics, English, German, Polish, Slovene, Croatian, Russian, French, Spanish, and Italian were discerned. Only the Czech part (with diacritic marks) was allowed to get to the final corpus. 0.1 % of paragraphs were filtered out because the majority of the content was not in Czech, 1.0 % of paragraphs were thrown away because of a content in multiple languages, and 3.6 % of paragraphs were filtered out since they were too small to reliably determine a language (in fact, these paragraphs

[7] `https://www.mediawiki.org/wiki/Markup_spec`

[8] `http://corpus.tools/wiki/wiki2corpus`

[9] E.g. `https://cs.wikipedia.org/wiki/1984_(roman)` and its talk page `https://cs.wikipedia.org/wiki/Talk:1984_(roman)`

[10] Relative corpus frequency is the number of occurrences of a word form per billion tokens in the corpus.

[11] Web corpora built in the past were used. In case there was no corpus in the target language, the list would ben obtained by bootstrapping, i.e. applying the same method several times to the corpus until the result frequency list stops changing.

would not contribute much to the quality of the corpus even though they were in Czech). 4.7 % of paragraphs were removed in total in this step of postprocessing the data.

Examples of paragraphs of text removed because of the relative frequency of word forms is larger in a reference web corpus of other language than in csTenTen12 follow (non-Czech words are striked out). Example 1[12]: *23 ~~Solo Pieces for La Naissance de L'Amour~~ je soundtracková album velšského multiinstrumentalisty Johna Calea. Album vyšlo v roce 1993 u vydavatelství ~~Les Disques du Crépuscule~~. Album produkoval ~~Jean-Michel Reusser~~.* Example 2[13]: *~~Nice~~ hotel ~~at~~ a ~~good location. Rooms very good, but beds~~ a ~~little bit hard. The staff was nice and helpful. Nice location close~~ to ~~Konakli center with lot of shops and market~~ on ~~Wednesdays. Nice~~... celá recenze s možností překladu.*

Near-duplicate paragraph deduplication was carried out using Onion[14] [9], a tool based on comparing hashes of n-grams of tokens. In the case of this corpus, paragraphs containing more than 90 % of 5-tuples of tokens seen before (i.e. in a part of the input read earlier) were removed. The smoothing mode was on with the minimum length of a stub set reduced to 10 tokens.[15]

The text was split to sentences using a tool looking for fullstops (or other end of sentence markers) followed by a space and a capital letter and dealing with abbreviations according to a predefined list.

2.4 Morphological Annotation

The corpus was lemmatised and morphologically annotated using Czech morphological analyzer Majka [17]. The analyser determined the part of speech and other grammatical categories (where applicable): gender, number, case, aspect, modality and other.[16] The tags were desambiguated by Desamb [12,4]. A gender respecting lemma was added to allow creating name phrases from lemmas properly.[17]

The most frequent parts of speech identified in the corpus are nouns (33 %), verbs (16 %), adjectives (12 %), prepositions (10 %), pronouns (9 %), and adverbs (7 %).[18]

[12] Text source: `https://cs.wikipedia.org/wiki/23_Solo_Pieces_for_La_Naissance_de_L'Amour`

[13] Text source: `https://www.ellagris.cz/turecko/turecka-riviera/alanya/royal-garden-select-626634`

[14] http://corpus.tools/wiki/Onion

[15] The full parameters: `onion -s -n 5 -t 0.9 -l 10`. More about tuning the parameters of onion can be found in a paper by V. Benko [1].

[16] See `https://www.sketchengine.co.uk/tagset-reference-for-czech` for the full tagset reference.

[17] For example, the base form of "veřejné knihovně" is not "veřejný knihovna" where "veřejný" (masculine) is the lemma of "veřejná" (feminine) but "veřejná knihovna" where the gender of the noun is respected by the adjective properly.

[18] Not counting the punctuation, abbreviations, foreign words.

