FRI Natural language processing course 2021
Project report

University of Ljubljana
Faculty of Computer and Information Science

Literacy situation models knowledge base creation

Erik Kastelec, Bjorn Bračko, and Matic Isovski

Abstract

The abstract goes here. The abstract goes here.

Keywords

Keyword1, Keyword2, Keyword3 ...

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.

This article will focus on capturing information in a form of a Knowledge Graph (KG). In the recent years event-centric KGs started to replace the conventional entity-centric KGs. Event-centric KGs allow us to more easily capture causal and temporal relations as well as relations between entities.

We can see that event extraction is one of the main sub tasks for our knowledge base creation. In the recent years methods for sentence-level event extraction are being replaced with document-level event extractors.

To understand how extracted events are connected, event causal relation extraction (ECE) methods are used. Causal relations define what events caused another event. For event causal relation extraction there are two available datasets, Causal-TimeBank [1] and EventStoryLine[2].

Transformer-based machine learning technique BERT was introduced in Devlin et al. [3]. In Hosseini et al. [4] they show that BERT can be used to predict directionality in causal pairs. They also present a CREST framework, which unifies causal relation datasets.

In Gao et al. [5] they present a document level main event detection based on event participating in multiple causal links.

A rule-based, domain-sensitive multilingual temporal tagger HeidelTime was introduces in [6].

In Wei et al. [7] they propose a translation-based approach for cross-lingual data augmentation, which can be used on labeled data from a different language to extend the available training data.

In recent 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) [8], 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) [9].

Co reference resolution is crucial for matching pronouns to named entities. Introduction of coref149 [10] and senticoref [11], which is comparable to English-based corpora allowed advancement in more complex neural based methods, which require a lot of training data.

Methods

0.1 Corpora

For evaluation of task of literary text comprehension we used a collection of 1002 books gathered as a part of Project Gutenberg [12]. From this collection we selected a sub-sample of English short stories, which were used for subjective evaluation of our proposed solution. For Slovene literary text we selected a few of the most known short stories, where results were also subjectively evaluated.

Datasets used for sub tasks of coreference resolution, event extraction and causal relation extraction are provided in the following subsections.

0.2 Entity co-occurrence knowledge graph

To extract usefull information from text we used Classla [9] and Stanza [13] 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.

Majority of named entities are captured via Classla and Stanza pipelines, but there still exists a problem of detecting coreferences that refer to detected named entities. Coreference resolution for Slovene language was made possible by introduction of coref149 [10] and senticoref [11] 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 [14], where they used BERT [3] contextual embedding as an input to the neural based coreference scorer. This model was used in our paper for coreference resolution of Slovene text.

Extracted named entities and their coreferences are often ambigous. 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 [15].

To combat in-text ambiguities we rely on string matching using Levenshtein distances between strings. We adapted the deduplication method from FuzzyWuzzy library[16] 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 [17] 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.

After deduplication process we use dependency parsed text as 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. [18]. Evaluation done by Zhang et al. [19] showed state of the art performance compared to other Open Information Extraction (Open IE) tools.

SVO triplets can be viewed as an event connection 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 knowledge graph (ECKG). ECKG is than used to perform entity relation analysis. From co-occurrences we can gather interactions between different entities, which can be seen as character, when it comes to literary text.

Keeping entities in a graph allows us to use network analysis. Protagonist detection is done by evaluating entity node degrees and number of appearances in comparison to other characters).

0.3 Relationship extraction

Our next task was to extract semantic relationships from novels in order to obtain information that would allow as to grow some knowledge. Extracted relationships usually occur between two or more entities of a certain type and fall into a number of semantic categories. But to gain knowledge that makes sense, the removal of repeated relations (disambiguation) and generally refers to the extraction of many relationships is required.

On extracted entities, we performed relationship extraction using two different approaches, rule-based and model-based. Usually, good precision values are achieved with Rule-based approaches. Using partial syntactic parsing can simplify the linguistic structures containing instances of semantic relations.

Relation statements can be represented as:

$$r^i = (x^i, s_1^i, s_2^i), \tag{1}$$

x is the tokenized sentence, s_1 and s_2 are the spans of the two entities within that sentence. Two statements can consist of two different sentences, but they can both contain the same entity pair. If both contain the same entity pair, they should have the same s_1-s_2 relation.

