Proceedings of the 6th International Workshop on Computational Terminology (COMPUTERM 2020), pages 72–79
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Supporting terminology extraction with dependency parses

Małgorzata Marciniak, Piotr Rychlik, Agnieszka Mykowiecka
Institute of Computer Science, Polish Academy of Sciences
Jana Kazimierza 5, 01-248 Warsaw, Poland
{mm, rychlik, agn}@ipipan.waw.pl

Abstract

Terminology extraction procedure usually consists of selecting candidates for terms and ordering them according to their importance for the given text or set of texts. Depending on the method used, a list of candidates contains different fractions of grammatically incorrect, semantically odd and irrelevant sequences. The aim of this work was to improve term candidate selection by reducing the number of incorrect sequences using a dependency parser for Polish.

Keywords: terminology extraction, candidates filtering, dependency parsing, prepositional phrases

1. Introduction

Extracting important domain related phrases is a part of very many NLP tasks such as information extraction, indexing or text classification. Depending on a particular scenario either more precise or more robust solutions are preferable. In our terminology extraction work, the aim is to prepare preliminary lists for building terminology resources or text indexing. As manual checking of the prepared list is expensive, we are interested in a solution in which the top of the ordered candidates list is of the highest quality. One of the problems of all term extraction methods is the fact that some extracted sequences are incorrect. The sequences recognized using statistical methods or shallow grammars can sometimes be semantically odd or even incorrect at the syntactic level. We identify two types of errors. In the first, the shallow patterns cover only part of the phrase, e.g., *resolution microscopy*. In the second, parts of two independent phrases are merged into a sequence which does not form a coherent phrase, e.g., *high resolution microscopy designed*. The aim of this work was to improve term candidate selection by reducing the number of incorrect sequences using a dependency parser for Polish. The answer to the question whether using a deep parser improves term identification would have been evident if the parsing were perfect. In such a case, at least all syntactically incorrect phrases (the errors of the second type mentioned above) would have been eliminated. However, errors of the first type are rather hard to identify on syntactic grounds.

Dependency analysis classifies all modifiers as adjuncts, some of them are necessary term parts and indicate a particular subtype, e.g., *basic income*, while others are just modifications which specify frequency, intensity or quality features and do not constitute a part of a term, e.g., *bigger income*. That is why we propose a hybrid approach, not just dependency parsing.

In this paper, we will not discuss the computational aspects of dependency parsing. Although it can significantly slow down the extraction process, it might still be useful in cases where the potential user wants to improve the quality of the output. Besides, not all sentences of the processed text need to be analyzed by a dependency parser, but only those containing examined terms.

2. Related Work

Terminology extraction (sometimes under the name of keyword/keyphrase extraction) is quite a popular NLP task which is tackled by several tools available both as open access and commercial systems. An overview of biomedical terminology extraction is presented in (Lossio-Ventura et al., 2016), several keyphrase extraction systems described in the scientific literature were later presented in (Merrouni et al., 2019). The latter paper mainly describes solutions which were proposed within the area of text mining or artificial intelligence, while quite a lot of other approaches were proposed at more natural language processing and terminology extraction oriented venues, e.g., TermSuite (Cram and Daille, 2016) and Sketch Engine (Kilgarriff et al., 2014). Competitions in automatic term extractions have been also organised, e.g., at SemEval workshop (Kim et al., 2010) or (Augenstein et al., 2017).

Terminology extraction systems can be divided into two groups. In one group, term extraction is treated as any other extraction task and is usually solved as a classification task using statistical, e.g., CRF, (Zhang, 2008), (Yu et al., 2012), or deep learning methods, e.g., (Zhang et al., 2016), (Meng et al., 2017). The other approach, also accepted by the extraction tool we use (TermoPL), comes from collocation/phrase recognition work. Most of the term extraction systems which were developed along these lines follow the standard three phase procedure consisting of text preprocessing, potential term selection and term scoring. Text preprocessing depends on the source of texts and the language in which they are written and usually consists of filtering out unnecessary information, tokenization and sometimes POS tagging. As a lot of work was done for English, most approaches for candidate selections are based on selecting just word n-grams on the basis of the simple frequency based statistics, e.g., (Rose et al., 2010) or on the shallow grammars usually written as a set of regular expressions over POS tags, e.g., (Cram and Daille, 2016). Deep syntactic grammars are hardly used at all. One solution in which dependency grammar is used to extract term candidates is described in Gamallo (2017). Dependency parses were also analyzed in Liu et al. (2018). All the above approaches to candidate selection are approximate (for different reasons), i.e. some term candidates are improper while

others are omitted. In our work, we used shallow grammars with additional specification of morphological values dependencies. As Polish is an inflectional language, this approach allows a lot of grammatically incorrect phrases to be filtered out while, at the same time, it is not limited to the sequences recognized properly by a deep parser for Polish, which for a specific domain might not have enough coverage.

