TermoUD — a language-independent terminology extraction tool

Małgorzata Marciniak and **Piotr Rychlik** and **Agnieszka Mykowiecka**
Institute of Computer Science, Polish Academy of Sciences
Jana Kazimierza 5, 01-248 Warsaw, Poland

Abstract

The paper addresses TermoUD — a language-independent terminology extraction tool. Its previous version, i.e. TermoPL (Marciniak et al., 2016; Rychlik et al., 2022), uses language dependent shallow grammar which selects candidate terms. The goal behind the development of TermoUD is to make the procedure as universal as possible, while taking care of the linguistic correctness of selected phrases. The tool is suitable for languages for which the Universal Dependencies (UD) parser exists. We describe a method of candidate term extraction based on UD POS tags and UD relations. The candidate ranking is performed by the C-value metric (contexts counting is adapted to the UD formalism), which doesn't need any additional language resources. The performance of the tool has been tested on texts in English, French, Dutch, and Slovenian. The results are evaluated on the manually annotated datasets: ACTER, RD-TEC 2.0, GENIA and RSDO5, and compared to those obtained by other tools.

1 Introduction

The purpose of automatic term extraction (ATE) is to identify recurring phrases that are relevant to the domain of a given text. Such phrases can then be interpreted as candidates for key phrases, index terms or potential domain lexicon entries.

The first among many approaches to this problem is selecting term candidates based on one of the following methods: n-grams (Rose et al., 2010); a set of patterns defining sequences of part-of-speech (POS) tags allowed within phrases (Hulth, 2003); phrases identified by a syntactical parser or an NP-chunker (Cram and Daille, 2016). All the generated candidate terms are then ranked with an ordering procedure based, among other things, on tf/idf (Salton, 1988), the C-value (Frantzi et al., 2000), or the mutual information value. The top elements of the obtained list are treated as domain terminology. Methods based solely on n-grams are language independent but achieve worse results (especially for inflectional languages) than those based on shallow-parsers which are language dependent. One of the important objectives of developers of ATE tools is to make them language independent. The JATE system (Zhang et al., 2016) selects candidate terms on the basis of syntactic analysis. However, it is designed to make it easy to adapt to different domains and/or languages — a flexible mechanism to determine how candidate phrases are constructed has been defined.

The second approach to ATE involves combining solely statistical features extracted from the processed text used in heuristics selecting terminology. An example of this approach is YAKE! (Campos et al., 2020), which supports 9 languages. The score assignment in YAKE! combines features such as letter case, a position within the text, word frequency and the number of different sentences in which a given term appears and, finally, the number of different contexts in which a term appears. Scores for 1-grams are combined to give the ranking of n-grams. The method gives good results for keyphrase extraction from short texts. The method doesn't work so well for inflection-rich languages, as the statistics are counted on forms, moreover; the processing of texts longer than a few pages is time-consuming.

The newest approach solves the problem of terminology extraction as a sequential tagging task and applies machine learning methods similar to the Named Entities Recognition. This approach was first used in the Termeval 2020 shared task (Rigouts Terryn et al., 2020), whose organisers published English, French and Dutch data collected in the Annotated Corpora for Term Extraction Research (ACTER) (Terryn et al., 2020).

All of the above approaches have various limitations. Although the latter approach has proven quite effective on the ACTER corpus, its use is limited to cases where we have annotated data, which

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics
System Demonstrations, pages 178–186

are quite rare. Using only statistical heuristics to identify phrases has proven ineffective, especially for highly inflected languages, and defining POS patterns or grammatical rules requires knowledge of the language in question. Here we propose a method that, while not completely universal, can be used without additional modifications or resources for a great many languages.