Table 1: Sizes of parts of csTenTen17 by the source subcorpus

	Total	Wiki 17	Wiki Talk	Web 17	Web 16	Web 15
Tokens	12,586,415,546	0.97 %	0.09 %	58 %	32 %	8.4 %
Words	10,502,222,474					
Sentences	738,085,256					
Paragraphs	227,097,470					
Documents	35,995,251	1.00 %	0.09 %	62 %	30 %	7.6 %

Table 2: Lexicon sizes

Word form	Lemma	Gender respecting lemma	Tag	Part of speech
40,445,706	29,100,249	34,014,060	2,247	15

2.5 Final Sizes

The final corpus consists of 12,586,415,546 tokens in 35,995,251 documents. 91 % of tokens of the final corpus come from the Czech TLD .cz. Sizes of parts of the corpus by the source can be found in Table 1. Sizes of lexicons are in Table 2. Document counts in TLDs and web sites are presented by Table 3.

Table 3: Document count – the largest web domains and domain size distribution

TLDs		Web domains		Web domain size distribution	
cz	91 %	webnode.cz	660,000	At least 1 document	350,000
com	2.3 %	idnes.cz	540,000	At least 5 documents	190,000
eu	1.9 %	blogspot.cz	450,000	At least 10 documents	130,000
org	1.8 %	wikipedia.org	390,000	At least 50 documents	53,000
net	1.2 %	lidovky.cz	180,000	At least 100 documents	34,000
info	1.0 %	zive.cz	170,000	At least 500 documents	9,100
		tyden.cz	150,000	At least 1,000 documents	4,900
		estranky.cz	130,000	At least 5,000 documents	950
		e15.cz	120,000	At least 10,000 documents	460
		denik.cz	120,000	At least 50,000 documents	38
		tiscali.cz	120,000	At least 100,000 documents	13
		sluzby.cz	110,000	At least 500,000 documents	2
		rozhlas.cz	110,000		
		mobilmania.cz	100,000		
		penize.cz	98,000		
		ihned.cz	97,000		

2.6 Access To The Corpus

Since the corpus is a part of the HaBiT project[19] [8], it can be accessed via corpus manager Sketch Engine [6] at the project site.[20]. Functionality provided by Sketch Engine covers concordance search, wordlist search, collocation and word frequency calculation, Word Sketches, thesaurus and more.

3 Comparison With Other Recent Corpora

Our older paper on a Czech web corpus from 2012 is followed in this section. [16] There are the following recent Czech corpora used in the comparison:

- csTenTen17 – the new corpus,
- czTenTen12 (v. 9) – the previous version of csTenTen from 2012,
- Araneum Bohemicum III Maius (17.04, v. 1.3.61) – web corpus downloaded by V. Benko from 2013 to 2016. Crawled and processed by similar tools as in the case of TenTen corpora. [2]
- csSkELL (v. 2.2) – Czech web corpus of example sentences gained from websites provided by Czech WebArchive to 2016. Processed by similar tools as TenTen corpora.[21]
- SYN 2015 – Czech national corpus, a reference representative corpus containing fiction, non-fiction and journalism texts mostly from 2010 to 2014.[22] [7] This corpus is a non-web ballanced and representative corpus to compare less controlled web corpora to.[23]

3.1 Basic Properties

Tables 4 and 5 display values of six metrics calculated for the compared corpora. We observe the largest corpus has the largest dictionary and the least varied vocbulary.

Documents in csTenTen17 are shorter than in its predecessor. That might be caused by a similar composition of genres in the web, e.g. not much fiction that tends to contain long documents. The length of sentences is quite similar for all selected corpora.

csTenTen17 may be the corpus least contaminated by foreign text. That can be explained by an additional method of removing unwanted languages described in Section 2.3.

[19] `https://habit-project.eu/`

[20] `https://corpora.fi.muni.cz/habit/run.cgi/first?corpname=cstenten17_mj2`

[21] `https://www.sketchengine.co.uk/cskell/`

[22] `https://www.korpus.cz/`

[23] Although the full text of the corpus is not publicly available, a wordlist with frequencies was enough to carry out wordlist based measurements.