The second step of the process – candidate ranking – is also carried out in very different ways. The frequency of a term or frequency based coefficients play the most prominent role. The most popular is tf-idf, but the C-value (Frantzi et al., 2000), used in this paper, also became widely used. Unlike many other coefficients, the C-value takes into account not only the longest phrases or sequences of a given length, but also sequences included in other, longer, sequences.

Although in some approaches the ranking procedure may be very complex, the idea of an additional phase of filtering improperly built pre-selected phrases, as suggested in our paper, is not very popular. There are, however, some solutions with a post filtering phrase, e.g. (Liu et al., 2015), in which the candidates are compared to different external terminology resources. This approach was not adopted in our work, as it cannot be used to identify new terms and it requires resources adequate for a specific domain. Another postulated modification of the overall processing schema is the final re-ranking procedure adopted in (Gamallo, 2017).

As in many other NLP tasks, evaluation of the terminology extraction results is both crucial and hard to perform. Evaluation can either be performed manually or automatically. In the first case, apart from the cost of the evaluation, the main problem is that sometimes it is hard to judge whether a particular term is domain related or comes from general language. Automatic evaluation requires terminological resources (which, even if they exist, are usually not complete), or preparing the gold standard labelled text (which has similar problems to direct manual evaluation). In statistical methods, the automatic evaluation procedure is usually used. In (Merrouni et al., 2019), the results of several systems show the overall very poor recall (0.12-0.5) and a little higher precision (0.25-0.7) with the F1 measure usually below 0.3. Manual verification usually covers the top few hundred terms which are judged by a domain expert to be domain related terms or not. In this approach, only the precision of the results can be evaluated at reasonable cost. Gamallo (2017) reports precision of 0.93 for the first 800 terms extracted from English biomedical texts using an approach similar to that adopted by us. In (Marciniak and Mykowiecka, 2014), the then existing version of the TermoPL gave precision of 0.85 for 800 top positions of the terms list obtained from medical clinical reports. The recall counted on four reports (a very small dataset) was 0.8. The poorer results obtained for Polish data are mainly caused by the poor quality of text with many errors and missing punctuation marks (both commas and dots).

The results of the two groups of methods described above cannot be directly compared, but the good quality of the linguistically based methods is the reason why we want to develop this approach to terminology extraction.

3. Tools Description

3.1. TermoPL

As the baseline method of term selection for our experiments we chose one implemented in the publicly available tool – TermoPL (Marciniak et al., 2016). The tool operates on the text tagged with POS and morphological features values and uses shallow grammar to select the term candidates. Grammar rules operate on forms, lemmas and morphological tags of the tokens. They thus allow for imposing agreement requirements important for recognizing phrase borders in inflectional languages, such as Polish. TermoPL has a built-in grammar describing basic Polish noun phrases and also allows for defining custom grammars for other types of phrases. The program was originally developed for the Polish language so it is capable of handling the relatively complex structural tagset of Polish (Przepiórkowski et al., 2012). It is also possible to redefine this tagset and process texts in other languages. To eliminate sub-sequences with borders crossing strong collocations, the NPMI (Bouma, 2009) based method of identifying the proper sub-sequences was proposed (Marciniak and Mykowiecka, 2015). According to this method, subphrase borders are subsequently identified between the tokens with the smallest NPMI coefficient (counted for bigrams on the basis of the whole corpus). So, if a bigram constitutes a strong collocation, the phrase is not being divided in this place, and this usually blocks creation of semantically odd nested phrases.
The final list of terms is ordered according to the C-value adapted for taking one word terms into account. The C-value is a frequency dependent coefficient but takes into account not only the occurrences of the longest phrase, but also counts occurrences of its sub-sequences.

