Semantic relation extraction from Karst data

Mark Žakelj[1], Luka Keserič[2], Aleš Kert[3], Miha Šemen[4]

Abstract

In this report we describe or approaches for extracting hyponym-hypernym and non-hierarchical relationship from a definition that's either unmarked or has pre-selected regions of interest. The first task is tackled using a token classification model, generating labels for the definienda and the genuses, and connecting them in a semantic network. The second task is tackled using two approaches - one where regions are already known, using an R-BERT model, and the other where we're looking for possible relations in a sentence. The models were evaluated on the Termframe Karstology corpora and the SemEval 2010 Task 8 corpus.

Keywords

BERT, Relationship extraction, Hyponym-hypernym, Semantic relations, Semantic network

[1] *mz5635@student.uni-lj.si, 63170037*
[2] *lk6760@student.uni-lj.si, 63210486*
[3] *ak8754@student.uni-lj.si, 63170010*
[4] *ms9232@student.uni-lj.si, 18203183*

Introduction

Relationship extraction is a problem in natural language processing that aims to find relations between different words or sequences of words in a corpus. We developed a model for the rather specific domain of karstology and only used definitions, where one word or short phrase is being defined. The relations in ground truth data were all based on the phrase being defined (definiendum).

One type of relation is the hypernym-hyponym relation, which is a hierarchical relation. The other type of relationship is non-hierarchical relation. We prepared two different training and testing sets, one including hierarchical and the other including non-hierarchical relations. We also prepared different data sets for English and Slovene corpora.

We used BERT based models for both languages and both types of relations. We tested two approaches, one where we used BERT for token classification and the other, where we assumed the regions of interest and only used the relation classification (extraction) model.

Related work

Our sequence labeling approach is similar to Named entity recognition and POS tagging tasks. These approaches have been already used for definition extraction in [1], where Bert based Named entity recognition has been fine-tuned for the task. In similar fashion, [2] has used Roberta to generate embeddings and then train CRF on top of it to predict the most probable sequence of tags.

SemEval2020 task 6 [3] has also proposed a problem of definition extraction which is similar to our problem, but on a general language domain.

Data preparation

Termframe corpus

The Termframe corpus contains definitions extracted from Karst-related scientific literature, which allows us to analyze the relationships between different Karst-specific terms, constructing a semantic network in the process. The corpus contains definitions from three different languages - English, Slovene, and Croatian - of which we use only the former two. The English corpus contains 745 definitions and the Slovene corpus contains 787 definitions. The definitions have been annotated and so contain data about the definienda, which are the core of the definition, and the various terms that are in some sort of a relation with the definienda.

Additionally, the students from the Faculty of Arts constructed a set of new test corpora in English and Slovene, the former containing 103 new definitions and the latter 100. These were used in the evaluation process, while the original corpora were used for training our models. After the automatic evaluation on the test corpora, the results were sent for manual evaluation to our colleague on the Faculty of Arts.

Preparation

For the hypernym-hyponym extraction (task 1), we used only the hierarchical tags, namely *DEFINIENDUM* and *GENUS*, which represent hypernym-hyponym relation in a definition. Each word was assigned either the relevant hierarchical tag or the *OTHER* tag, as seen in 1.

For the non-hierarchical relation extraction (task 2) in the first approach we transformed the corpora into the following shape - each relation within each sentence was given a separate row. The relations were defined between the definienda and the non-hierarchically related phrases. The definienda were marked as entity 1 and the related phrase as entity 2.

Sentence	Word	Tag
...	...	...
6	periglacial	B-DEFINIENDUM
6	environment	I-DEFINIENDUM
6	is	O
6	defined	O
6	as	O
6	any	O
6	place	B-GENUS
6	where	O
...	...	...

Table 1. Few lines of the token classification output

```
HAS_LOCATION <e1> cavern </e1> - a very large
    chamber <e2> within a cave </e2> .
```

We only used the non-hierarchical tags, namely *HAS_CAUSE*, *HAS_LOCATION*, *HAS_FORM*, *HAS_FUNCTION*, *HAS_SIZE*, *COMPOSITION*

For the second approach,

Dataset split

We were provided with 2 corpora - one intended for training and the other for testing. The training corpus was further split 90/10, to get a training/validation split. For the English corpora this meant 670 definitions in the training set, 75 in the validation set, and 103 in the test set. We also evaluated our models on the Slovene corpora resulting in a split of 708 definition in the training set, 79 in the validation set, and 100 in the test set.

Methodology

Hyponym-hypernym extraction

We approached the problem of hypernym-hyponym relation extraction as a token classification problem, where we assign a label to each token in the sentence.

This classification was done using the BertForTokenClassification from the HuggingFace framework, that has a linear layer on top of the BERT hidden-state output, which was fine-tuned the tokenized sentences from the training set.

Our approach thus generated a list of predicted token tags. Because the BERT tokenizer splits words into subwords, we further had to implement an algorithm that combines the tokens into their original form, which meant combining the tokens containing the '##' string with its previous neighbour.

To acquire the detected definienda and the genuses, these sentences were masked using their predicted tags, leaving us with a collection of words belonging to either class. These were then grouped up by combining all words belonging to the same tag that neighbour eachother, creating a list of definienda and genuses present in the sentence. These terms were then assigned their own node number, and a directed edge between each definiendum - genus pair was added to construct semantic network.

Non-hierarchical relations extraction

The second task, the extraction of semantic relations between two entities in a sentence, was approached using two different approaches with different starting assumptions. The first approach presumed that the entities in question are already known (regions in the sentence) and our task is to identify the type of semantic relation that exists between them. The second approach didn't make such an assumption, instead presuming an unmarked definition is given.