Table 4: Basic comparison of corpora: Token counts and type-token ratio. The higher TTR, the more varied vocabulary.

Corpus	Token count	Word lexicon	Type-token ratio
csTenTen17	12,600,000,000	40,400,000	0.003,2
czTenTen12	5,070,000,000	18,700,000	0.003,7
Araneum Bohemicum	1,200,000,000	8,460,000	0.007,1
csSkELL	1,730,000,000	8,010,000	0.004,6

Table 5: Basic comparison of corpora: Average document length (the number of tokens) (structure `<text>` used in the case of SYN 2015), average sentence length, "the-score". The-score, being the rank of word "the" in a list of lowercased words, is a very simple metric offering a basic idea about contamination of the corpus by foreign (English) text. The higher the value, the better.

Corpus	Avg. doc tokens	Avg. sentence tokens	The-score
csTenTen17	350	17	7,387
czTenTen12	550	18	730
Araneum Bohemicum	460	17	517
csSkELL	N/A	19	475
SYN 2015	1,055	15	1,145

3.2 Keyword Comparison

Keyword comparison as a way of telling differences between corpora was performed by Kilgarriff in [5]. Using the same method – putting csTenTen17 as the focus corpus and other corpora in the place of the reference corpus – the words with the highest relative frequency in comparison to words in other corpus or subcorpus are the highest ranked by the keyword score:

$$keywordscore = \frac{fpm_{foc}(w) + n}{fpm_{ref}(w) + n}$$

where $fpm(w)$ represents occurrences per million of word w, foc is the focus corpus, ref is the reference corpus, and n is a smoothing parameter.

Table 6 shows differences in the content of csTenTen17 in comparison to other corpora. It can be observed the new corpus covers topics trending recently such as "babiš", "eet", "trump", "sýrii", "krymu", "instagram", "severus", "snape", "naruto", "parlamentnílisty". (The last might be a tokenisation error as well.) There is also a lot of finance and trade related material in the 2017 corpus, e.g. "půjčka", "půjčky", "nebankovní", "směnnost", "prodám", "skladem". These words may indicate the presence of non-text in the corpus that should be investigated (short phrases without subject predicate pairs, or even computer

Table 6: Keyword comparison of csTenTen17, the 2017 subcorpus, to other corpora. Settings: lowercased word forms, minimum frequency 10, smoothing parameter 1 preferring rare words over common words.

	csTenTen12		Araneum Bohemicum		csSkELL	
Rank	**Word**	**Score**	**Word**	**Score**	**Word**	**Score**
1	pujcka	16.9	odst	56.8	č	49.4
2	babiš	14.2	písm	21.3	půjčka	15.4
3	pujcky	13.3	č	21.0	vč	14.4
4	půjčka	9.6	vč	12.3	prodám	13.8
5	trump	9.2	hellip	8.2	pujcka	13.6
6	babiše	8.7	tis	8.2	pujcky	10.9
7	eet	8.2	zák	7.7	nbsp	8.1
8	č	7.4	obr	7.5	hellip	7.7
9	trumpa	6.3	mld	6.9	naruto	7.7
10	azure	6.2	hl	6.4	kdyz	7.6
11	zadavatel	5.2	ust	5.7	ú	7.1
12	ú	4.9	okr	5.6	skladem	6.9
13	gmbh	4.8	naruto	4.9	nabízíme	6.9
14	sýrii	4.6	atp	4.8	severus	6.7
15	dodavatelský	4.4	ú	4.5	panička	6.5
16	zadavatele	4.4	parlamentnílisty	4.4	snape	6.4
17	nebankovní	4.3	azure	4.3	nebankovní	6.3
18	půjčky	4.1	dodavatelský	4.1	kontaktujte	6.2
19	krymu	4.1	protoe	4.0	koupelna	6.0
20	směnnost	4.1	oponent	4.0	pleť	5.8
21	instagram	4.1	xvi	3.9	půjčky	5.7
22	vyžádejte	4.0	ev	3.7	pred	5.7

generated text). There is also a lot of differences in tokenisation, especially in the case of Araneum: “odst”, “obr”, “vč” (these abbreviations were not recognised in our corpus). Word “protoe” may be a misspelling. Finally, some words without diacitics scored high, e.g. “pujcka”, “kdyz”, “pred”.