3.2. COMBO

In our experiments we use a publicly available Polish dependency parser – COMBO (Rybak and Wróblewska, 2018). COMBO is a neural net based jointly trained tagger, lemmatizer and dependency parser. It assigns labels based on features extracted by a biLSTM network. The system uses both fully connected and dilated convolutional neural architectures. The parser is trained on the Polish Dependency Bank (http://zil.ipipan.waw.pl/PDB). In our work we used the version trained on PDB annotated with a set of relations extended specifically for Polish (http://zil.ipipan.waw.pl/PDB/DepRelTypes).

4. Data Description

The experiment was conducted on the textual part of an economics articles taken from Polish Wikipedia. It was collected in 2011 as part of the Polish Nekst project (POIG.01.01.02-14-013/10). The data contains 1219 articles that have economics related headings and those linked to them.
The data was processed by the Concraft tagger (Waszczuk, 2012) which uses Morfeusz morphological dictionary and a guesser module for unknown words. The processed text has about 460K tokens in around 20,000 sentences. There are about 46,600 different token types of 17,900 different lemmas or known word forms within the text.

$$
\begin{aligned}
NPP &: \$NAP\ NGEN^*;\\
NAP[agreement] &: AP^*\ N\ AP^*;\\
NGEN[case = gen] &: NAP;\\
AP &: ADJ \mid PPAS \mid\\
&\quad\ ADJA\ DASH\ ADJ;\\
N[pos = subst, ger];&\\
ADJ[pos = adj];&\\
ADJA[pos = adja];&\\
PPAS[pos = ppas];&\\
DASH[form = \text{"-"}];&
\end{aligned}
$$

Figure 1: The built-in grammar represents noun phrases comprised of nominal phrases built from nouns or gerunds optionally modified by adjectival phrases located either before or after them. Nominal phrases can be modified by any number of nominal phrases in the genitive.

5. Phrase identification

A selection of candidate phrases is performed by a shallow grammar defined over lemmatized and morphologically annotated text. TermoPL recognizes the maximal sequences of tokens which meet the conditions set out in a grammar.

The built-in grammar, see Fig. 1, recognizes noun phrases where the head element can by modified by adjectives appearing before or after it, such as *międzynarodowe stosunki gospodarcze* 'international economic relations'. All these elements must agree in number, case and gender, which is marked in the rules by the *agreement* parameter. The noun phrase can be modified by another noun phrase in the genitive, e.g., *ubezpieczenie [odpowiedzialności cywilnej]*$_{gen}$ 'insurance of civil responsibility'. All these elements can be combined, e.g., *samodzielny publiczny zakład [opieki zdrowotnej]*$_{gen}$ 'independent public health care'. The \$ character marks a token or a group of tokens which should be replaced by their nominal forms when base forms are generated. It does not affect the type of phrase being recognized. In the economics texts, the built-in grammar collects 61,966 phrases when the NPMI driven selection method is used (without the NPMI it collects 82,930 phrases).

The built-in grammar does not cover noun phrases modified by prepositional phrases which quite often create important terms, e.g., *spółka z ograniczoną odpowiedzialnością* 'limited liability company'. This decision was made because it was difficult to recognize the role of a prepositional phrase in a sentence. A phrase very similar to the one above, e.g., *umowa spółki z uniwersytetem* 'a company agreement with the university' (word for word translation: ('agreement' 'company' 'with' 'university') should not lead to a conclusion that *spółka z uniwersytetem* creates a term – both nouns *firma* 'company' and *uniwersytet* 'university' are complements of the noun *umowa* 'agreement'. If the only criterion is a shallow grammar, we are unable to distinguish between such uses.

When analyzing the results obtained by the grammar defined in Fig. 1, we realised that some nominal phrases can have a noun phrase complement in the instrumental case. It applies to phrases such as, e.g., *handel [ropą naftową]*$_{instr}$ 'trading in petroleum', *gospodarka nieruchomościami*$_{instr}$ 'management of real estate' *opodatkowanie [podatkiem dochodowym]*$_{instr}$ 'taxation of income'. But a similar problem, as for prepositional phrases, occurs for noun complements in the instrumental case, as we don't know if they are complements of a preceding nominal phrase or if they refer to another element in the sentence. For example: *rząd obłożył [papierosy] [akcyzą]*$_{instr}$ (word for word translation: 'government' 'charged' 'cigarettes' 'excise duty') 'the government charged cigarettes with excise duty', where *akcyzą* 'excise duty' is the complement of *obłożył* 'charged' and not *papierosy* 'cigarettes'.