0.3.1 Rule-based

In this approach, different patterns were used to extract relationships between entities [20]. We used spaCy framework [21], an open-source software library for advanced natural language processing, that features NER, POS tagging, dependency parsing, word vectors and more. After that, we constructed a knowledge graph, where nodes represent entities and links represent relationships.

Defining the right set of patterns is the most difficult part. Because the approach is manual, it often results in large sets of noisy patterns. There must be some sort of semantic categorization of relationships between the entities. Taking the verb, which has the time component, may not be a good way to have build a knowledge graph. There could be synonyms of verbs that have different relationship types with similar, identical semantic values.

0.3.2 Model-based

To avoid noisy patterns, NLP model can be used for event extraction. In this step, we used an OpenNRE [22], open-source unified framework for relation extraction models. This framework provides different pre-trained models, dataset on which the models were trained and tools for performing additional learning of the models on custom data. GPU parallel computing is also supported. We tried CNN and BERT encoder models.

Convolutional neural network models are commonly used in information extraction tasks. But CNN just considers the correlation between consecutive words and ignores the correlation between discontinuous words. CNNs help us with parallelization, local dependencies and distance between positions. OpenNRE CNN model is pre-trained on WIKI80 dataset.

Bidirectional Encoder Representations from Transformers is a transformer-based machine learning technique. Transformers were introduced to deal with the long-range dependency challenge, which cannot be extracted using CNNs. They provide a self-attention mechanism, that allows models to look at other words in the input sequence to get a better understanding of a certain word in the sequence.

0.4 Event causality identification

For event causality identification we implemented the method introduced by Jian Liu et. al. [23]. The propose a mixture model using a knowledge aware reasoner and a mention masking reasoner, which can leverage both external knowledge to enrich the representation of events as well as learn event-agnostic, context specific patterns which grants the model a decent ability to generalize on previously unseen data.

0.4.1 Knowledge aware reasoner

Given a pair of events e_1 and e_2 the knowledge aware reasoner first retrieves the related knowledge in ConceptNet [24] and then encodes the knowledge into contexts for reasoning. We only consider 18 semantic relations that are potentially useful for event causality identificaiton: CapableOf, IsA, HasProperty, Causes, MannerOf, Causes- Desire, UsedFor, HasSubevent, HasPrerequisite, NotDesires, PartOf, HasA, Entails, ReceivesAction, UsedFor, CreatedBy, MadeOf, and Desires. We also limit the total knowledge retrieved.

0.4.2 Mention masking reasoner

With the mention masking reasoner we aim to explore event agnostic, context-specific patterns for reasoning. We replace e_1 and e_2 with a '[MASK]' token to exclude event information. Then a BERT encoder is used to obtain embedded representations of events $F_{MASK}^{(e_1,e_2)}$.

By using the mention masking reasoner, we force our model to predict whether e_1 and e_2 form a casual relation based on context specific clues as the masked representation does not contain any event-specific learning.

$$L = -\delta_{A,B} * log(p(l=1|A,B)) + (1-\delta_{A,B}) * log(1-p(l=1|A,B)), \tag{2}$$

where $\delta_{A,B}$ is the Kronecker delta which takes the value 1 when both A and B express a causal relation and 0 otherwise. $p(l=1|(A,B)) = \frac{1}{1+exp(F_{MASK}^{A}{}^{T} F_{MASK}^{B})}$ defines the distributional similarity score.

0.4.3 The attentive sentinel

With the attentive sentinel we aim to learn a trade-off between the knowledge aware reasoner and the mention masking reasoner, by learning an attentive gate as their combination of weights:

$$g_{e_1,e_2} = \sigma(W(F_{KG}^{(e_1,e_2)} \oplus F_{MASK}^{(e_1,e_2)}) + b), \tag{3}$$

where W and b are model parameters, $\oplus$ is the concatenation operator. We then adopt a weighted summation to integrate $F_{KG}^{(e_1,e_2)}$ and $F_{MASK}^{(e_1,e_2)}$ as the final feature:

$$F_{e_1,e_2} = g_{e_1,e_2} * F_{KG}^{(e_1,e_2)} + (1-g_{e_1,e_2}) * F_{MASK}^{(e_1,e_2)}. \tag{4}$$

The attentive sentinel balances the knowledge aware reasoner and the mention masking reasoner to make the final prediction.