Keyword comparison to a non-web representative ballanced corpus shown in Table 7 reveals the new corpus contains relatively a lot of money lending text and also some internet related technical words.

Table 8 shows the 2016 subcorpus is polluted with an online gambling related spam.

3.3 Word Sketches

Multi Word Sketch for “chytat stéblo” (“to grasp a straw”, usually found in idiom “tonoucí se stébla chytá” – “grasping at straws”) in csTenTen17 and czTenTen12 are displayed in screenshots from Sketch Engine in Figures 1 and 2. As can be

Table 7: Keyword comparison of csTenTen17, the 2017 subcorpus, to SYN 2015 with the same settings as in Table 6. Lines affected by a different tokenisation, or misspellings were omitted to focus on differences in text type and genre.

	SYN 2015	
Rank	**Word**	**Score**
3	půjčka	27.1
7	prodám	17.3
8	nabízíme	16.5
11	nebankovní	15.4
13	klikněte	14.4
16	kontaktujte	13.0
20	skladem	11.6
21	naruto	11.6
23	půjčky	10.8
26	email	9.7
27	ikdyž	9.6
28	neváhejte	9.4
29	zadavatel	9.4
30	php	9.3
31	html	9.2
32	ahojky	9.2
33	online	9.2
35	trump	9.1

Table 8: Keyword comparison of the 2016 subcorpus to the 2017 part of csTenTen17 with the same settings as in Table 6.

	csTenTen17	
Rank	**Word**	**Score**
1	kasino	1479
2	sloty	1299
3	casino	969
4	automaty	708
5	kasina	649
6	kasinu	471
7	kasinové	463
8	hazardní	443
9	sajid	432
10	blackjack	422
11	jackpoty	408
12	jackpot	400
13	roztočení	385
14	lisu	309
15	beste	305
16	slot	301
17	činohra	301
18	zelenom	294
19	sizzling	267
20	karolínka	246

seen, the bigger corpus provides more collocations to study the meaning of the phrase. For example, "chytat stéblo záchrany" ("to grasp a straw of rescue") can be found only in a single case in the 2012 version of the corpus while there are four occurrences of the phrase in the new data.

"Chytat stéblo" is located in csSkELL, the smallest corpus in the comparison, only five times which is not enough to get relevant information about the phrase. Word Sketches of Araneum Bohemicum are not compared since the corpus is tagged by another tagger and its Word Sketches are based on a different grammar.

3.4 Thesaurus

According to our inspection of a computer generated thesaurus based on words sharing the same collocations in relations in Word Sketches, the size of a corpus contributes to finding better synonym candidates for low

Fig. 1: Multi Word Sketch for "chytat stéblo" ("to grasp a straw") in csTenTen17. Collocations occurring at least three times are displayed in several grammatical relations. The number of "chytat" collocating with "stéblo" in the corpus is 903.

