FRI Natural language processing course 2021
Project report

University of Ljubljana
Faculty of Computer and Information Science

Literary character exploration for literary text comprehension

Erik Kastelec, Bjorn Bračko, and Matic Isovski

Abstract

Literary text comprehension can be tackled in different ways. In this article we focused on literary character recognition and presented three methods for their importance evaluation. In addition to that we showed that Entity Co-occurrence Graphs (ECG) can be used for literary character importance evaluation as well as relation extraction and protagonist/antagonist detection. To test our proposed solutions we constructed a manually annotated corpus of 11 English and 11 Slovene literary works. Corpus as well as code are made available at: https://github.com/erikkastelec/NLP-project1

Keywords
Natural language processing, Information extraction, Co-occurrence Graph, Literary character detection

Advisors: Slavko Žitnik

Introduction

Literary text comprehension is a complex and challenging task in natural language processing. To extract the relevant information, we need to address a number of challenging language processing tasks. We present the complete process of information extraction and Knowledge base creation.

This article focuses on capturing literary character information in a form of an Entity Co-occurrence Graph (ECG), which holds the information about relations between entities. Similar approaches were introduced in the article, where they tackled explanation of Entity Co-occurrence in Twitter, using Knowledge Graphs [1].

First, named entities must be extracted and co-referenced. Next, ambiguous and deduplicated entities must be filtered. Lastly, entity relation analysis can be done, hence character information can be extracted.

An important part of character information extraction is protagonist/antagonist prediction. This can be done using social network extraction, as proposed by Matt Fernandez, Michael Peterson and Ben Ulmer in article [2]. Their approach is divided to three steps: character detection and co-reference, sentiment analysis and network analysis. After mention detection and co-reference resolution, character candidates are obtained. Then, using sentiment analysis, relations between character pairs are extracted. Lastly, a network is constructed using characters as nodes and relations as connections. With network analysis, protagonist/antagonist information is extracted.

Over the last few years a lot of work has been done to improve natural language processing (NLP) for Slovene language. Large accurately annotated corpora are needed for NLP task, such as the latest training corpus ssj500k introduced in Krek et al. (2019) [3], which was used for training models for CLASSLA pipeline, which can be used for text preprocessing as well as named entity extraction (NER). CLASSLA library was introduced in Ljubešić and Dobrovoljc (2019) [4].

Co-reference resolution is crucial for matching pronouns to named entities. Introduction of coref149 [5] and senticoref [6], which is comparable to English-based corpora, allowed advancement in more complex neural based methods, which require a lot of training data. Both were used as part of the RSDO project [7], SloCOREF [8], a Co-reference resolution tool for Slovene language. It offers a simple mention detection and mention clustering, using different models with different embeddings.

Methods

0.1 Corpus

Part of this paper was focused on construction of new corpus containing annotated Slovene and English literary works. Our new corpus consists of 22 literary works, 11 of them are in English and 11 in Slovene. For each work we analyzed basic information such as number of words, language, title, author and release year.

In addition to that we denoted important characters and rank them by importance. If applicable, we determined protagonists, antagonists and relations between characters, which were classified as positive, negative or neutral.

We tried to select literary works of differing lengths and from different time periods, as that can effect the used language. Statistics about length can be seen in Table 1. Books range from year 1597 to 1997, with the average being 1874. This is a result of primarily picking fairy tales.

Fairy tales often contain plot twists and deception. For example, in Little Red Cap, the Wolf pretends to be Little Red Cap. This provides a challenging task of determining positive and negative relations between characters.

We also did a simple corpus analysis, from which results are shown in Table 1. We can see a similarity in terms of character information and their relations. That is because we used few of the same stories in both languages. Interestingly, English works tend to be longer than Slovenian (longer stories as well as longer sentences) and have also greater vocabulary size. Lastly, we see that the type/token ratio is quite low, which indicates a low vocabulary richness, hence less difficulty for automated analysis.