The first aproach uses the R-BERT architecture, described in [4], implemented in [5], and can be seen in Figure 1, which tackles this problem by introducing special tokens which are used as entity delimiters. This modified tokenized sentenced is sent through a pre-trained BERT model, giving a hidden output. This output is then combined into 3 vectors - one containing the output of the first, CLS token, one containing the average of outputs of the first entity and one of the second entity. These vectors are concatenated and sent through a fully-connected layer, whose outputs are then sent through a softmax activation function, giving us the prediction probabilities.

The second approach is similiar to the first, except that it doesn't presume the entities (regions) in interest are already known. Here we evaluated two different methods, the first is similiar to the hyponym-hypernym extraction, where we seek to classify tokens belonging to certain entities by using Bert for token classification. The second method instead finds only the definiendum of a definition, and then uses a growing sliding window and R-BERT to determine entities (regions) with the best possible relation.

The first approach seeks to use the same method that worked on the hierarchical task, but instead of labeling only 2 types of tags, it instead labels 6 non-hierarchical tags, together with the definiendum. The resulting classification is then grouped, like in the previous task, and semantic relations are then added to the semantic network, in the direction from the definiendum to the entity, with the edge representing the non-hierarchical relation presented by the predicted tag.

The second method aims to take advantage of the near perfect results the hyponym-hypernym extraction model achieved when detecting the definienda. It seeks to incorporate the two already described models, token classification and R-BERT, by first detecting the definiendum in a definition, which represent the first entity, and then iteratively looking for the best fit for the second entity by incorporating a growing sliding window. The growing sliding window works by placing the second entity at the beginning of a sentence, with no overlap with the first, and sliding to towards the end of the sentence, applying the already trained R-BERT model. During each step we look at the outputs of the layer (logits) and say that a possible detection has occured, when a logit is larger than a certain threshold. At this point, the window size starts increasing, with the growing continuing until the value of the logit has decreased dramatically, or a detection of a different relation has occured, at which point the semantic relation is saved and

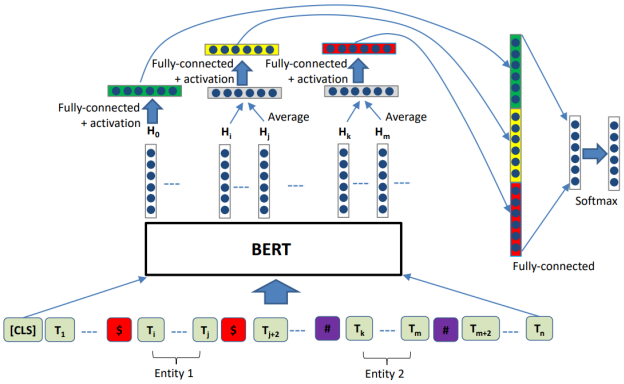

Figure 1. R-BERT architecture. Introduces special tokens as entity delimiters. Combines the hidden outputs of the BERT model and applies a fully connected layer on them. Used for the second task. Acquired from [4].

the sliding window continues from the location of the penultimate token in the window that failed to meet the threshold. This method was evaluated by classifying the tokens within a relation with its corresponding tag and comparing it with the ground truth.

Results

Hyponym-hypernym extraction

For the first task, the previously described model was evaluated different BERT models. Specifically, for the English corpora, the cased BERT [6] and the SciBERT [7] pre-trained models were used. For the Slovene corpora we first evaluated the English-trained cased BERT [6] model as a benchmark, and additionally used the SloBERTa [8] and the CroSloEngual [9] models, both available on the HuggingFace framework.

The results of this approach can be seen in Table 2 for the English corpora and in Table 3 for the Slovene corpora. It turns out that classifying the definiendum in a definition is a much easier task than identifying the genus, with the F1-score being a third higher.

We also notice that while the BERT cased [6] model is better at finding the genus in a sentence, the SciBERT [7] is better at identifying the definienda, which could be useful when looking for various terms that are related to the definiendum, as we do in the latter task. In the Slovene models we notice that SloBERTa [8] isn't able to give use very good results on the given corpora, falling short even when compared to a cross-lingual BERT cased model [6]. On the other hand the CroSloEngual model [9] is able to give fairly decent results on the Slovene corpus.

All of the models were trained for 4 epochs with a batch size of 4 and a max length of 128.

Although the model identified all the definienda in the first thirty sentences, which were usually at the beginning, there were some deviations. Depending on the context the definiendum was usually preceded by articles, such as *a*, *an*

	BERT Cased		
	Precision	Recall	F1-score
DEFINIENDUM	0.95	0.96	0.96
GENUS	0.63	0.66	0.64
Macro average	0.79	0.81	0.80
	SciBERT		
	Precision	Recall	F1-score
DEFINIENDUM	0.98	0.98	0.98
GENUS	0.58	0.60	0.59
Macro average	0.78	0.79	0.79

Table 2. Token classification model performance on the English corpora for the hypernym-hyponym relation extraction.

	BERT Cased		
	Precision	Recall	F1-score
DEFINIENDUM	0.82	0.78	0.80
GENUS	0.54	0.43	0.48
Macro average	0.67	0.61	0.64
	SloBERTa		
	Precision	Recall	F1-score
DEFINIENDUM	0.49	0.49	0.49
GENUS	0.30	0.25	0.27
Macro average	0.39	0.37	0.38
	CroSloEngual		
	Precision	Recall	F1-score
DEFINIENDUM	0.95	0.93	0.94
GENUS	0.63	0.57	0.60
Macro average	0.79	0.75	0.77

Table 3. Token classification model performance on the Slovene corpora for the hyponym-hypernym relation extraction.

or *the*. *The* , also known as a definite article, is used to refer to specific or particular nouns whereas *a*/*an*, also known as an indefinite article, is used to modify non-specific or non-particular nouns. Eleven out of the thirty sentences started with an article followed by the definiendum.