Our tool for terminology extraction performs selection of candidate terms by using dependency parsing. The presented method is language independent and time-efficient. Nowadays, dependency parsers are very popular and are available for many languages (e.g. SpaCy works for 20, and Stanza for 70 languages) and are robust enough to be used in NLP applications. They are quite naturally used a lot in relation extraction, e.g. (Fundel et al., 2006) or (Geng et al., 2020), but there is still little interest in using dependency parsing in terminology extraction. The only two known approaches are (Gamallo, 2017) and (Liu et al., 2018). Gamallo used dependency parsers for bilingual term alignment. In the second paper, the authors used dependency parses for candidate selection for Chinese and achieved better results (both in terms of recall and precision) than using only POS based rules. In (Marciniak et al., 2020), the authors proposed the post-processing of selected phrase-candidates by checking the consistency of dependency parses of already selected phrases.

2 Extraction Process

To make our program as universal as possible, we had to define a set of rules to identify, without any changes, noun phrases in dependency trees constructed by parsers processing sentences in different languages.

2.1 Identification of Candidate Phrases

The UD project assumes a consistent structure of annotation schemes for many languages. In the terminology candidate identification algorithm described below, we use this consistency to define rules for selecting nominal phrases that are based on four sets of information. Two sets consist of UD POSs. The first – `head-pos` – contains UD POSs of nodes that can be heads of the term phrases, i.e. `NOUN`, `PROPN` and `VERB` (if the considered node is classified as a gerund). The second – `non-head-pos` – contains UD POSs of nodes that can be part of the term phrases but not their

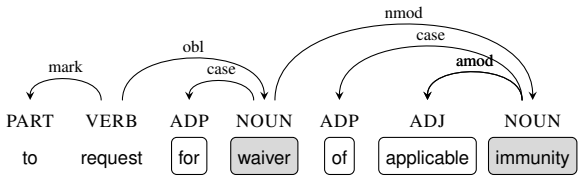

Figure 1: Dependency graph corresponding to phrase: *to request for waiver of applicable immunity*. All framed words are terminology nodes, heads are gray; obligatory relations: `case`, facultative relation: `amod`, `nmod`, `obl`.

heads, i.e. `ADJ`, `ADP`, `ADV`, `DET` and `NUM`. The next two sets consist of relations: `obligatory-rel` and `facultative-rel`. The first set groups relations between words that should appear together in terminology phrases, while the second set contains relations between words that may or may not appear in a sentence. The appropriate relations are listed below:

`obligatory-rel`: `amod:flat`, `case`, `case:poss`, `ccomp`, `compound`, `compound:prt`, `expl:pv`, `fixed`, `flat`, `iobj`, `nmod:arg`, `nmod:flat`, `nsubj:ger`,`obj`, `obl:agent`,`obl:arg`, `xcomp`.

`facultative-rel`: `acl`, `advmod`, `advmod:emph`,`amod`, `appos`, `nmod`, `nmod:poss`, `nummod`, `nummod:gov`, `obl`

Note that even if we add the relations that are only typical for a certain language to the lists, it will not destroy the generality of the solution. These relations will not affect the results obtained for other languages.

To create UD structures from plain text, we use the Stanza (Qi et al., 2020) dependency parser. The structure obtained in this way consists of sentence nodes with relations pointing to their dependent nodes, see Fig. 1. We select all nodes which may be included in terminology phrases creating the list of potential terminology nodes. This list includes all nodes whose UD POS belong to one of the above-mentioned sets: `head-pos` or `non-head-pos`. For hyphenated compound words, which are allowed in many languages, all nodes of the UD structures representing them are placed in the list of potential terminology nodes. All nodes from structures representing hyphenated compound words, except those that are heads of these structures, are also placed in the list of hyphenated nodes. Each of the nodes in the list of hyphenated nodes will be selected for creating phrases if and only if its head is also selected.

In the structure, we leave only relations between the terminology nodes and check if, for each node,

all obligatory-relations are in the current structure. Doing so, we avoid some truncated phrases, as we do not want to create phrases with nodes that have unrealized requirements.