Corpus can be accessed at: https://github.com/erikkastelec/NLP-project1/blob/master/books/corpus.tsv

Table 1. Corpus statistics

language	slovenian	english	both
number of works	11	11	22
shortest (num. words)	681	1068	681
longest (num. words)	73799	349736	349736
avg. num. of words	33165	78121	55644
avg. num. of words in sentence	18	20	19
avg. num. of characters	7	7	7
avg. num. of relations	5	7	6
avg. vocab. size	5827	4380	5104
avg. type/-token ratio	0.252	0.184	0.218

0.2 Literary character extraction and importance evaluation

This sections explains the process of entity (literary character) extraction and ranking of their importance. The whole pipeline can be seen in Figure 1.

0.2.1 Named Entity Recognition (NER) pipeline

To extract usefull information from text we used Classla [4] and Stanza [9] pipelines. Stanza NLP pipeline allows us to perform tokenization, part-of-speech (POS) tagging, lemmatization, dependency parsing as well as named entity recognition (NER). We used it for English language, while we used Classla, a fork of Stanza, which is adapted for processing Slovenian, Croatian, Serbian and Bulgarian language, for processing Slovene text.

0.2.2 Named entity deduplication

Extracted named entities are often ambiguous. When it comes to people, first and last names are often only used the first time the entity occurs and are in the future referred by only the first or last name. Organizations are also often referred to by only a part of the full name or only their initials [10].

To combat in-text ambiguities we rely on string matching using Levenshtein distances between strings. We adapted the deduplication method from FuzzyWuzzy library[11] to be better suited for named entity deduplication. We improved the selection of the canonical example in which all the similar entities are merged, by accounting for number of occurrences in the text, references that only use first and last name and organizations, which are referred by only their initials. As a result we get proper entity names as well as number of occurrences of entity in text.

Deduplication using Levenshtein distance matching is slow when it comes to large collections of items. We mitigate this issue by reducing search space with simstring [12] method, which is approximate dictionary matching using different similarity measures (Dice, Jaccard and cosine). We use 2-grams as a feature and cosine distance as similarity measure. Entities that have similarity higher than threshold are then used with the FuzzyWuzzy deduplication that is explained earlier. Deduplication process significantly reduces the number of nodes referring to the same entity and thus improves performance of the algorithm.

0.2.3 Co-reference resolution

Majority of named entities are captured via Classla and Stanza pipelines, but there still exists a problem of detecting co-references that refer to detected named entities. Co-reference resolution for Slovene language was made possible by intro-

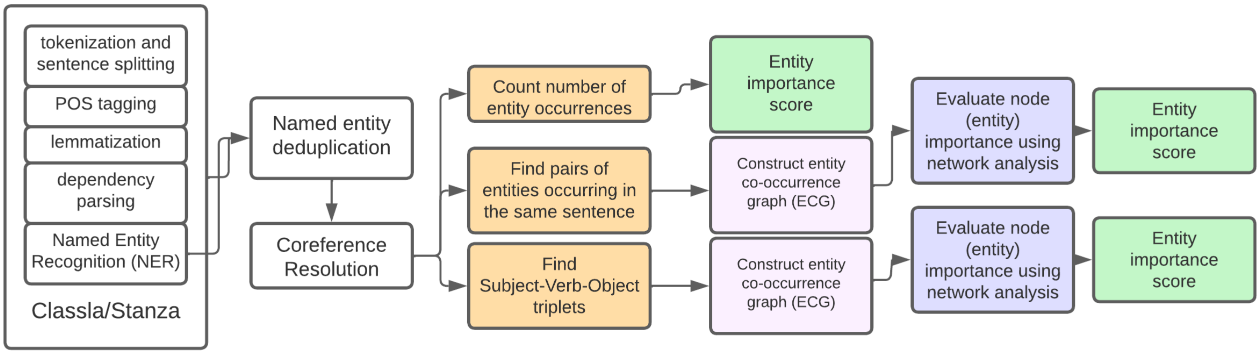
tokenization and sentence splitting
POS tagging
lemmatization
dependency parsing
Named Entity Recognition (NER)
Classla/Stanza
Named entity deduplication
Coreference Resolution
Count number of entity occurrences
Find pairs of entities occurring in the same sentence
Find Subject-Verb-Object triplets
Entity importance score
Construct entity co-occurrence graph (ECG)
Construct entity co-occurrence graph (ECG)
Evaluate node (entity) importance using network analysis
Evaluate node (entity) importance using network analysis
Entity importance score
Entity importance score

Figure 1. Visualization of pipeline used for extraction of literary characters and evaluation of their importance

duction of coref149 [5] and senticoref [6] dataset, which is comparable to English-based corpora. Large datasets allow models to learn more general patterns. Better generalization of models trained on larger datasets was shown in Klemen and Žitnik [13], where they used BERT [14] contextual embedding as an input to the neural based co-reference scorer. This model was used in our paper for co-reference resolution of Slovene text. For English with used coreferee library [15] in combination with spacy library [16].