Interestingly, there are no certain reasons whatsoever why the model chose to include the articles as they do not bring anything to the meaning of the definiendum itself. What is even more fascinating is that the model did not have any problems detecting the definienda, even if they were composed of several words. If we strictly focus on the efficiency of the model finding only the correct definiendum (without the article) the accuracy for the first 30 sentences lies at 93.3%.

The model also showed difficulties in finding the right genus or hypernym. The most common mistakes were including the conjunction, including words not part of the genus, misrecognizing the genus, and failing to recognize genus in the sentence.

Those kinds of mistakes were expected, as also the students from the Faculty of Arts had some problems recognizing and finding the right genus. As the definienda were mostly at

the beginning of the sentence, the genuses did not follow those structures as they could have been almost anywhere in the sentence. It is therefore that the model had certain problems with finding the right genus thus leading to the above-mentioned mistakes.

Non-hierarchical relations extraction

The results for the first approach, where entities (regions) are known in advance, used to tackle this problem can be seen in Table 4 for the English corpora and in Table 5. The models used in this task are similiar to the previous task, except SloBERTa [8], which isn't used due to poor previous results on our corpora. All of the models were trained for 6 epochs, with a batch size of 4, and a maximal sentence length of 128.

On the English corpora, the best performing model is the SciBERT nodel [7], winning over the BERT cased model [6] with a narrow margin. On the Slovene corpora, the CroSloEngual model [9] gives the best performance, scoring slightly better than even the English models.

Additionally, we evaluated this model on the SemEval 2010 Task 8 [10], which contains various relations between entities, such as message-topic, cause-effect, entity-origin, etc., which aren't domain-specific. The model was fine-tuned on the specified corpus for 8 epochs, achieving an accuracy of 85.3% and an F1-score of 0.889.

	R-BERT with BERT Cased		
	Precision	Recall	F1-score
HAS_CAUSE	0.92	0.8	0.86
HAS_LOCATION	0.65	0.63	0.64
HAS_FORM	0.62	0.75	0.68
COMPOSITION	0.89	0.96	0.92
HAS_FUNCTION	0.77	0.71	0.74
HAS_SIZE	1.0	1.0	1.0
Macro average	0.81	0.81	0.81
	R-BERT with SciBERT		
	Precision	Recall	F1-score
HAS_CAUSE	0.86	0.83	0.85
HAS_LOCATION	0.79	0.81	0.8
HAS_FORM	0.71	0.85	0.77
COMPOSITION	0.96	0.92	0.94
HAS_FUNCTION	0.73	0.57	0.64
HAS_SIZE	1.0	1.0	1.0
Macro average	0.84	0.83	0.83

Table 4. Known-entity R-BERT model performance on the English corpora for semantic relation extraction.

Due to the more difficult nature of the second task, we used a different evaluation method than in the first task, where we evaluated correctly identified sequences of tokens. In the second task, we used a token-by-token comparison to calculate our scores, because the entities are considerably longer and thus a single misidentified token causes the sequence to be an invalid detection, which impedes our ability to compare different approaches.

	R-BERT with BERT Cased		
	Precision	Recall	F1-score
HAS_CAUSE	0.68	0.65	0.67
HAS_LOCATION	0.73	0.9	0.81
HAS_FORM	0.8	0.76	0.78
COMPOSITION	0.3	0.5	0.37
HAS_FUNCTION	1.0	0.5	0.67
HAS_SIZE	0.85	0.61	0.71
Macro average	0.73	0.65	0.67
	R-BERT with CroSloEngual		
	Precision	Recall	F1-score
HAS_CAUSE	0.91	0.91	0.91
HAS_LOCATION	0.96	0.87	0.91
HAS_FORM	0.82	0.86	0.84
COMPOSITION	0.62	0.83	0.71
HAS_FUNCTION	0.83	0.83	0.83
HAS_SIZE	0.88	0.83	0.86
Macro average	0.84	0.86	0.85

Table 5. Known-entity R-BERT model performance on the Slovene corpora for semantic relation extraction. SloBERTa not included in this evaluation.

For the second approach we also limited ourselves to the SciBERT model [7] for the English corpora, because of its superior ability to recognize definienda, and the CroSloEngual [9] model, because it is the best performing model on the Slovene corpora. These models follow the R-BERT architecture. The growing sliding window uses a fixed threshold of 7.

The results for the first method, token classification, can be seen in Table 6, and for the growing sliding window in Table 7. As can be seen from our experiments, the model that uses token classification is better able to recognize words belonging to a certain relation, than the growing sliding window model. Although on the Slovene corpora, the growing sliding window model is able to produce a larger recall score with certain relations, such as composition, form, and location.

Additional evaluation

Hierarchical relations are evaluated using exact entity match, meaning that if for example we correctly detected 3 words of a 4 word *GENUS*, this is considered a wrong prediction. If we instead considered *GENUSES*, where at least 3/4 of the words inside the phrase are correctly identified, as correct, the described example would be considered as a right prediction. We considered every possible threshold level from 0 to 1 and plotted the curve. We calculated the recall curve and precision curve. For recall, we took the ground truth regions and checked how much does annotation cover the ground truth. For precision, we took annotation regions and checked how much does the ground truth cover the annotation region. This is better shown at the example 8.