$$
\begin{aligned}
NP &: NPPINST \mid PPP \mid NPAPGEN;\\
PPP &: NPAPGEN\ PREP\ NAP^+;\\
NPPINST &: NPAPGEN\ NINST\ NGEN^*;\\
NPAPGEN &: \$NAP\ NGEN^*;\\
NAP[agreement] &: AP^*\ N\ AP^*;\\
NGEN[case = gen] &: NAP;\\
NINST[case = inst] &: NAP;\\
AP &: ADJ \mid PPAS \mid\\
&\quad\ ADJA\ DASH\ ADJ;\\
N[pos = subst, ger];&\\
ADJ[pos = adj];&\\
ADJA[pos = adja];&\\
PPAS[pos = ppas];&\\
DASH[form = \text{"-"}];&\\
PREP[pos = prep];&
\end{aligned}
$$

Figure 2: The final grammar with added modifiers being noun phrases in the instrumental case and prepositional phrases.

Both constructions described above, i.e. prepositional modifiers and noun complements in the instrumental case, are taken into account in the grammar given in Fig. 2. It collects 72,758 phrases when the NPMI driven selection method is used, which is over 10,000 more than for the built-in grammar (without the NPMI the grammar collects 113,687 phrases). Although the number of new terms is high, there are a couple of new top candidates on our list. The top 100 terms contains three correct phrases with prepositional modifiers *spółka z ograniczoną odpowiedzialnością* 'limited liability company', *ustawa o rachunkowości* 'accounting act' and *podatek od towarów* 'tax on goods', and no term with a noun complement in the instrumental case. The first such phrase *prawo o publicznym obrocie papierami wartościowymi* 'law on public trading of securities' is in position 637.

As we wanted to know how productive the above grammatical constructions are, we have defined two grammars describing them alone. This allows us to check how many phrases might be introduced to the term candidate list by these constructions.

type	number of phrases absolute	relative
correct term	452	0.66
incorrect modification	114	0.17
incorrect – other reason	120	0.17

Table 1: Manual evaluation of the top phrases with a preposition modifier

frequency	number of phrases absolute	relative
30-38	5	0.01
20-29	5	0.01
10-19	29	0.04
5-9	101	0.15
3-4	244	0.35
2	302	0.44

Table 2: Number of top phrases with a preposition modifier in different frequency groups.

The first dedicated grammar (NPPP) defines nominal phrases with a prepositional modifier. It consists of all rules given in Fig. 2 except the first, third and the seventh one. When the NPMI method is used, the grammar selects 22,150 terms. We evaluate all phrases which occurred at least 2 times and have a C-value of at least 3.0, i.e. 686 phrases. The results are given in Tab 1 – 66.6% of them are correct phrases (i.e. for these phrases precision is 0.666), 16.5% are phrases where a preposition phrase does not modify the preceding noun phrase, and for 16.9% a reason for not accepting the phrase is different. Many incorrect phrases are incomplete, such as *różnica między sumą przychodów uzyskanych* 'difference between the sum of revenues obtained' which is a part of *różnica między sumą przychodów uzyskanych z tytułu . . . a kosztami uzyskania przychodów* 'difference between the sum of revenues from . . . and tax deductible costs'.

The second grammar (NPInst) defines nominal phrases modified by noun phrases in the instrumental case. It consists of all rules given in Fig. 2 except the first and the second one. It selects fewer phrases, namely 1390. As there was only 44 phrases with a C-value of at least 3.0, we evaluated all 110 phrases which occurred at least twice. The results are given in Tab. 3. The example of an incorrect phrase recognised by the grammar is *budownictwo kosztorysantem* ('architecture' 'estimator') which actually is built up from two phrases and occurred three times in sentences similar to the following: *w [budownictwie kosztorysantem] jest rzeczoznawca* 'in architecture, an appraiser is an estimator'. Tab. 4 gives the frequency of the evaluated phrases. The statistics show that such constructions are not common. Moreover, we observe that only a small number of nouns and gerunds (acting as nouns) were used to create valid phrases in our data. These are: *obrót* 'trading' (21), *zarządzanie* 'management' (21), *handel* 'trade' (9), *opodatkowanie* 'taxation' (5), *gospodarka/gospodarowanie*'management' (4).

type	number of phrases absolute	relative
correct term	84	0.76
incorrect modification	10	0.09
incorrect – other reason	16	0.14

Table 3: Manual evaluation of selected phrases with an instrumental modifier.

frequency	number of phrases absolute	relative
10-16	4	0.04
5-9	11	0.10
3-4	20	0.18
2	75	0.68

Table 4: Number of top phrases with an instrumental modifier in different frequency groups.