We repeat the process of making phrases in the loop for all nodes from the list of potential terminology nodes. For each node we take into account all combinations of dependent nodes, where nodes connected by obligatory-relations and those from the list of hyphenated nodes must be included in the phrase, while nodes connected by facultative-relations may be omitted. As the candidate term phrase, we accept only those for which the head element is included in the head-pos set. The list of established phrases for the considered node is passed to the upper node (if relevant), and the considered node is removed from the list of potential terminology nodes. The whole procedure is repeated until the list of potential terminology nodes is empty. Pseudocode for the algorithm is given in the Appendix A.

Many forms of a given term can occur in the processed texts, especially in the case of inflectional languages. Therefore, the program identifies terms by their lemmatized forms. To present terms in a more readable way, we choose their most frequently occurring forms, preferably from those in nominative case (if applicable) and/or in the singular.

The method described above allows the extraction of discontinuous phrases. Figure 1 gives a phrase from the ACTER part of the corpus on corruption. Our method extracts 5 term candidates for this structure, i.e., *waiver*, *applicable immunity*, *immunity*, *waiver of applicable immunity* and the phrase with a gap: *waiver of immunity*. All phrases, except the second one, are terms according to the manual annotation.

While our goal is to build a tool that can process texts in many languages, we are aware that omitting all language-dependent features may degrade the results. One such feature is the use of determiners, which for some languages are obligatory while in others they are used sporadically. Some pronouns, (indicative, possesive), which are usually excluded from terminological phrases, can also play role of determiners, so we focused not on syntactic classes but on the det relation. In Table 1, we have included the ratio (multiplied by 100) of the number of det relations to the number of nouns (both common and proper) for 20 languages from the PUD set used for CoNLL 2017 shared task (Zeman et al., 2017). Its value varies from above 60 for French to 2.4 for Japanese. We have chosen an arbitrary threshold equal to 20 below which we assume that terminology phrases do not include determiners. For languages with this coefficient larger than 20 we allow for determiners within terminological phases. In this case we include determiners to the set non-head-pos and make det relation obligatory.

name	tokens	N	PN	det-rel	%
French	24,131	4,672	1,272	3,857	64.9
Portuguese	21,917	4,600	1,393	3,726	62.2
Italian	22,182	4,392	1,756	3,751	61.0
Spanish	22,822	4,818	1,250	3,514	57.9
German	21,000	4,249	1,219	2,771	50.7
English	21,176	4,036	1,741	2,047	35.4
Swedish	19,076	4,035	1,216	1,017	19.4
Hindi	23,829	5,597	1,358	791	11.4
Indonesian	19,034	4,687	2,113	718	10.6
Turkish	16,536	5,829	1,525	686	9.3
Russian	19,355	4,897	1,209	476	7.8
Czech	18,565	4,482	1,091	423	7.6
Icelandic	18,831	4,101	1,464	318	5.7
Thai	22,322	6,052	1,491	413	5.5
Chinese	21,415	5,410	1,361	338	5.0
Korean	16,584	8,052	1,677	457	4.7
Finnish	15,807	4,223	1,504	245	4.3
Arabic	20,747	5,578	1,728	285	3.9
Polish	18,338	4,504	1,326	196	3.4
Japanese	28,788	7,424	1,363	210	2.4

Table 1: Frequency of using det relations in the corpora used for training dependency parsers. The columns include number of nouns, proper nouns, det dependency relations and the ratio (multiplied by 100) by the latter to the sum of all nouns.

2.2 Ranking of Phrases

The method of identifying term candidates described above leads a large number of phrases including their subphrases. For ranking candidates we use C-value coefficient which depends on the frequency of an evaluated phrase (the higher the frequency of the phrase in the text under study, the higher the C-value), its length (longer terms are preferred) and the number of different contexts in which it occurs (the C-value increases with the number of different contexts). We adapted this method to rank term candidates extracted using dependency relations. In particular, since the obtained phrases can be discontinuous, the definition of phrase contexts had to be reformulated.