In the appendix we present the precision and recall graphs for hierarchical and non-hierarchical tasks, both evaluated us-

	SciBERT		
	Precision	Recall	F1-score
DEFINIENDUM	0.96	1.00	0.98
HAS_CAUSE	0.56	0.76	0.64
HAS_LOCATION	0.63	0.62	0.63
HAS_FORM	0.27	0.61	0.38
COMPOSITION	0.74	0.63	0.68
HAS_FUNCTION	0.36	0.16	0.23
HAS_SIZE	0.63	0.79	0.70
Macro average	0.62	0.66	0.62
	CroSloEngual		
	Precision	Recall	F1-score
DEFINIENDUM	0.90	0.90	0.90
HAS_CAUSE	0.38	0.70	0.50
HAS_LOCATION	0.64	0.68	0.66
HAS_FORM	0.50	0.60	0.54
COMPOSITION	0.27	0.58	0.37
HAS_FUNCTION	0.24	0.55	0.34
HAS_SIZE	0.50	0.53	0.51
Macro average	0.54	0.66	0.58

Table 6. Token classification evaluation on a token-by-token basis. SciBERT used on English corpora, CroSloEngual on Slovene.

	R-Bert with SciBERT		
	Precision	Recall	F1-score
HAS_CAUSE	0.55	0.40	0.46
HAS_LOCATION	0.44	0.43	0.43
HAS_FORM	0.27	0.44	0.33
COMPOSITION	0.71	0.58	0.64
HAS_FUNCTION	0.32	0.07	0.12
HAS_SIZE	0.52	0.78	0.62
Macro average	0.56	0.55	0.54
	R-BERT with CroSloEngual		
	Precision	Recall	F1-score
HAS_CAUSE	0.26	0.74	0.39
HAS_LOCATION	0.39	0.70	0.50
HAS_FORM	0.25	0.77	0.38
COMPOSITION	0.20	0.75	0.31
HAS_FUNCTION	0.16	0.36	0.23
HAS_SIZE	0.45	0.49	0.47
Macro average	0.45	0.64	0.47

Table 7. Growing sliding window evaluation on a token-by-token basis. SciBERT used on English corpora, CroSloEngual on Slovene.

Word	Truth	Annotation
word0	O	O
word1	GENUS	O
word2	GENUS	GENUS
word3	GENUS	GENUS
word4	GENUS	GENUS
word5	O	O

Table 8. One region example that is considered correct for precision thresholds less or equal to 1.0 (3/3) and for recall thresholds less or equal to 0.75 (3/4)

ing the SciBert token classification model 2. When comparin the tags, we did not differentiate between B- and I- prefixes.

We can interpret these graphs by comparing lines for one tag on precision and recall graphs. Having higher curve on the recall graph and lower on the precision means, that the model often labels additional neighbouring words of ground truth as the target class. Having lower recall curve and higher precision curve means, the model annotations don't cover the ground truth that well.

Semantic network generation

In order to generate a semantic network in a form of a graph as seen in Figures 4,5 and 6,7 in the appendix, we had to pre-process the predicted relations from the models with some predefined rules.

In order to achieve a higher connectivity of the network, we considered a node to be correctly annotated (green colour of nodes) by the model if the annotation of a phrase had at least one word that was correctly annotated. For multiword phrases that means at least one correctly identified word for a correct annotation and the same holds if multiple words were annotated by the model and only a partial overlap of them was correctly annotated (e.g. "soil or rock" as a phrase annotated as *GENUS*, while "soil" and "rock" are *GENUS* separately).

Since we modified how we consider the correctly annotated phrases, we also had to adapt the false annotations by the model (depicted with red colour of nodes). The phrase was identified as false if any of the words in a phrase were mislabelled, where we considered multiple instances of the falsely annotated phrase (e.g. as considered before, in the annotated phrase "soil or rock", the instance of "or" is considered as false as well as anything before and after it, since "soil" and "rock" are the ground truths).

Last type of nodes to consider are the ones depicting unidentified phrases, that the model did not label (labelled with grey colour of nodes) that we obtained by filtering correctly labelled phrases from the ground truth phrases.

Lastly, the edges of the network are directed edges, where the direction of the edge is indicated by an arrow pointing at a node at one of the two endpoints of the edge. The direction indicates the type of phrase (node) at each end specific for that relation, where the arrow always points from the *DEFINIENDUM* to the selected type of relation (e.g. *GENUS*, *HAS_SIZE*, etc.). So one node can be both pointed to and pointed from as well as point to itself (self loop in a graph), the latter is less likely since we usually do not define a phrase with itself. Some connections between the nodes occur more often so to keep the readability of the graph as high as possible and not display every edge, we labelled multiple connections with darker edges between the nodes (more edges, darker the edge colour).

Discussion

Our experiments have shown that for hyponym-hypernym extraction, the best approach is to use token classification to classify the definition into definienda and genuses. For English, the BERT cased model works the best, while for Slovene, the CroSloEngual model is superior to others.

Semantic relationship extraction can be viewed from two angles - one where the regions are predefined, the other where only definiendum is known. For the former it is best to approach it with an R-BERT model, with a SciBERT base for English and CroSloEngual for Slovene. For the latter, it turns out that token classification is the superior approach, if evaluating token-by-token wise, having an edge over the growing sliding window method.

Overall, we think that our methods are able to extract interesting data that is shown in the networks in Figures 4,5, 6,7, in the appendix. While the whole network looks unordered, the largest weakly connected components show, that the model is able to capture some interesting data about the relationships between terms, even forming hierarchical chains in some places.

There also exist some possible improvements to our models, namely in the hyponym-hypernym relationship extraction, where it has been noted that the model performance is hindered by the occasional inclusion of articles before the definienda, or the inclusion of conjunctions with the genuses, which could be improved by removing such words from the tags after the BERT model has made its prediction.

References

[1] TobiasLee. Defteval2020. https://github.com/TobiasLee/DeftEval2020, 2018.