0.4.4 Model prediction and training

To make the final prediction, we perform a binary classification by taking F_{e_1,e_2} as input:

$$o_{e_1,e_2} = \sigma(W_o F_{e_1,e_2} + b_o), \tag{5}$$

where o_{e_1,e_2} is the probability of there being a causal relationship between the two events; w_o and b_o are model parameters. For training we use cross-entropy as the loss function:

$$J(\Theta) = -\sum_{s} \sum_{e_i,e_j \in E_s, e_i \neq e_j} y_{e_i,e_j} log(o_{e_1,e_2}) + (1-y_{e_i,e_j}) log(1-o_{e_1,e_2}), \tag{6}$$

where Θ denotes the parameter set of our model; s ranges over each sentence; e_i and e_j range over each event in s. We used the Adam [25] algorithm to optimize model parameters.

0.4.5 Datasets used for training and evaluation

For training and evaluating the model we used the Causal-TimeBank dataset [26], which contains annotated causal relations of events within a single sentence.

Because of the lack of annotated datasets on causal event relations extracted from literary works, we used the model trained on the Causal-TimeBank dataset. As shown in Liu et. al. [23] this method generalizes very well and is suitable for transfer learning and cross-task adaptation.

We then manually evaluated the preformance of the model on the Litbank event dataset [27], which contains 100 annotated works of English-language fiction.

Results

Use the results section to present the final results of your work. Present the results in a objective and scientific fashion. Use visualisations to convey your results in a clear and efficient manner. When comparing results between various techniques use appropriate statistical methodology.

Discussion

Use the Discussion section to objectively evaluate your work, do not just put praise on everything you did, be critical and exposes flaws and weaknesses of your solution. You can also explain what you would do differently if you would be able to start again and what upgrades could be done on the project in the future.

Acknowledgments

Here you can thank other persons (advisors, colleagues ...) that contributed to the successful completion of your project.

References

[1] Paramita Mirza and Sara Tonelli. An analysis of causality between events and its relation to temporal information. In *COLING*, 2014.

[2] Tommaso Caselli and P. Vossen. The event storyline corpus: A new benchmark for causal and temporal relation extraction. In *NEWS@ACL*, 2017.

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

[4] Pedram Hosseini, David A. Broniatowski, and Mona T. Diab. Predicting directionality in causal relations in text. *ArXiv*, abs/2103.13606, 2021.

[5] Lei Gao, Prafulla Kumar Choubey, and Ruihong Huang. Modeling document-level causal structures for event causal relation identification. In *NAACL*, 2019.

[6] Jannik Strotgen and Michael Gertz. A baseline temporal tagger for all languages. In *EMNLP*, 2015.

[7] Sam Wei, Igor Korostil, Joel Nothman, and Ben Hachey. English event detection with translated language features. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 293–298, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[8] 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.

[9] 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.

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

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

[12] Project gutenberg. Accessible: www.gutenberg.org. [Accessed: 28. 3. 2022].

[13] 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.

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

[15] 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.

[16] Adam Cohen. Fuzzywuzzy library. Accessible: https://github.com/seatgeek/fuzzywuzzy. [Accessed: 28. 3. 2022].

[17] 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.

[18] 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.

[19] 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.

[20] Kang Liu, Yubo Chen, Jian Liu, Xinyu Zuo, and Jun Zhao. Extracting events and their relations from texts: A survey on recent research progress and challenges. *AI Open*, 1:22–39, 2020.

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

[22] Xu Han, Tianyu Gao, Yuan Yao, Deming Ye, Zhiyuan Liu, and Maosong Sun. OpenNRE: An open and extensible toolkit for neural relation extraction. In *Proceedings of EMNLP-IJCNLP: System Demonstrations*, pages 169–174, 2019.

[23] Jian Liu, Yubo Chen, and Jun Zhao. Knowledge enhanced event causality identification with mention masking generalizations. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3608–3614, 2021.

[24] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] Paramita Mirza, Rachele Sprugnoli, Sara Tonelli, and Manuela Speranza. Annotating causality in the tempeval-3 corpus. In *Proceedings of the EACL 2014 Workshop on Computational Approaches to Causality in Language (CAtoCL)*, pages 10–19, 2014.

[27] Matthew Sims, Jong Ho Park, and David Bamman. Literary event detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3623–3634, 2019.