When determining the context of a given phrase, we take into account its UD structure and the maximum structure that contains it. For example, for the

phrase *waiver of immunity* from Figure 1, the maximum structure will be the structure corresponding to the maximum phrase *waiver of applicable immunity*. From the maximum structure, we select those nodes that do not belong to the structure of the examined phrase and are directly adjacent to some of its nodes. We then concatenate the lemmas of the tokens corresponding to the nodes found in the order in which these tokens appear in the sentence. We treat the resulting string of characters as the context of the examined phrase. For the phrase *waiver of immunity*, its context is *applicable*.

3 Evaluation

To compare the results with other approaches, we evaluate our tool on the following corpora annotated with terminology: ACTER, GENIA, ACL RD-TEC, and RDSO5. So, we tested the method on four languages: English, French, Dutch and Slovenian. For comparison, the D-Terminer (Rigouts Terryn, 2021; Rigouts Terryn et al., 2022a) and Sketch Engine (Jakubíček et al., 2014) were also used to process the same datasets. In the case of TermoUD, we tested the plain tool without additional existing filters developed for TermoPL, e.g., removing stopwords from candidate terms.

The result of TermoUD is a sorted list of all detected phrases, with no indication of where a split between terms and non-terms is suggested. Since the ranking method used in the tool assigns the same values to many terms, the evaluation cannot be carried out at any point in the ranking list, but only in those places where the value changes. Therefore, it is not possible to compare our method with others for the lists of terms of the same length.

3.1 ACL RD-TEC

The ACL Reference Dataset for Terminology Extraction and Classification, version 2.0 (ACL RD-TEC 2.0) (QasemiZadeh and Schumann, 2016) has been developed with the aim of providing a benchmark for the evaluation of term and entity recognition tasks based on specialised text from the computational linguistics domain. It consists of 300 abstracts from articles published between 1978 and 2006 in which both single and multi-word lexical units with a specialised meaning are manually annotated. 6,818 occurrences of terms are identified in total and 1918 of them are different strings.

To compare the results of TermoUD with the best tool for English, i.e. D-terminer, we use both to extract terms and compare the results. To make the comparison more reliable, we unify upper and lower case letters, so *natural language processing* and *Natural Language Processing* are treated as the same phrase. If phrases have different character sets, we consider that they are different, e.g., *word-sense disambiguation algorithms* and *word sense disambiguation algorithms*. The results of the comparison are given in Table 3. For the comparison, we select the number of elements returned by TermoUD, which is similar to the length of the manually annotated list of terms. A specific problem with the manually annotated ACL RD-TEC data is the high number of phrases containing acronyms surrounded by parentheses (128 cases), e.g., *Question Answering (QA) systems*. Neither of the two tools recognised these phrases, in effect lowering results equally. The D-terminer doesn't indicate any phrase with an acronym inside, while TermoUD indicates them without parentheses.

	selected terms	prec.	recall	F1
D-terminer	613	0.51	0.16	0.25
TermoUD(1)	171	0.60	0.05	0.10
TermoUD(2)	1276	0.26	0.17	0.21
TermoUD(3)	2610	0.28	0.38	0.33

Table 2: Results for D-terminer and TermoUD applied to the ACL RD-TEC corpus. Three lists of TermoUD differ in the numbers of selected terms and are given for three consecutive C-values.

3.2 Genia

The GENIA corpus (Kim et al., 2003) consists of 2000 MEDLINE abstracts containing about 400K words. Included in the collection are articles containing such MeSH terms as human, blood cell and transcription factor. The annotation for biological terms refers to concepts defined within the GENIA ontology, which contains 47 biologically relevant nominal categories. From a linguistic point of view, the selected terms were nominal phrases in which the noun was followed by an optional sequence of adjectives and noun modifiers. There are about 80K annotated phrases in the corpus. The data is challenging because the structure of biological terms varies widely, and the vast majority of terms (76% of 36230) occur only once.