[2] Andrei-Marius Avram, Dumitru-Clementin Cercel, and Costin Chiru. UPB at SemEval-2020 task 6: Pretrained language models for definition extraction. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, Barcelona (online), December 2020. International Committee for Computational Linguistics.

[3] Sasha Spala, Nicholas A Miller, Franck Dernoncourt, and Carl Dockhorn. Semeval-2020 task 6: Definition extraction from free text with the deft corpus, 2020.

[4] Shanchan Wu and Yifan He. Enriching pre-trained language model with entity information for relation classification. *CoRR*, abs/1905.08284, 2019.

[5] Jangwon Park. R-bert. https://github.com/monologg/R-BERT, 2020.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[7] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. In *EMNLP*. Association for Computational Linguistics, 2019.

[8] Matej Ulčar and Marko Robnik-Šikonja. Slovenian RoBERTa contextual embeddings model: SloBERTa 2.0, 2021. Slovenian language resource repository CLARIN.SI.

[9] M. Ulčar and M. Robnik-Šikonja. FinEst BERT and CroSloEngual BERT: less is more in multilingual models. In P Sojka, I Kopeček, K Pala, and A Horák, editors, *Text, Speech, and Dialogue TSD 2020*, volume 12284 of *Lecture Notes in Computer Science*. Springer, 2020.

[10] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. 2019.

Appendix

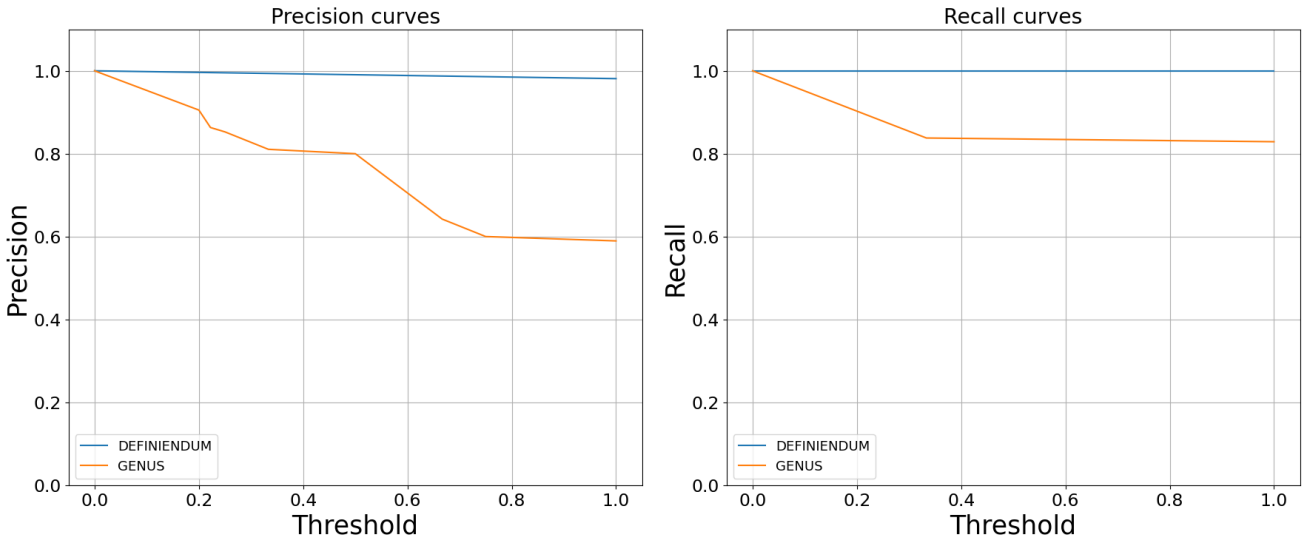

Figure 2. Precision and recall graphs of the SciBert token classification model evaluated on the English **hierarchical** test set.

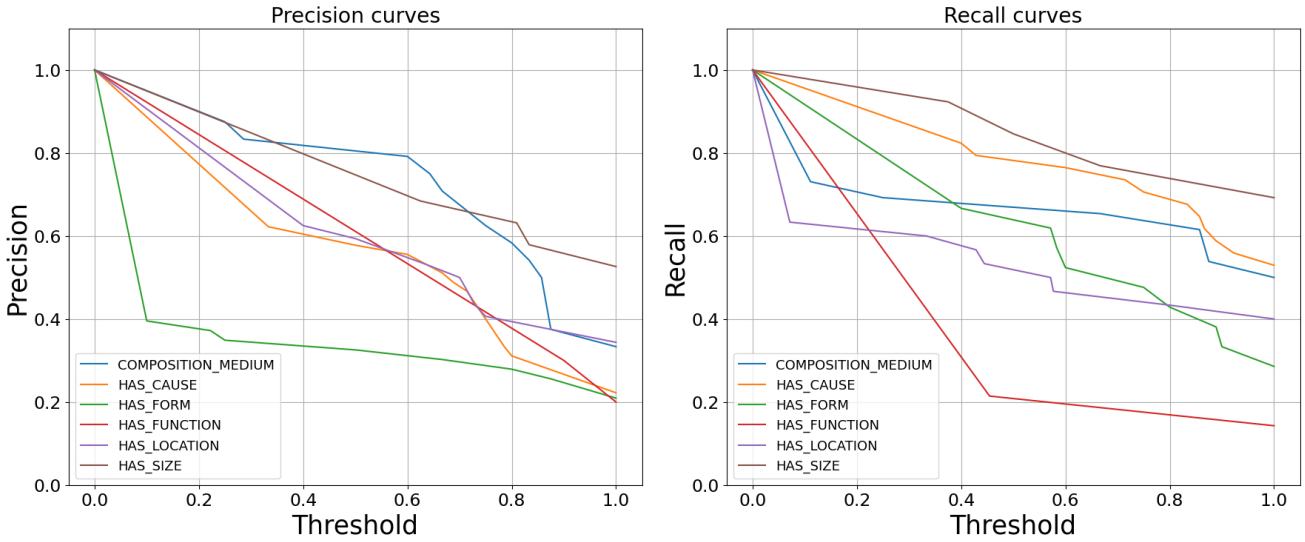

Figure 3. Precision and recall graphs of the SciBert token classification model evaluated on the English **non-hierarchical** test set.