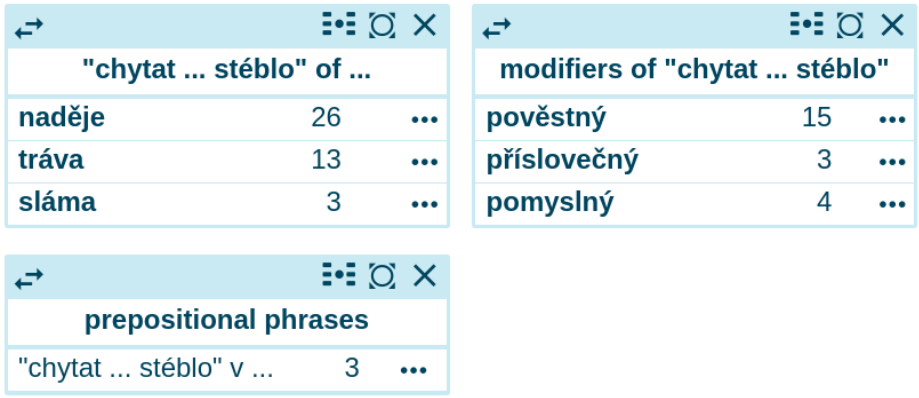


Fig. 2: Multi Word Sketch for "chytat stéblo" ("to grasp a straw") in csTenTen12. Collocations occurring at least three times are displayed in several grammatical relations. The number of "chytat" collocating with "stéblo" in the corpus is 506.

	CsTenTen17	Frequency	Cluster
1	prehistorický	17,885	pravěký 39,218
2	obstarožní	5,849	
3	rozhrkaný	1,235	otřískaný 2,208
4	lidožravý	2,248	
5	humanoidní	5,611	
6	ohyzdný	4,195	šeredný 4,196

	CzTenTen12	Frequency	Cluster
1	prehistorický	8,754	pravěký 18,423
2	humanoidní	2,994	
3	lochneský	183	lochnesský 143 lochnesské 57
4	obstarožní	3,035	
5	porouchaný	4,325	
6	druhohorní	1,851	třetihorní 2,067

	CsSkELL	Frequency	Cluster
1	prehistorický	3,220	pravěký 7,574
2	druhohorní	1,003	třetihorní 1,196
3	vyhynulý	2,256	vymřelý 1,047
4	potopní	67	
5	mýtický	3,216	mytický 2,558 mytologický 3,453
6	lochneský	34	Lochnesský 61 Lochneský 25

Fig. 3: Thesaurus of word “předpotopní” (“antediluvian”, “prehistoric”) based on words sharing the same collocations in relations in Word Sketches in three corpora. Note both csTenTen17 and czTenTen12 provide candidates meaning old, battered, chipped (“obstarožní”, “rozhrkaný”, “otřískaný”) while these important synonyms were not extracted from smaller csSkELL.

frequency words. For example, there are better results for adjective "předpotopní" ("antediluvian", "prehistoric") extracted from csTenTen17 (12 bn. tokens) and czTenTen12 (5.1 bn. tokens) than from csSkELL (1.7 bn. tokens) as can be seen on Figure 3.

4 Conclusion and Future Work

A new ten-billion-word Czech corpus was built from documents recently published on the web. The corpus can be searched by a publicly accessible corpus manager.

To focus on quality of the data, which is important for all kinds of corpus use, we would like to correct the tokenisation of abbreviations and to address the part of the corpus from 2016 containing online gambling advertisement spam. Furthermore, the users of the corpus would benefit from identification of topics and genres of documents. That will be another field to focus on in the future.

Acknowledgements. This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin infrastructure LM2015071.