As with the previous data set, we used the D-terminer to identify terms in the Genia corpus. It performed very well, achieving an F1 value of 0.45 and finding almost 40% (13,487) of all annotated terms. As expected for such data (many singular oc-

currences), our program performed worse in terms of precision at the top of the returned list (Table 3). We tested phrases with lengths up to 4 and 6 elements. The method recognized 116,499 phrases with length up to 4 tokens. The list contains 71% of manually annotated terms. While the list of phrases with length up to 6 tokens is longer and consists of 153,122 elements, but it contains only slightly more of manually annotated phrases – 73%. The top of the lists (the first 8,986 elements) is the same for both tested lengths of phrases and the precision is 0.52. Significant differences appear in the terms that were placed in positions above 33,000.

	terms				
	selected	good	prec.	recall	F1
D-terminer	23,813	13,487	0.57	0.37	0.45
TermoUD					
4&6 top(1)	1173	740	0.63	0.02	0.04
4&6 top(2)	8,986	4,665	0.52	0.13	0.21
4 (1)	32,688	9,856	0.30	0.27	0.28
4 (2)	43,939	14,412	0.33	0.40	0.36
6 (1)	33,066	9,905	0.30	0.27	0.28
6 (2)	57,664	14,940	0.26	0.41	0.32

Table 3: Results for D-terminer and TermoUD applied to GENIA. The following results are reported for TermoUD: a) two results for the top parts of the lists (common for longer and shorter phrases) b) two results (consecutive C-values) for phrases of lenghts up to 4 and 6 which have the number of selected terms below and above the number of manually selected terms.

3.3 ACTER

The Annotated Corpora for Term Extraction Research (ACTER) dataset (Rigouts Terryn et al., 2022b) includes domain-specific corpora in three languages (English, French, and Dutch) and four domains (corruption, dressage (equitation), heart failure, and wind energy). Manual annotations are available for terms and Named Entities for each corpus, with almost 20k unique annotations in total. The corpus was used by those participating in the shared task at the Computerm workshop (Rigouts Terryn et al., 2020). The best result was achieved by a BERT based architecture used for sequential token classification, TALN-LS2N (Hazem et al., 2020). As the corpus contains a relatively short texts, our ranking methods are not too efficient, so the results for the top part of the candidate terms lists, shown in Table 4, are significantly worse (similar to the results of the standard methods taking part in the shared task). The worst result were obtained for Dutch texts. The TALN-LS2N results demonstrate some differences in the data across languages. For English, recall is much higher than precision, while for French they are similar. In our experiment, for English data, precision was higher than recall. The D-terminer application was trained on ACTER data, so we cannot use it as a comparison.

	terms					
	all	selected	good	prec.	recall	F1
TermoUD, English						
corp	1087	1008	245	0.24	0.23	0.23
equi	1427	661	255	0.39	0.18	0.24
htfl	2459	3466	494	0.14	0.20	0.17
wind	1434	1028	282	0.27	0.20	0.23
TermoUD, French						
corp	1103	1230	245	0.20	0.22	0.21
equi	1079	619	192	0.31	0.18	0.23
htfl	2202	3305	453	0.14	0.21	0.16
wind	870	840	152	0.18	0.17	0.18
TermoUD, Dutch						
corp	1215	845	161	0.19	0.13	0.16
equi	1457	1673	182	0.11	0.12	0.12
htfl	2137	2586	193	0.07	0.09	0.08
wind	1159	735	96	0.13	0.08	0.10
TALN-LS2N, English						
htfl	2479	-	-	0.35	0.71	0.47
TALN-LS2N, French						
htfl	2220	-	-	0.46	0.52	0.48

Table 4: Results for ACTER data. The 'all' column represents the number of different terms annotated within the data. htfl data was used as test data for TALN-LS2N while the other sets were used as train data. In the TALN-LS2N approach there was no initially selected list of terms – all tokens were tested.

3.4 RSDO5

The Slovenian corpus RDSO5 (Jemec Tomazin et al., 2021) was created to train tools for automatic term identification. It consists of around 250,000 tokens and gathers texts from four domains: biomechanics, linguistics, chemistry, and veterinary. 38,000 phrases were manually marked as terms in the data, among them 6165 different strings. Slovenian is an inflectional language, which means that each term may occur in many forms, e.g. *virusni sev* 'virus strain' has the following inflected forms in the data: *virusnih sevov, virusni sevi, virusnimi sevi, virusnim sevom, virusni sev, virusnega seva, virusnih sevih, virusnemu sevu, virusna seva*. TermoUD gives a list of unique terms as its output, so we join various manually selected term forms with the help of the lemmas provided by the Stanza parser. As a result, we obtain a list of 4200 items.