6. Filtering phrases with COMBO

In the postprocessing phase, we match the phrases found by NPPP and NPInst TermoPL grammars to some fragments of the dependency trees generated by COMBO for sentences containing these phrases. We imposed a few simple constraints that must be satisfied by matched tree fragments. The first one concerns prepositional phrases. If the preposition in the phrase being examined is associated with a part of the sentence that is not included in the phrase, then this phrase is certainly not a valid term. In other words, it means that no link in the dependency tree is allowed to connect something from outside the matched fragment with the preposition that lies inside this fragment. Examples of a good and a bad prepositional phrase are shown in Fig. 3 and 4, respectively. The first of these phrases *spółka z ograniczoną odpowiedzialnością* 'limited liability company' is a good example of an economics term. The second one *przedsiębiorstwo pod własną firmą* is a nonsense phrase that has a word for word translation 'enterprise under own company', which is equally nonsensical.

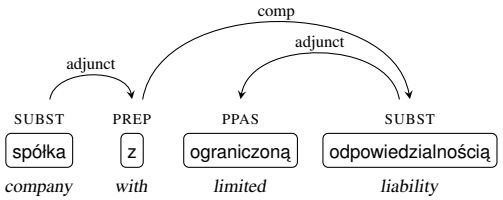

Figure 3: Dependency graph corresponding to correct prepositional phrase *spółka z ograniczoną odpowiedzialnością*.

It turns out that there are phrases that in some sentences are good candidates for terms, and in others not. A string *podatek od dochodu* which has the word for word translation 'tax from income' can be a noun modifier, e.g., *[podatek od dochodu] należy zapłacić w terminie do . . .* 'income tax must be paid by . . . ', or it can be a valency constraint in the following sentence *Wyliczając kwotę do za-*

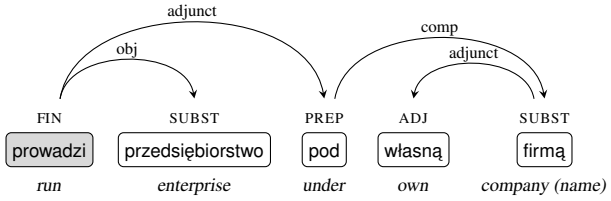

Figure 4: Dependency graph corresponding to the incorrect prepositional phrase *przedsiębiorstwo pod własną firmą*.

płaty należy odjąć [podatek] [od dochodu]. 'When calculating the amount to be paid, tax must be deducted from the income.' In the first example (see Fig. 5), the term *podatek od dochodu* is accepted by the constraint we mentioned above. In the second example, the same constraint rejects this phrase as a term (see Fig. 6).

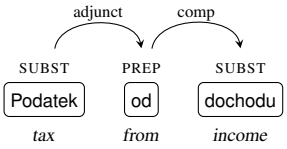

Figure 5: Accepted term *podatek od dochodu*.

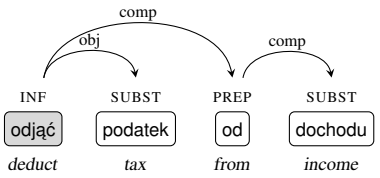

Figure 6: Rejected term *podatek od dochodu*.

The second constraint we impose on dependency graphs concerns the consistency of its matched fragment. A fragment of the graph corresponding to the examined phrase is consistent if, passing from the node considered as the head of the phrase, we pass through all its nodes. Fig. 7 presents an inconsistent graph for the phrase *koszty dojazdów środkami* (with word for word transtation 'travel costs by means'), which is syntactically correct, but without sense. However, when we consider the broader context, the phrase *pokrycie kosztów dojazdów środkami komunikacji miejscowej* 'coverage of travel costs by local transport', we obtain a phrase that makes sense and has a consistent graph. The graph for the phrase *podatek od dochodu* depicted in Fig. 6 is also inconsistent with this constraint (although it would anyway be rejected by the first rule described above). Finally, we eliminate graphs that correspond to some types of truncated phrases. They are depicted in Fig. 8-10. Fig. 8 shows an example in which a named entity phrase should not be divided. The phrase *Ustawa o Funduszu Kolejowym* 'Act on the Railway Fund' may not be shortened to the phrase *Ustawa o Funduszu* 'Act on the Fund', although it is still acceptable at the syntactic level. The other two examples show situations in which an adjective or participle, modifying an object or a complement, should not be cut