0.2.4 Construction of entity co-occurrence graph (ECG)

After process we use dependency parsed text to generate subject-verb-object (SVO) triplets. This is done using pattern-based framework for language-neutral predicate-argument extraction patterns presented in White et al. [17]. Evaluation done by Zhang et al. [18] showed state of the art performance compared to other Open Information Extraction (Open IE) tools.

SVO triplets can be viewed as a connection between two entities. After correcting entity names using deduplication from previous section and resolving co-references we get named entity pairs, which can be used to construct a entity co-occurrence graph (ECG).

In addition to ECG from entity co-occurrence in SVO triplets we construct a second graph by analyzing named entity co-occurrences in the same sentence. If two entities co-occurre we make a connections between them in the graph or increase the weight of the connection if it already exists. Inverted weights are used when computing betweenness centrality, during entity importance evaluation.

0.2.5 Entity importance evaluation

Number of occurrences of entities in the text and two Entity Co-occurrence Knowledge Graphs (ECG) constructed from entity co-occurrence in the same SVO triplet and entity co-occurrences in the same sentence are used as an input to three different approaches of entity (literary character) importance evaluation:

1. EOIS - Entity Occurrence Importance Score
2. ECSVO - Entity Co-occurence in SVO triplets
3. ECS - Entity Co-occurrence in Sentence

EOIS uses number of occurrences of entity in the text to determine it's importance and is the simplest of the three measures.

Both ECSVO and ECS methods use network analysis to determine entity importance. The difference is that ECSVO uses ECG constructed from SVO triplets, while ECS utilizes ECG constructed from entity co-occurrence in the same sentence. Both

Node importance in ECG produced by ECSVO and ECS was evaluated using betweenness centrality, which is a widely used measure that captures a node's role in allowing information to pass from one part of the network to the other [19]. In addition to betweenness centrality we tested degree centrality measure.

0.3 Social network extraction for relationship and Protagonist/Antagonist prediction

This sections explains the process of character relationship extraction and protagonist/antagonist prediction.

0.3.1 Relationship Extraction

For each pair of characters we obtain a list of sentences in which they appear together and thus, by our assumption, interact. Next we extract the verb phrase connecting the two entities. We then perform sentiment analysis on the verb phrase. We do this by using the SentiWordNet [20] lexicon for the English language and the Slovene sentiment lexicon KSS 1.1 [21] for the Slovenian language. For each word, we take its lemma and extract the score from the lexicon. We average the score of all sentences for each pair of characters. We obtain a list of character pairs and the sentiment of the relationship for that pair.

From this we can construct a signed graph representing the social network for the document.

0.3.2 Protagonist/Antagonist prediction

For the protagonist and antagonist prediction we split our features into two groups.

The first batch of features consists of:

- degree of the character
- number of appearances in the story overall
- value for the sum of all edge weights for positive edges
- value for the sum of all edge weights for negative edges

While the second batch contains features obtained from triad analysis on the relationship graph we obtained before. As is shown in Leskovec et. al. [22] and Fernandez et. al. [2], using triad analysis we can gain information about the characters in regards to other characters. For example characters at the negative end of a T_1 triad is much more likely to be the antagonist, while the protagonist will often be in T_3 triads with other allies or positive characters [2]. And a T_2 triad might be helpful in identifying a love triangle within the story.

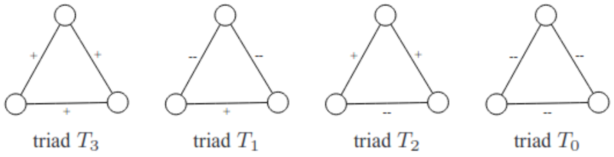

Figure 2. Undirected signed triads [22].