Table 5 contains the results of applying the TermoUD tool to four subcorpora of the RSDO5 cor-

pus. The third column of Table 5 gives the number of various terms (not term forms) which are identified in the data. For the evaluation we took lists of terms that have a length equal to the list of manually selected phrases. For this reason, the values of the precision, recall and F1 measure are equal. In the table, we give only the first value.

	tokens	diff. terms	prec. (nb)
bim	61,375	797	0.21 (169)
jez	109,421	1000	0.25 (249)
kem	65,620	773	0.24 (186)
vet	76,138	1630	0.21 (349)

Table 5: Results for TermoUD applied to RSDO5

We are not aware of other experiments performed on the RSDO5 corpus, so we have no data to evaluate the quality of TermoUD's performance. We decide to compare the obtained results with the free trial Skech Engine (Jakubíček et al., 2014) which gives the first 100 one-word terms and the first 100 multi-word terms, and almost all of them are unique terms. We select the same number of terms in the same proportion from our lists. The results of selected terms are given in Table 6. A comparison of the results shows that TermoUD is better at providing the first 100 one- and multi-words terms. Only the results for multi-word terms of biomechanic texts are at a similar level.

	Sketch Engine		TermoUD	
	one	multi	one	multi
bim	0.13	0.17	0.45	0.19
jez	0.26	0.31	0.42	0.47
kem	0.23	0.17	0.45	0.38
vet	0.14	0.24	0.53	0.46

Table 6: Precision of 100 extracted terms by Sketch Engine and TermoUD.

4 Conclusion and Future Work

TermoUD's method of selecting traditional candidate terms restricted to nominal phrases allows multiple languages to be processed with the same tool. As linguistic knowledge is already contained in the UD parsers, no language adjustments are needed. For example, it is irrelevant whether adjectives can come before or after a noun in a given language. An additional, unique feature of UD-based candidate term selection is its ability to extract discontinuous phrases, see Figure 1.

The best current methods of terminology extraction use machine learning and sequential tagging. The results obtained by these methods are much better than TermoUD's, especially measured by precision. These methods also facilitate the expansion of term types, e.g. to include those which are adjectives and verbs. However, the methods require the preparation of training data, that exists for only a few languages, text types and domains.

The quality of the results obtained by TermoUD depends on the quality of the parser for the language in question, especially how good the lemmatisation is. This feature is particularly important for languages with rich morphology, as we need to recognise and join various inflected forms of candidate terms.

The important feature that differentiates the two approaches is the list of results. For the classification-based methods, we get a list of terms, whereas TermoUD generates a sorted list of all term candidates. The disadvantage of the TermoUD tool is the need to establish where the list is divided into terms and non-terms, but the advantage is we can choose how many of the candidates we would like to choose. Machine learning methods only provide a list of accepted terms, which is fragmentary knowledge as we do not know the phrases that were rejected and should have been classified as terms. As the ranking coefficient used in TermoUD is highly dependent on term frequencies, our method gives much better results for larger data. Terms used in text only once are always located very low on the final list.

In the near future, we plan to deal with the analysis of coordinated phrases, which are quite a challenge for all terminology extraction tools and the UD mechanism seems to enable their correct handling. Moreover we should test the tool on languages from other families, and improve the term ordering method. as our lists contain on average about 80% of terms, changes in ordering method may significantly improve the results.

TermoUD[1] is available from `http://zil.ipipan.waw.pl/TermoPL`, the same page as TermoPL, the previous version of the tool described in the paper. TermoPL is also a part of Korpusomat (Kieraś and Kobyliński, 2021), a simple tool for creating linguistic corpora in Polish `https://korpusomat.pl/`. TermoUD will be available from the multilingual version of the Korpusomat tool currently under development.