References

1. Benko, V.: Data deduplication in slovak corpora. Slovko 2013: Natural Language Processing, Corpus Linguistics, E-learning pp. 27–39 (2013)
2. Benko, V.: Aranea: Yet another family of (comparable) web corpora. In: International Conference on Text, Speech, and Dialogue. pp. 247–256. Springer (2014)
3. Herman, O., Suchomel, V., Baisa, V., Rychlý, P.: Dsl shared task 2016: Perfect is the enemy of good language discrimination through expectation-maximization and chunk-based language model. In: Nakov, P., Zampieri, M., Tan, L., Ljubešić, N., Tiedemann, J., Malmasi, S. (eds.) Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3). pp. 114–118. Association for Natural Language Processing (ANLP) (2016), https://aclanthology.info/pdf/W/W16/W16-4815.pdf
4. Jakubíček, M., Horák, A., Kovář, V.: Mining phrases from syntactic analysis. In: Lecture Notes in Artificial Intelligence, Proceedings of Text, Speech and Dialogue 2009. pp. 124–130. Springer-Verlag, Plzeň, Czech Republic (2009)
5. Kilgarriff, A.: Getting to know your corpus. In: Text, Speech and Dialogue. pp. 3–15. Springer (2012)
6. Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P., Suchomel, V.: The sketch engine: ten years on. Lexicography **1** (2014), http://dx.doi.org/10.1007/s40607-014-0009-9
7. Křen, M., Cvrček, V., Čapka, T., Čermáková, A., Hnátková, M., Chlumská, L., Jelínek, T., Kováříková, D., Petkevič, V., Procházka, P., et al.: Syn2015: representative corpus of contemporary written czech. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation, LREC 2016. pp. 2522–2528 (2016)
8. Pala, K., Horák, A., Rychlý, P., Suchomel, V., Baisa, V., Jakubíček, M., Kovář, V., Nevěřilová, Z., Rambousek, A., Gambäck, B., Sikdar, U., Bungum, L.: Habit system (2017), http://corpora.fi.muni.cz/habit/

9. Pomikálek, J.: Removing Boilerplate and Duplicate Content from Web Corpora. Ph.D. thesis, Masaryk University, Brno (2011)
10. Pomikálek, J., Rychlý, P., Kilgarriff, A.: Scaling to billion-plus word corpora. Advances in Computational Linguistics **41**, 3–13 (2009)
11. Pomikálek, J., Suchomel, V.: chared: Character encoding detection with a known language. In: Aleš Horák, P.R. (ed.) Fifth Workshop on Recent Advances in Slavonic Natural Language Processing. pp. 125–129. Tribun EU, Brno, Czech Republic (2011)
12. Šmerk, P.: Unsupervised Learning of Rules for Morphological Disambiguation. In: Lecture Notes in Artificial Intelligence 3206, Proceedings of Text, Speech and Dialogue 2004. pp. 211–216. Springer-Verlag, Berlin (2004)
13. Spoustová, J., Spousta, M.: A high-quality web corpus of czech. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12). European Language Resources Association (ELRA), Istanbul, Turkey (may 2012)
14. Suchomel, V., Pomikálek, J.: Efficient web crawling for large text corpora. In: Proceedings of the Seventh Web as Corpus Workshop. Lyon, France (2012)
15. Suchomel, V., Michelfeit, J., Pomikálek, J.: Text tokenisation using unitok. In: Eighth Workshop on Recent Advances in Slavonic Natural Language Processing. pp. 71–75. Tribun EU, Brno (2014)
16. Suchomel, V.: Recent czech web corpora. In: Aleš Horák, P.R. (ed.) 6th Workshop on Recent Advances in Slavonic Natural Language Processing. pp. 77–83. Tribun EU (2012)
17. Šmerk, P., Rychlý, P.: Majka – rychlý morfologický analyzátor. Tech. rep., Masarykova univerzita (2009), `http://nlp.fi.muni.cz/ma/`

An Update of the Manually Annotated Amharic Corpus

Pavel Rychlý[1] and Gezahegn Tsegaye Lemma[2]

[1] Faculty of Informatics
Masaryk University
Brno, Czech Republic
[2] University of Calabria
Cosenza, Rende, Italy
pary@fi.muni.cz

Abstract. The paper describes a an update of the manually annotated Amharic corpus WIC 2.0. It lists the problems of the previous version of the corpus and shows that even small changes in the corpus annotation could lead to a higher quality of trained part-of-speech taggers.

Key words: text corpus; Amharic corpus; part-of-speech tagging

1 Introduction

Amharic language has tens of million native speakers but, in the corpus linguistics, it is one of under-resourced languages. There are not many corpora and language tools for Amharic, but there is a steady progress in creating both language data and tools.

One of very basic tools for natural language processing is a part of speech (PoS) tagger. PoS taggers assign a PoS tag for each word from an input. They usually learn a language model (or a set of rules) using manually annotated corpora. It is very hard to build a tagger without at least small annotated corpus. One of such approaches is described in [5]. Most taggers requires hundreds thousand of tokens for a reliable processing. Building a corpus of that size is expensive in the amount of human work.