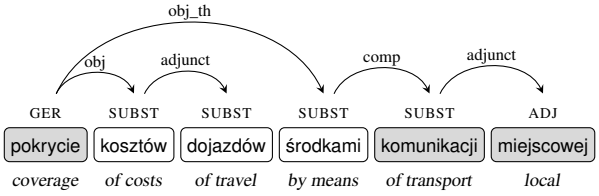

Figure 7: Graph inconsistency for the phrase *koszty dojazdów środkami*.

from the phrase on its right side, as they are usually necessary components of terms. The phrase *podatek dochodowy od osób fizycznych* 'personal income tax' (see Fig. 9) cannot be shortened to *podatek dochodowy od osób*. Similarly, the phrase *opodatkowanie podatkiem dochodowym* 'taxation on income' (see Figure 10) cannot be shortened to *opodatkowanie podatkiem*.
Sometimes, truncated phrases can be identified by their inconsistent graphs as shown in Fig. 7.

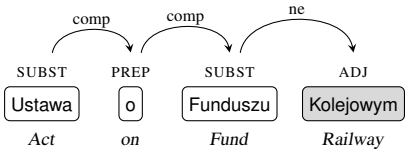

Figure 8: Truncated named entity (ne) phrase.

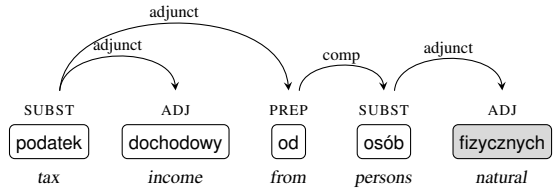

Figure 9: Truncated phrase *podatek dochodowy od osób*.

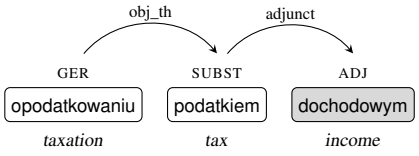

Figure 10: Rejected term *opodatkowanie podatkiem*.

We can now use all the above constraints to filter phases. If a phrase is supported by more than 50% of its dependency trees (which means that these trees satisfy all constraints), it is considered as a good term candidate. Otherwise, it is rejected.

7. Evaluation of the method

We compare the manual evaluation of all phrases obtained by two separate grammars with the results of filtering described in Sec. 6. The filtering gives the binary information: correct/incorrect phrase so we assume that the result

is proper if an incorrect phrase is manually classified as incorrect modification or incorrect 'other reason'. Tables 5-6 give the evaluation of phrases with prepositional modifiers and phrases with instrumental modifiers respectively, classified by the dependency parser. The results depicted there show that the proposed approach is not precise enough. For phrases with prepositional phrases, 74% of correct phrases are correctly classified as valid terms, but there are about twenty percent of the proper terms which are discarded. There are even more incorrect sequences which are classified as correct (about quarter). For instrumental modifications, there are far fewer incorrect sequences accepted as good, while the percentage of the correct terms which are classified as bad is even higher than for prepositional modifiers. The answer to the question whether these results are due to our classification strategy not being good enough or to the insufficient quality of the parser needs further research.

Manual eval. \ COMBO	correct	incorrect
correct	365	87
incorrect	131	103

Table 5: Comparison of the manual evaluation of the phrases with a preposition modifier with the dependency parser filtering. The results achieved for classification a phrase as correct by the parser: precision=0.74, recall=0.81.

Manual eval. \ COMBO	correct	incorrect
correct	50	34
incorrect	4	22

Table 6: Evaluation of the phrases with an instrument modifier filtered by the dependency parser. The results achieved for classification a phrase as a correct one by the parser: precision=0.93, recall=0.60.

type	in top3.0	out top3.0	out
correct term	391	27	34
incorr modif.	67	15	32
incorr. – other	59	33	38
total	487	75	104

Table 7: Phrases with a preposition modifier – with NPMI.