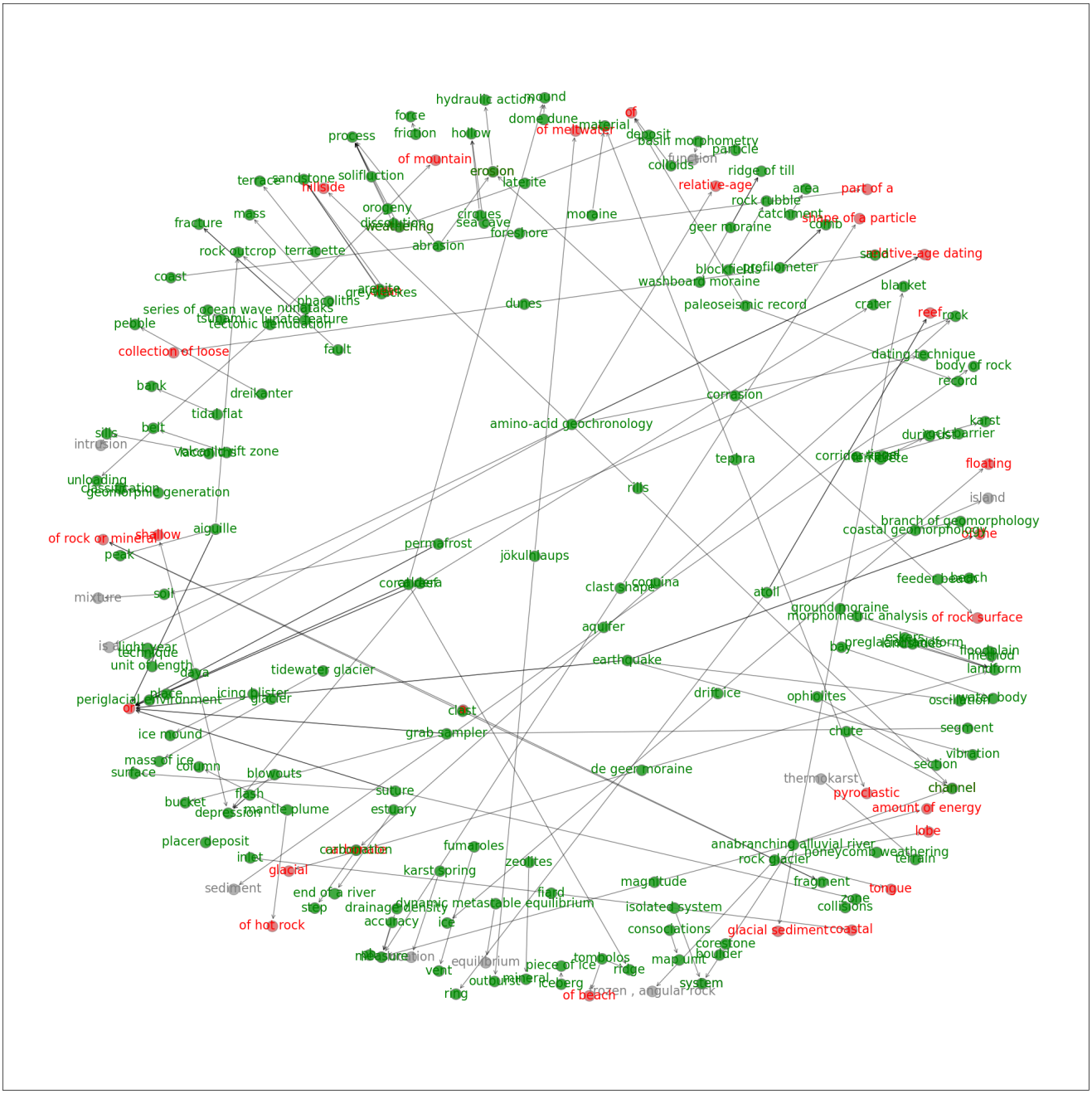

Figure 4. Extracted knowledge graph. This is the graph that we extracted from the English test data. For more information about the graph, please refer to the **Semantic network generation** subsection under the **Results** section.