[1]A system demonstration and the results of the tool are available from the same page.

5 Limitations

ThermoUD requires the existence of a UD parser. It does not consider candidate terms like adjectives, verbs, coordinated phrases and phrases containing coordinated phrases. We only evaluated the tool on Indo-European languages as we are not aware of any terminology-annotated datasets for other languages. We used our extraction method to extract terminology from Finish texts. Used texts and obtained results are available on the project page given above.

References

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257 – 289.

Damien Cram and Beatrice Daille. 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.

Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. 2000. Automatic recognition of multi-word terms: the C-value/NC-value method. *Int. Journal on Digital Libraries*, 3:115–130.

Katrin Fundel, Robert Küffner, and Ralf Zimmer. 2006. RelEx—Relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.

Pablo Gamallo. 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.

ZhiQiang Geng, GuoFei Chen, YongMing Han, Gang Lu, and Fang Li. 2020. Semantic relation extraction using sequential and tree-structured LSTM with attention. *Information Sciences*, 509:183 – 192.

Amir Hazem, Mérieme Bouhandi, Florian Boudin, and Beatrice Daille. 2020. TermEval 2020: TALN-LS2N system for automatic term extraction. In *Proceedings of the 6th International Workshop on Computational Terminology*, pages 95–100, Marseille, France. European Language Resources Association.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*.

Miloš Jakubíček, Adam Kilgarriff, Vojtěch Kovář, Pavel Rychlý, and Vít Suchomel. 2014. Finding terms in corpora for many languages with the Sketch Engine. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 53–56, Gothenburg, Sweden. Association for Computational Linguistics.

Mateja Jemec Tomazin, Mitja Trojar, Simon Atelšek, Tanja Fajfar, Tomaž Erjavec, and Mojca Žagar Karer. 2021. Corpus of term-annotated texts RSDO5 1.1. Slovenian language resource repository CLARIN.SI.

Witold Kieraś and Łukasz Kobyliński. 2021. Korpusomat – stan obecny i przyszłość projektu. *Język Polski*, (2):49–58.

J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

Ying Liu, Tianlin Zhang, Pei Quan, Yueran Wen, Kaichao Wu, and Hongbo He. 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*, pages 796–802. Springer, Cham.

Małgorzata Marciniak, Agnieszka Mykowiecka, and Piotr Rychlik. 2016. TermoPL — a flexible tool for terminology extraction. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation, LREC 2016*, pages 2278–2284, Portorož, Slovenia. ELRA, European Language Resources Association (ELRA).

Małgorzata Marciniak, Piotr Rychlik, and Agnieszka Mykowiecka. 2020. Supporting terminology extraction with dependency parses. In *Proceedings of the 6th International Workshop on Computational Terminology*, pages 72–79, Marseille, France. European Language Resources Association.

Behrang QasemiZadeh and Anne-Kathrin Schumann. 2016. The ACL RD-TEC 2.0: A language resource for evaluating term extraction and entity recognition methods. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1862–1868, Portorož, Slovenia. European Language Resources Association (ELRA).

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Ayla Rigouts Terryn. 2021. *D-TERMINE : data-driven term extraction methodologies investigated*. Ph.D. thesis.

Ayla Rigouts Terryn, Veronique Hoste, Patrick Drouin, and Els Lefever. 2020. TermEval 2020: Shared task on automatic term extraction using the annotated corpora for term extraction research (ACTER) dataset. In *Proceedings of the 6th International Workshop on Computational Terminology*, pages 85–94, Marseille, France. European Language Resources Association.

Ayla Rigouts Terryn, Veronique Hoste, and Els Lefever. 2022a. D-terminer: Online demo for monolingual and bilingual automatic term extraction. In *Proceedings of the Workshop on Terminology in the 21st century: many faces, many places*, pages 33–40, Marseille, France. European Language Resources Association.