8. Results

In this section, we analyze results of TermoPL using the extended grammar given in Fig. 2. A set of phrases for which the C-value is at least 3.0 are called hereinafter top3.0. For the plain method of term selection (without NPMI), the top3.0 consists of 5,935 terms. Phrases with prepositional modifiers are 11.9% of the top3.0 set. 7.6% of them are correct phrases and 4.3% are incorrect ones.

Then, we test if the NPMI method can prevent us from introducing incorrect phrases with prepositional modifiers into the top3.0 set. For some phrases, the NPMI method reduces their C-value which means they are pushed to the end of the list. Moreover, some phrases may not even appear on the term list. The top3.0 set for TermoPL with NPMI consists of 5,078 phrases. The statistics are given in Tab. 7, where the 'out top 3.0' column indicates the number of phrases whose C-value fell below the 3.0 level, and the 'out' column indicates the number of phrases which disappeared from the list. This method introduced 487 prepositional phrases into top3.0, which is 9.5%. 7.7% of them are correct phrases and 1.8% are incorrect ones. Tab. 8 gives the location of the correct 391 phrases on the top3.0 list ordered by C-value.

		number of phrases	
C-value	position	absolute	relative
<50-278)	1-78	1	0.2%
<20-50)	79-343	9	2.3%
<10-20)	344-899	31	7.9%
<5-10)	900-2129	113	28.9%
<3-5)	2130-5078	237	60.6%

Table 8: Distribution of the correct phrases with prepositional modifiers in top3.0 of TermoPL with NPMI.

type	in top3.0
correct-accepted	365
incorrect-accepted	131
correct-deleted	86
incorrect-deleted	101

Table 9: Phrases with a preposition modifier filtered by the dependency parser from the plain TermoPL results.

As we expected, application of the NPMI method in candidate phrase recognition reduces the number of incorrect phrases in the top3.0. In our experiment, they drop from 4.3% to 1.8% of all the top3.0. It slightly declines the share of phrases with prepositional modifiers on the top3.0 list from 11.9% to 9.5%. Moreover, it seems that this method works better for phrases which are incorrect because of 'other reasons' (e.g. truncated ones), as from the top3.0, it eliminates 71 of 130 such phrases (i.e. 54.6%) while for incorrect modifiers it eliminates 47 of 114 phrases which is 41.2%.

For nominal phrases with instrumental modifiers only 37 correct and 2 incorrect (because of 'other reasons') were on the top3.0 list generated without NPMI. Statistics are given in Tab. 3. It gives 0.66% of all top3.0 phrases, where 0.62% are correct new phrases. When the NPMI method is used, the top3.0 list contains 30 correct and 2 incorrect phrases with instrumental modifiers, which gives 0.63% of all top3.0 phrases including 0.59% correct new phrases.

To assess the usefulness of the dependency parsing we checked how many phrases with prepositional modifiers were accepted or deleted from the top3.0 of the TermoPL

results generated without NPMI. The results for preposition modifiers are given in Tab. 9. So, filtering prepositional phrases by dependency grammar results in removing 187 phrases, where 101 of them were incorrect (i.e., their removal was justified).

9. Conclusion

The purpose of this work was to test whether dependency parsing can be useful in filtering out incorrectly constructed phrases in automatic terminology extraction. We tested this approach on phrases containing prepositional modifiers and nominal modifiers in the instrumental case.

We realised that noun phrases with prepositional modifiers are important in the terminology extraction task, as they constitute about 10% of the top term phrases. The phrases with instrumental case modifiers are much less important as they create only 0.65% of the top phrases. However, it is worth noting that there are only two incorrect such phrases among the top3.0. These constructions are much rarer and the most frequent phrases usually form correct terms.

There are about 6% of correct and 2% of incorrect preposition phrases on the top3.0 list generated without applying NPMI and filtered with the dependency parser. These results seem slightly worse than the results obtained by the NPMI method alone. It occurs that dependency parsing filters out an additional 43 incorrect phrases from the top3.0 list when the NPMI method is applied. Unfortunately, it also filters out 85 correct phrases. This observation requires further investigation.

As the quality and efficiency of the dependency parsing is constantly improving, we hope that these methods will better support the selection of term candidates. We also plan to check how the proposed filtering methods will work on terms with other syntactic structures.