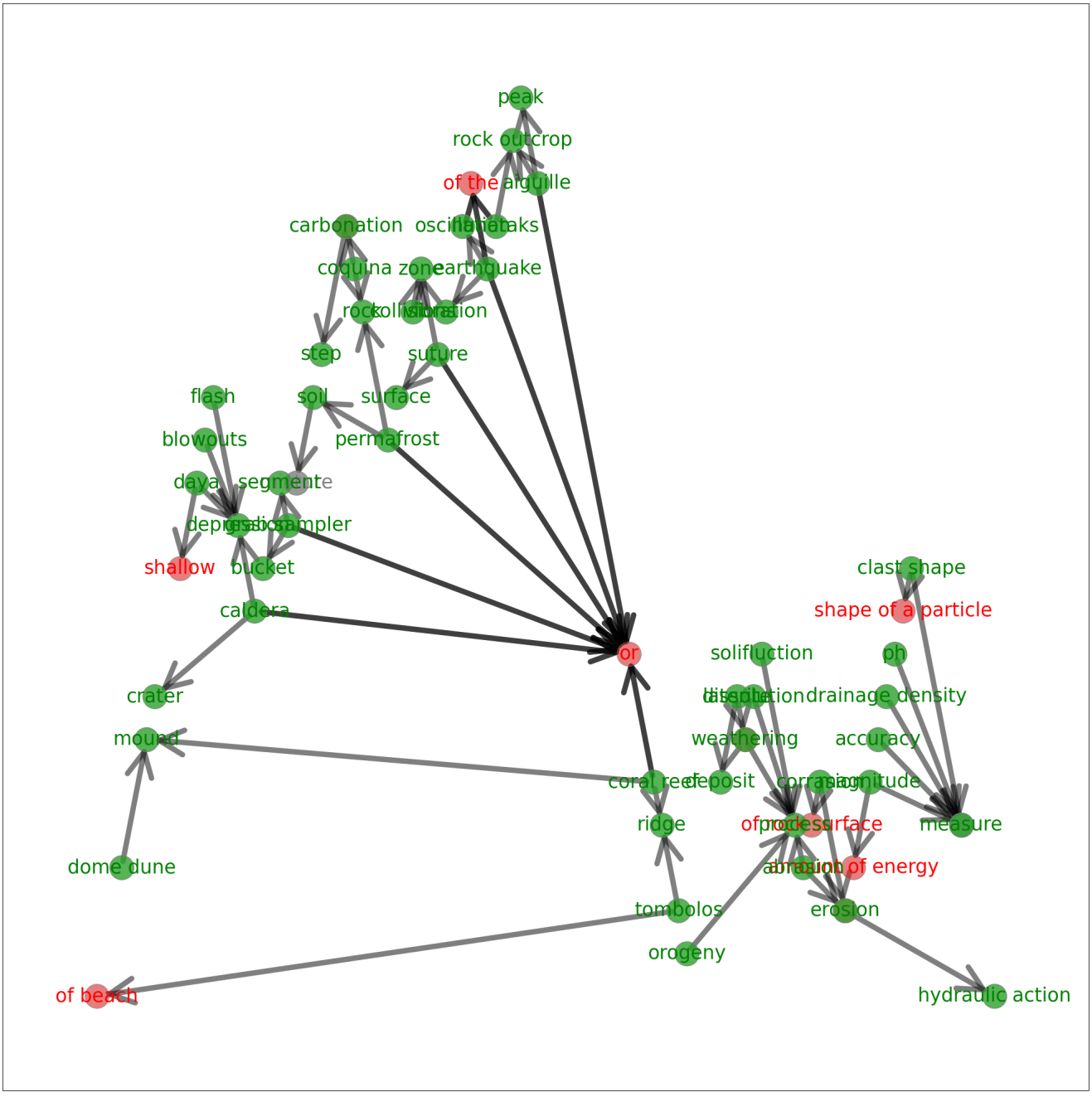

Figure 5. Largest connected components of extracted knowledge graph. This is the graph that we extracted from the English test data. It's limited in the number of phrases, since it displays only the largest three connected components of the network. For more information about the graph, please refer to the **Semantic network generation** subsection under the **Results** section.