The features we obtain from triad analysis are:

- number of occurrences in T_1 triads
- number of occurrences in T_3 triads

To make the final prediction we use separate classifiers for protagonist prediction and antagonist prediction.

Results

We evaluated three methods for character recognition and importance ranking presented in 0.2 (EOIS, ECSVO and ECS), as well as relationship extraction and protagonist/antagonist prediction introduced in 0.3.

There can be small differences, between detected character name and annotated name, such as different capitalization or different spelling, which often happens in plays. Because of these anomalies we decided to try to match such characters to ground truth spelling.

0.4 Literary character recognition

First we take a look at performance of our pipeline 0.2 for literary character recognition. In books with longer text there are often too many character to annotate all of them, so we decided to calculate metrics based on first n predictions, where n is number of manually annotated entities. This makes it possible to compare stories of different lengths. For evaluation we used recall, precision and F score. There can be small differences, between detected character name and annotated name, such as different capitalization or different spelling, which often happens in plays. Because of these anomalies we decided to try to match such characters to ground truth spelling.

We compared all three methods on books from our corpora, which is introduced in section 1. Evaluation of methods on the whole corpus, only English and only Slovene books can be seen in Tables 2, 4, 3. For all three tables results for ECS and ECSVO methods were achieved using betweenness centrality. Table 5 contains results for whole corpora evaluation using degree centrality.

	EOIS	**ECS**	**ECSVO**
precision	**0.7103**	0.6036	0.5624
recall	**0.5867**	0.4628	0.3906
F1 score	**0.6257**	0.4962	0.4410
mAP	**0.7713**	0.6070	0.5081

Table 2. Results for evaluation of all books from our corpus

	EOIS	**ECS**	**ECSVO**
precision	**0.6253**	0.4940	0.3653
recall	**0.5874**	0.4940	0.3565
F1 score	**0.5978**	0.4940	0.3602
mAP	**0.7701**	0.6167	0.5081

Table 3. Results for evaluation of all Slovene books from our corpus using degree betweenness centrality measure

	EOIS	**ECS**	**ECSVO**
precision	0.8271	0.7542	**0.8333**
recall	**0.5854**	0.4198	0.4375
F1 score	**0.6642**	0.4992	0.5521
mAP	**0.7730**	0.5937	0.5081

Table 4. Results for evaluation of English books from corpus using degree betweenness centrality measure

0.5 Literary character ranking

Next we look at ability of our methods to predict importance of literary character. Our corpus contains characters, which are ranked by their importance. Predicted ranking and ground truth are used to compute mean average precision (mAP), which is often used for evaluation of recommendation systems. Evaluation was done on combined corpus of books in English and Slovene as well as for only Slovene and only

	EOIS	ECS	ECSVO
precision	**0.7103**	0.5864	0.5814
recall	**0.5867**	0.4456	0.4096
F1 score	**0.6257**	0.4791	0.4600
mAP	**0.7713**	0.6070	0.5320

Table 5. Results for evaluation of all books from corpus using degree centrality measure

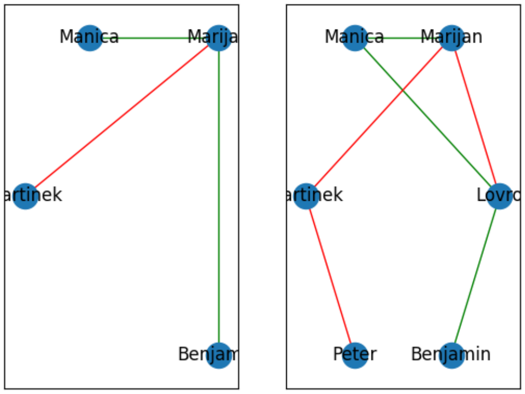

Figure 3. Example of relation detection for Slovene literary work Deseti brat. Prediction on the left, ground truth on the right. For visibility neutral relations are not shown

English. Results can be seen on Tables 2, 4, 3.

0.6 Relationship extraction

Next we evaluated the relationship extraction using precision, recall and F1 value. We also include the accuracy where we ignore False Negatives (correctly predicted relationships / all predictions) as the performance of the extraction is heavily influenced by the performance of the method used to obtain relationship pairs.

The results for relationship extraction using entity pairs provided by ECSVO and ECS method 0.2 evaluated on the whole corpus are presented in Table 6.