In this respect, Amharic language is in a good possition, there is the Walta Info Corpus (WIC). It consists of about 210,000 words in 1,065 documents. Texts were taken from the Web news published by the Walta Information Center (www.waltainfo.com) in 2001.

There were several attempts to use the WIC Corpus for training automatic part-of-speech taggers, for example [1,8,2]. All of them found that the corpus has many annotation inconsistencies: missing tags, misspelling of tags, multiword expressions and others. Somewhat cleaned version was described in [6] and it was published in the Clarin repository [4].

Aleš Horák, Pavel Rychlý, Adam Rambousek (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2018, pp. 124–128, 2018.

Table 1. PoS tag frequency of the two most ambiguous words in the WIC Corpus

ያህል (*about*)		በተለይም (*specifically*)	
ADJ	5	ADJ	5
ADJC	1	ADV	15
ADJP	3	N	11
ADV	141	NC	3
CONJ	3	NP	1
N	1	PREP	2
NC	4	PRON	1
NP	24	UNC	37
NPC	4	V	48
UNC	25	VP	2
VPC	1	VREL	2

2 WIC tag set

Amharic language has a rich morphology: Nouns and adjectives are inflected and there are complex rules for deriving verbs. Several part-of-speech tag systems were proposed earlier, all working with about 10 tags for basic part of speech. In some cases, nouns, pronouns, adjectives, verbs and numerals have variants of words with attached prepositions and/or conjunctions. For example, nouns (tag N) could be combined with a preposition as prefix (tag NP), with a conjunction as suffix (tag NC), or with a preposition as prefix and a conjunction as suffix (tag NPC). In total, there are 30 different PoS tags in the WIC Corpus.

2.1 PoS tag ambiguity

The WIC Corpus contains very high ambiguity on word form level. There are almost 34,000 types (different word forms including numbers and punctuation marks), almost 20,000 out of them are hapax legomena (occuring only ones). There are more than 4600 types with at least two different PoS tag. The most ambiguous words are ያህል (*about*) and በተለይም (*specifically*) both with 11 different PoS tags. The list of all tags with respective counts in the corpus are listed in Table 1.

Another examples of an ambiguous type is number *10*. In the original version of the corpus, it has 5 alternative PoS tags (each with only one occurrence) in addition to the correct one *NUMCR* (cardinal number). We can guess that many of these PoS tags are plain errors in the annatation. They are not surprising if we consider the annotation process during the corpus building. Annotators wrote the tags on a paper and they were later transcribed into the electronic form.

3 Error correction

During our work with the coprus, we have identified ambiguous words and tried to verify PoS tags for them. We have listed 68 words resulting in more than 200 combinations of word-tag. These combinations covers 5000 tokens, we have checked substantial part of all occurrences for each word-tag combination.

We have identified 139 word-tag combination where all hits in the corpus are errors. All the errors were corrected, there are more than 2,300 tokens affected.

In the majority of studied words, most of the occurences of the given word are correct. For exmaple, word ሁለት (*two*) has 150 hits in the corpus, 139 correct and 11 incorrect (with 3 different PoS tags). On the other hand, word ለመስራት (*to work, to make*) has 40 hits, 34 incorrect (PoS tag *NP*) and 6 correct (PoS tag *VP*).

We estimate that there could be about 10 % of errors in the PoS tag annotation.

4 Evaluation

The new version of the corpus is different in only a bit more than 1 % of tokens. One can think that this number is too small to have any effect on the PoS taggers trainded on the corpus. The more technical changes in our first release of the corpus have very limited efect, and it changed much more tokens. On the other hand the changes described in this paper affect several high frequent words.