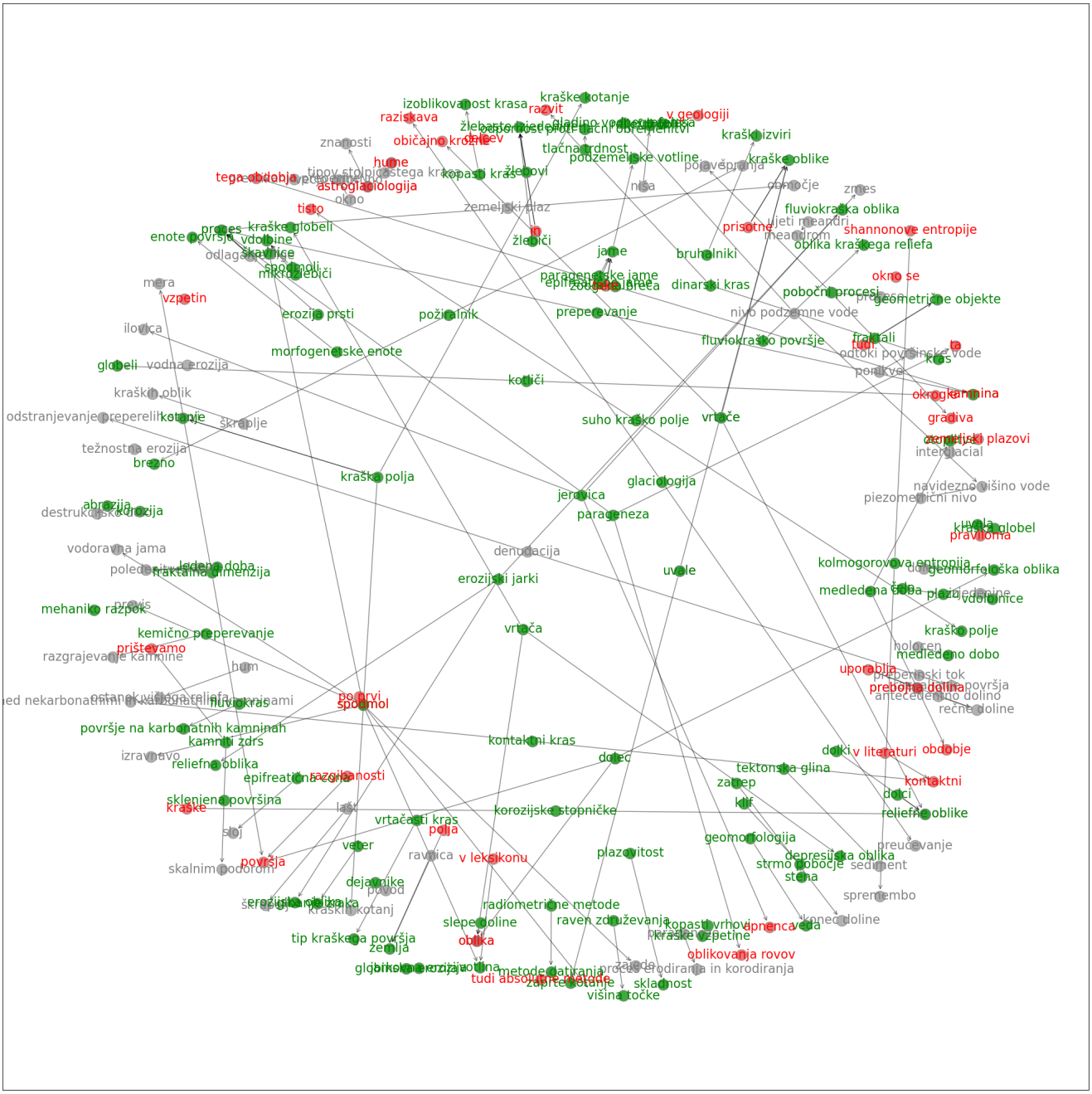

Figure 6. Extracted knowledge graph. This is the graph that we extracted from the Slovene test data. For more information about the graph, please refer to the **Semantic network generation** subsection under the **Results** section.

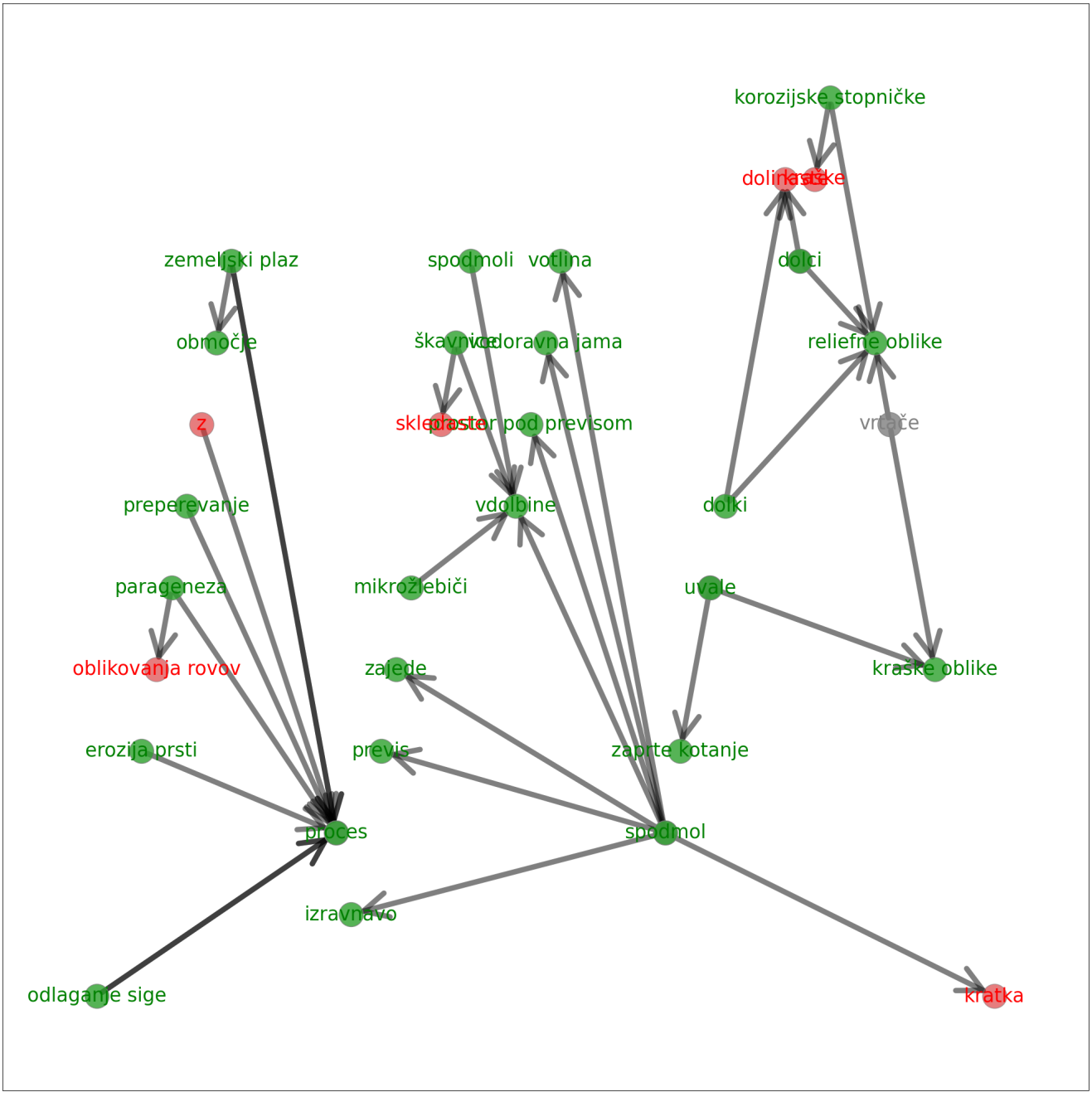

Figure 7. Largest connected components of extracted knowledge graph. This is the graph that we extracted from the Slovene test data. It's limited in the number of phrases, since it displays only the largest three connected components of the network. For more information about the graph, please refer to the **Semantic network generation** subsection under the **Results** section.