	ECS	ECSVO
accuracy of predictions	**0.56**	0.375
precision	0.2258	**0.25**
recall	**0.2154**	0.0536
f1	**0.2205**	0.0882

Table 6. Evaluation of relationship extraction

We also include an evaluation in which we provide the true character list to the extractor, so it does not depend on the performance of our character recognition methods. These results are shown in Table 7.

	ECS	ECSVO
accuracy of predictions	**0.56**	0.375
precision	0.2143	**0.25**
recall	**0.2308**	0.0536
f1	**0.2222**	0.0882

Table 7. Evaluation of relationship extraction using preset characters

0.7 Protagonist/antagonist detection

Lastly we evaluate the protagonist and antagonist detection using precision, recall and f1 score. We created a train/test split among the books in the corpus and trained a random forest classifier. The results are shown in Table 8.

	ECS	ECSVO
precision	**0.7857**	0.4615
recall	**0.3548**	0.2143
f1	**0.4889**	0.2927

Table 8. Evaluation of protagonist/antagonist prediction

Discussion

0.8 Literary character recognition

Results of literary character recognition performance evaluation on our whole corpus, which can be seen on Table 2, shows that EOIS method performed the best, followed by ECS and ECSVO. This holds true also for evaluations on only English and only Slovene, which can be seen on Table 3 and 4. both English. Result is strongly influenced by shorter stories, which contain between 1000 and 3000 words, where we have a hard time detecting enough sentences or Subject-Verb-Object pairs. This can be seen in Slovene tale Trnuljčica/Little Brier-Rose, where EOIS produces F score of 0.36 and mAP of 0.55, while both ECS and ECSVO produce F score of 0 and mAP of 0. Same tale is also a great example of how character detection is challenging when characters can not be recognized as named entities. It contains ambiguous entities such as prince, king, queen, wise women, frog, old woman, thirteenth fairy. In some stories such entities can represent actual characters, while in others they do not.
In addition to betweenness centrality measure we tried degree centrality measure, which improved ECSVO performance, while hindering performance of ECS.

0.9 Literary character ranking

Result of ranking match previously discussed character recognition results. EOIS method performs the best, followed by ECS and ECSVO. It is worth noting that the average mean precision over all the books can be effected by longer books, where character importance is hard to determine even by human annotator. Such works can lower the overall performance. Ranking is highly dependant on character recognition and suffers when important characters are not detected.

0.10 Relationship extraction

Relationship extraction results, as presented in Table 6 and Table 7, show that using the ECS method for pair extraction performed the best. We see that while the relationships we predict tend to be correct, we miss a large amount of relationships.

There are several key factors which all contribute to the lower performance. First, we do not try to identify cross-sentence relationships and instead rely on the assumption that the relationships are confined to a sentence, which is not always true. Second, performing sentiment analysis on the whole sentence causes us to include noise in the score calculation, while extracting only verb phrases can cause us to miss critical information. Ideally we would need to select which of the words in the sentence are relevant and process them as a whole. Fairy tales also tend to have mostly negative plots throughout much of the story until the end where things become ”happy”. Therefore we over-classify a lot of relations as negative. Deception is also a key element in many Fairy tales, where characters are pretending to be another, skewing the relationship between one character and a character who is pretending.

Finally, since we are starting out with an imperfect pair relations set, the method is hindered in the start as many relations are missed due to various problems in entity detection. We attempt to remedy this by providing the true characters and their synonyms to the entity pair retrieval methods 0.2. However, we see that there is practically no improvement. This could indicate that the problems lie in the co-reference, verb extraction or sentiment layers.

0.11 Protagonist/Antagonist prediction

Results of the protagonist and antagonist prediction evaluation, which are presented in Table 8, show acceptable results. We manage to correctly identify some protagonists and antagonists.

There are again key factors which are lowering performance. First, we ignore all temporal aspects of the relationships, and thus cannot correctly capture relationships that turn sour or a villain revealing themselves at the end of the story. Another issue is that antagonist don’t always form many relationships/enemies, so their character becomes harder to identify as villain. Antagonist also don’t necessarily have to be villains, they can just be the protagonists opposite of ideals, values or way of thinking. These kind of antagonists are even harder to detect.