Acknowledgements

Work financed as part of the investment in the CLARIN-PL research infrastructure funded by the Polish Ministry of Science and Higher Education.

Bibliographical References

Augenstein, I., Das, M., Riedel, S., Vikraman, L., and McCallum, A. (2017). SemEval 2017 task10: ScienceIE-extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation*, pages 546–555. Association for Computational Linguistics.

Bouma, G. (2009). Normalized (pointwise) mutual information in collocation. In Christian Chiarcos, et al., editors, *From Form to Meaning: Processing Texts Automatically, Proceedings of the Biennial GSCL Conference*, pages 31–40. Tubingen: Gunter Narr Verlag.

Cram, D. and Daille, B. (2016). TermSuite: Terminology extraction with term variant detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics—System Demonstrations*, pages 13–18. Association for Computational Linguistics.

Frantzi, K., Ananiadou, S., and Mima, H. (2000). Automatic recognition of multi-word terms: the C-value/NC-value method. *Int. Journal on Digital Libraries*, 3:115–130.

Gamallo, P. (2017). Citius at SemEval-2017 task 2: Cross-lingual similarity from comparable corpora and dependency-based contexts. In *Proceedings of the 11th International Workshop on Semantic Evaluation*, pages 226–229. Association for Computational Linguistics.

Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., and Pavel Rychlý, V. S. (2014). The Sketch Engine: ten years on. *Lexicography*, 1:7–36.

Kim, S. N., Medelyan, O., Kan, M.-Y., and Baldwin, T. (2010). Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26. Association for Computational Linguistics.

Liu, W., Chung, B. C., Wang, R., Ng, J., and Morlet, N. (2015). A genetic algorithm enabled ensemble for unsupervised medical term extraction from clinical letters. *Health Information Science and Systems*.

Liu, Y., Zhang, T., Quan, P., Wen, Y., Wu, K., and He, H. (2018). A novel parsing-based automatic domain terminology extraction method. In Shi Y. et al., editor, *Computational Science – ICCS 2018. Lecture Notes in Computer Science, vol 10862. Springer, Cham*, pages 796–802.

Lossio-Ventura, J. A., Jonquet, C., Roche, M., and Teisseire, M. (2016). Biomedical term extraction: overview and a new methodology. *Information Retrieval Journal*, pages 1573–7659.

Marciniak, M. and Mykowiecka, A. (2014). Terminology extraction from medical texts in Polish. *Journal of biomedical semantics*, 24.

Marciniak, M. and Mykowiecka, A. (2015). Nested term recognition driven by word connection strength. *Terminology*, 2:180–204.

Marciniak, M., Mykowiecka, A., and Rychlik, P. (2016). TermoPL — a flexible tool for terminology extraction. In Nicoletta Calzolari, et al., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation, LREC 2016*, pages 2278–2284, Portorož, Slovenia. ELRA, European Language Resources Association (ELRA).

Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., and Chi, Y. (2017). Deep keyphrase generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 582–592, Vancouver, Canada, July. Association for Computational Linguistics.

Merrouni, Z. A., Frikh, B., and Ouhbi, B. (2019). Automatic keyphrase extraction: a survey and trends. *Journal of Intelligent Information Systems*.

Adam Przepiórkowski, et al., editors. (2012). *Narodowy Korpus Języka Polskiego*. Wydawnictwo Naukowe PWN.

Rose, S., Engel, D., Cramer, N., and Cowley, W. (2010). Automatic keyword extraction from individual documents. *Text Mining: Applications and Theory*, pages 1 – 20, 03.

Rybak, P. and Wróblewska, A. (2018). Semi-supervised

neural system for tagging, parsing and lematization. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, page 45–54. Association for Computational Linguistics.

Waszczuk, J. (2012). Harnessing the CRF complexity with domain-specific constraints. The case of morphosyntactic tagging of a highly inflected language. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 2789–2804.

Yu, F., Xuan, H., and Zheng, D. (2012). Key-phrase extraction based on a combination of CRF model with document structure. In *Eighth International Conference on Computational Intelligence and Security*, pages 406–410.

Zhang, Q., Wang, Y., Gong, Y., and Huang, X. (2016). Keyphrase extraction using deep recurrent neural networks on twitter. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 836–845, Austin, Texas, November. Association for Computational Linguistics.

Zhang, C. (2008). Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems*, 4(3):1169–1180.