To meassure the efect of the changes, we have done the same evaluation process as during the previous release of the corpus. We have trained two different PoS taggers and evaluated the accuracy using 10-fold cross validation. We have divided the corpus into 10 parts each containing 20,000 tokens. For each part, we trained a tagger on nine remaining parts, ran the tagger on that part, and compared the result with the manual annotation. The whole evaluation task was done on the Fidel part of the corpus (the Ethiopian script). The evaluation was done before and after the proposed changes.

4.1 TreeTagger

TreeTagger [7] works well for tag-sets with a small number of tags. Both training and tagging is quite fast. The results are listed in Table 2.

We can see that the average accuracy is higher 0.5 percentage points. That is small but significant.

4.2 APtagger

APtagger [3] is a fast and accurate part-of-speech tagger based on the Averaged Perceptron. As neural motivation suggest, the tagger uses several random passes through the training data to learn the model. Each run is a bit different with different model parameters. We have used 10 iterations in training and the resulting accuracy differs in only fraction of percentage point between runs.

The respective results of APtagger are listed in Table 3.

Table 2. Accuracy of TreeTagger on ten parts of the WIC Corpus

Part	befor changes	after changes
1	85.1	85.8
2	85.4	86.0
3	85.7	86.3
4	88.2	88.5
5	89.2	89.8
6	86.8	87.3
7	89.9	90.4
8	91.6	91.7
9	89.8	90.3
10	82.3	83.3
Average	87.4	87.9

Table 3. Accuracy of APtagger on ten parts of the WIC Corpus

Part	befor changes	after changes
1	80.3	80.9
2	80.7	81.7
3	82.0	82.1
4	83.6	84.2
5	84.4	85.0
6	82.8	83.5
7	85.1	85.6
8	86.5	87.0
9	84.7	85.5
10	79.1	80.0
Average	82.9	83.6

We can see that TreeTagger is significantly better than APtagger. The average accuracy of APtagger is higher 0.7 percentage points after correction of errors in PoS tags.

5 Conclusion

In this paper, we have presented the a new release of the WIC Corpus with corrections of PoS annotation of 68 words. The changes affect about 2.500 tokens but even such small number of changes has a significant positive effect on the accuracy of two different PoS taggers.

The new version of the corpus is going to be available again in the Clarin repository.

Acknowledgments. This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin infrastructure LM2015071 and by the Grant Agency of CR within the project 18-23891S.

References

1. Gambäck, B., Olsson, F., Argaw, A.A., Asker, L.: Methods for amharic part-of-speech tagging. In: Proceedings of the First Workshop on Language Technologies for African Languages. pp. 104–111. Association for Computational Linguistics (2009)
2. Gebre, B.G.: Part of speech tagging for Amharic. Ph.D. thesis, University of Wolverhampton Wolverhampton (2010)
3. Honnibal, M.: Aptagger (2013), `https://explosion.ai/blog/part-of-speech-pos-tagger-in-python`, a Good Part-of-Speech Tagger in about 200 Lines of Python
4. Rychlý, P.: Amharic WIC corpus (2016), `http://hdl.handle.net/11234/1-2593`, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University
5. Rychlý, P.: Kerneltagger–a pos tagger for very small amount of training data. RASLAN 2017 Recent Advances in Slavonic Natural Language Processing p. 107 (2017)
6. Rychlý, P., Suchomel, V.: Annotated amharic corpora. In: International Conference on Text, Speech, and Dialogue. pp. 295–302. Springer (2016)
7. Schmid, H.: Treetagger| a language independent part-of-speech tagger. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart **43**, 28 (1995)
8. Tachbelie, M.Y., Menzel, W.: Morpheme-based language modeling for inflectional language–amharic. Amsterdam and Philadelphia: John Benjamin's Publishing (2009)

Subject Index

Author Index

RASLAN 2018

Twelfth Workshop on Recent Advances in Slavonic Natural Language Processing

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

Published by Tribun EU
Cejl 32, 602 00 Brno, Czech Republic

First edition at Tribun EU
Brno 2018

ISBN 978-80-263-1517-9
ISSN 2336-4289