Ayla Rigouts Terryn, Véronique Hoste, and Els Lefever. 2022b. Acter 1.5: Annotated corpora for term extraction research.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text Mining: Applications and Theory*, pages 1 – 20.

Piotr Rychlik, Małgorzata Marciniak, and Agnieszka Mykowiecka. 2022. Termopl: A tool for extracting and clustering domain related terms. In *Proceedings of the 22nd ACM/IEEE Joint Conference on Digital Libraries*, JCDL '22, New York, NY, USA. Association for Computing Machinery.

Gerard Salton. 1988. Syntactic approaches to automatic book indexing. In *Proceedings of the 26th Annual Meeting on Association for Computational Linguistics*, ACL '88, pages 204–210, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ayla Rigouts Terryn, Véronique Hoste, and Els Lefever. 2020. In no uncertain terms: a dataset for monolingual and multilingual automatic term extraction from comparable corpora. *Language Resources and Evaluation*.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinkova, Jan Hajic jr., Jaroslava Hlavacova, Václava Kettnerová, Zdenka Uresova, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria dePaiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonca, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. Conll 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

Ziqi Zhang, Jie Gao, and Fabio Ciravegna. 2016. JATE 2.0: Java automatic term extraction with Apache Solr. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2262–2269, Portorož, Slovenia. European Language Resources Association (ELRA).

A Candidate phrase extraction from UD structures (pseudocode for the algorithm)

1. Define sets:

`head-pos`, `non-head-pos`, `obligatory-rel`, `facultative-rel`.

2. Create the structure `ud`: for every token `j` in a sentence create a node n_j and define `ud[`n_j`]` as a set of pairs (`n`$_i$`,r`$_i$), where `n`$_i$ is the dependant (in the sense of the dependency relation `r`$_i$) of `n`$_j$ and corresponds to the token `i`.

3. Create the list `T-nodes` consisting of all nodes that might be included in term phrases. `T-nodes` will contain all nodes from `ud` with POS in `head-pos` or `non-head-pos`.

4. Identify the structures corresponding to hyphenated compound words. Add all nodes from this structures to `T-nodes`. Select all nodes from the identified structures that are not their heads and put them in the list `H-nodes`.

5. Remove nodes from the structure `ud` that are not in the list `T-nodes`.

6. Check if obligatory relations lead to the nodes that may create terms:

for each element `e` of `T-nodes`:
  for each pair `(n,r)` $\in$ `ud[e]`:
    if `r` $\in$ `obligatory-rel`:
      if `n` $\notin$ `T-nodes`:
        delete `e` from `T-nodes`;
    else: # `r` *is facultative*
      if *n* $\notin$ `T-nodes`:
        delete pair `(n,r)` from `ud[e]`;

7. For each node `n` in `T-nodes`, create an empty set `P[n]`. This set will contain lists of all possible phrases `p[d]` for which `d` is the head element, for all dependants `d` of `n`. These phrases will be represented by sets of nodes. After determining the

set P[n], the list p[n] can be created. Each phrase from p[n] must contain:

(a) node n,
(b) all nodes from one phrase in p[d],
    if (d,r) ∈ ud[h] and
    r ∈ obligatory-rel,
(c) none or all nodes from one phrase in p[d],
    if (d,r) ∈ ud[h] and
    r ∈ facultative-rel,
(d) all nodes x ∈ H-nodes, if (x,r) ∈ ud[n].

8. Select candidates for terminology phrases:

```
create empty list terms;
while T-nodes is not empty:
  for each n ∈ T-nodes:
    if ud[n] is empty:
      # phrases are established for all
      # dependent nodes of n
      remove n from T-nodes;
      create p[n];
      find the head node h of n;
      if h∈T-nodes:
        add p[n] to P[h];
        remove all pairs (n,r) from ud[h];
      if POS(n) ∈ head-pos:
        add all phrases from p[n] to terms;
```

9. Clean up:

(a) sort each element of terms according to the position of the nodes in the sentence,
(b) remove the node from the begining of a phrase, if it is a preposition referring to the head of the sentence, or if it is a determiner.