And again, we are working with features obtained from Relationship extraction, which is already of lower performance. This weakens our method significantly as we have many missing relationships and many incorrectly predicted relationships.

0.12 Conclusion

In this article we tackled the problem of literary work comprehension. We focused on literary character detection as well as their performance evaluation. We proposed three methods and showed that entity occurrence in the text can be a simple and effective way of approaching evaluation of character importance. We also showed that constructing Entity Co-occurrence Graphs(ECG) and performing network analysis on them can be a suitable way of determining character relations as well as finding protagonists and antagonists.

References

[1] Yiwei Wang, Mark James Carman, and Yuan-Fang Li. Using knowledge graphs to explain entity co-occurrence in twitter. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM ’17, page 2351–2354, New York, NY, USA, 2017. Association for Computing Machinery.

[2] Matt Fernandez, Michael Peterson, and Ben Ulmer. Extracting social network from literature to predict antagonist and protagonist. *Recuperado de: https://nlp. stanford. edu/courses/cs224n/2015/reports/14. pdf*, 2015.

[3] Simon Krek, Kaja Dobrovoljc, Tomaž Erjavec, Sara Može, Nina Ledinek, Nanika Holz, Katja Zupan, Polona Gantar, Taja Kuzman, Jaka Čibej, Špela Arhar Holdt, Teja Kavčič, Iza Škrjanec, Dafne Marko, Lucija Jezeršek, and Anja Zajc. Training corpus ssj500k 2.2, 2019. Slovenian language resource repository CLARIN.SI.

[4] Nikola Ljubešić and Kaja Dobrovoljc. What does neural bring? analysing improvements in morphosyntactic annotation and lemmatisation of Slovenian, Croatian and Serbian. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 29–34, Florence, Italy, August 2019. Association for Computational Linguistics.

[5] Slavko Žitnik. Slovene coreference resolution corpus coref149, 2018. Slovenian language resource repository CLARIN.SI.

[6] Slavko Žitnik. Slovene corpus for aspect-based sentiment analysis - SentiCoref 1.0, 2019. Slovenian language resource repository CLARIN.SI.

[7] Razvoj slovenščine v digitalnem okolju. Accessible: https://slovenscina.eu. [Accessed: 2022-05-13].

[8] Slavko Žitnik. Slocoref - coreference resolution for slovene language. Accessible: https://github.com/RSDO-DS3/SloCOREF. [Accessed: 2022-05-09].

[9] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 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*, 2020.

[10] Mónica Marrero, Julián Urbano, Sonia Sánchez-Cuadrado, Jorge Morato, and Juan Miguel Gómez-Berbís. Named entity recognition: Fallacies, challenges and opportunities. *Computer Standards Interfaces*, 35(5):482–489, 2013.

[11] Adam Cohen. Fuzzywuzzy library. Accessible: https://github.com/seatgeek/fuzzywuzzy. [Accessed: 2022-03-28].

[12] Naoaki Okazaki and Jun'ichi Tsujii. Simple and efficient algorithm for approximate dictionary matching. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 851–859, Beijing, China, August 2010.

[13] Matej Klemen and Slavko Zitnik. Neural coreference resolution for slovene language. *Computer Science and Information Systems*, 2021.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[15] Richard Hudson. Coreferee library. Accessible: https://github.com/msg-systems/coreferee. [Accessed: 2022-05-2].

[16] Industrial-strength natural language processing. Accessible: https://spacy.io. [Accessed: 2022-05-01].

[17] Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. Universal Decompositional Semantics on Universal Dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, Austin, Texas, November 2016. Association for Computational Linguistics.

[18] Sheng Zhang, Rachel Rudinger, and Ben Van Durme. An Evaluation of PredPatt and Open IE via Stage 1 Semantic Role Labeling. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS)*, Montpellier, France, September 2017.

[19] Jennifer Golbeck. Chapter 21 - analyzing networks. In Jennifer Golbeck, editor, *Introduction to Social Media Investigation*, pages 221–235. Syngress, Boston, 2015.

[20] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, 2010.

[21] Klemen Kadunc and Marko Robnik-Šikonja. Slovene sentiment lexicon KSS 1.1, 2017. Slovenian language resource repository CLARIN.SI.

[22] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1361–1370, 2010.