Knowledge Graph Construction from the Text

Yuxin He
yhe242@wisc.edu

Ching-Wen Wang
cwang553@wisc.edu

Yiwu Zhong
yzhong52@wisc.edu

Project Link
https://github.com/yuxinh3/Relation-Extraction-and-Knowledge-Graph

1. Abstract

Knowledge graph is an abstract data representation that can be constructed from the unstructured text. Such structured representation helps people and machines to understand the information more efficiently. In this project, our goal is to construct reasonable knowledge graphs from the text appearing our real life. To achieve our goal, we collected and annotated the text data manually from different data sources. Based on our collected text data, we evaluated several method candidates, selected the best method to extract the triplets from sentences, and constructed the knowledge graphs. Our results showed that certain methods performed well on the particular data sources but not on the other data sources. We finally provided in-depth result analysis and the visualization of constructed knowledge graphs from the chosen method.

2. Introduction

Tremendous information are created and transferred among people everyday. To digest the information from the news, the newspapers and the posts on social media, human naturally grasp the important information, convert them into a few keywords, and finally memorize them. These key information in mind are usually stored in "mind palace". Human can effortlessly query their "mind palace" anytime and find the desired information in seconds.

However, for machines, it's not an easy task to process such high volume data in a short time. Knowledge graph, hopefully, an abstract and structured data format, is widely used in industry, such as web search (e.g., Google search), question answering (e.g., voice assistants), recommendation systems (e.g., movie recommendation) and data integration (e.g., Wikipedia). Specially, knowledge graph consists of a set of nodes that are usually represent the entities (e.g., the nouns) and a set of directed edges among the nodes (e.g., the relationships between nodes). Take Fig. 1 as example. This knowledge graph includes the node of Bob and its relations to the other nodes, such as the date of birth and his interests in The Mona Lisa. Interestingly, Bob is closely related to the Leonardo Da Vinci via the intermediate node The Mona Lisa. Essentially, knowledge graph can be decomposed into a list of triplets with each as ⟨subject, predicate, object⟩ (e.g., ⟨BOB, is interested in, The Mona Lisa⟩). These triplets can be parsed from the large-scale unstructured text, such as the text in Wikipedia.

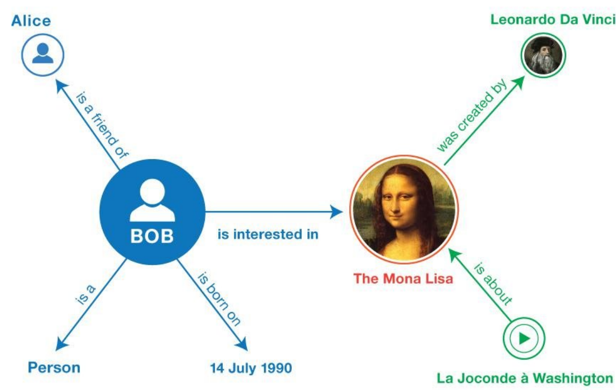

Figure 1. An example [13] of a simple knowledge graph.

With such structured information, knowledge graph allows machines and human to quickly digest the key information, ignore the redundant words in text and access the target node or edge in an efficient way. If we have a way to create such knowledge graphs from the text we usually meet everyday, it can be beneficial for our work and study efficiency. Now the question is: can we construct a knowledge graph from the text in our daily life?

The goal of this project is to extract a knowledge graph from the given sentences, especially from the text in our daily life. Constructing a knowledge graph from given text involves 2 steps: extracting the triplets from each sentence and merging all triplets into the graph format. In this project, we pay more attention to the first step, triplet extraction from text. Since previous models were trained and evaluated in different datasets and settings, it's not clear which model is the best in practice to fulfill our goal. To this end, we plan to collect our own sentence dataset and label the triplets in the sentences. The dataset are only divided into validation set and test set. We directly adopt 3 widely-used existing models (Stanford OpenIE [3] and SPICE [1]) or tool (spaCy library[1]) for triplet extraction, and evaluate the extracted triplets on the validation set. According to the evaluation results, we select and apply the best model on the test set and obtain a list of triplets. Finally, these extracted triplets form a knowledge graph using word lemmatization.

The roadmap of our project is shown in Fig. 2. For data collection and labeling, we collect the sentences from different sources in our daily life, such as the newspaper, the novel, and the image sentence descriptions. For each source, we collect a fixed number of sentences (e.g. 100 sentences) and manually label the subject, the predicate and the object in each sentence. These labels are used as ground-truth during evaluation. The more the extracted triplets can be matched to these human annotated labels, the better the model is.

To construct a knowledge graph, relation extraction [7, 17, 16, 3, 20, 1] is an essential step. The existing models or tools we used in this project include Stanford OpenIE [3], SPICE [1] and spaCy library. Stanford OpenIE [3] segments the sentences into smaller proportions then extracts the triplets by recursively traversing the dependency trees that are parsed from sentence fragments. SPICE [1] firstly uses a dependency parser which was pre-trained on a large-scale dataset to capture the syntactic dependencies between sentence words, and then extract triplets from dependency trees using a rule-based system [20]. The nuatural language processing (NLP) tool spaCy library includes pre-trained models and helps users build algorithms that process and understand text. spaCy library is first used to perform the part-of-speech tagging and dependency parsing of each word in the sentences so that user could get the overall idea on structure, then a rule-based strategy is applied to select the subject, the relation, and the object.

In summary, this project aims at evaluating different existing methods for relation extraction, and selecting the best model to construct the knowledge graph. There are multiple potential ways to evaluate the performance of existing models. In our project, we care more about which widely-used model can be the best practice in real life, such as the collected sentence data that appear in our daily life.

3. Related Work

Knowledge graph construction includes 2 steps: first, relation extraction which extracts the triplets (subject-relation-object) from text; second, entity linking (canonicalization) which merges the parsed entities and construct the knowledge graph. Our project mainly investigates the first step, namely relation extraction. We briefly introduce the related work of relation extraction in this section.

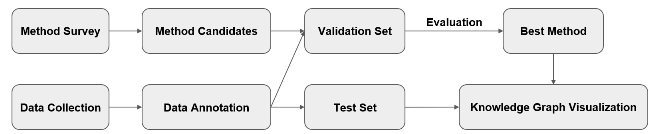

Figure 2. The roadmap of our project pipeline. After method survey, data collection, and data annotation, we evaluate the quality of the triplets that are extracted by different widely-used models on the validation set. Based on the evaluation results, we select the best method to apply on the sentences in test set. Finally, a knowledge graph will be constructed by the extracted triplets from the best model.

The existing methods for relation extraction can be divided into 2 types: rule-based algorithm and supervised learning-based methods. Specially, the deep learning methods [11][9] are prevalent in recent years and mostly belong to supervised learning methods. We used three existing methods (spaCy, Stanford OpenIE [3] and SPICE [1]) to extract the relation triplets and evaluated them on different datasets. These methods are mostly based on rule-based algorithm.

Rule-based Methods Rule-based methods are lexical pattern-based methods. Some basic steps includes entity recognition and relation extraction. Text are preprocessed into tagged part-of-speech (POS) and parsed dependency so that each words in the sentences are recognized by position and entity chunk. After that, the algorithms can present an overview on grammatical relationship [8][14]. Finally, rule-based algorithms are performed to extract relations according to different patterns:

- Simple co-occurrence method. This approach relies on the phenomena that biological entities that are mentioned together are highly related and performs a better recall [10][19]
- Relation classification method. The approach predefined a set of rules over the preprocessed texts [25]. For example, the algorithms may classify the relations into active-voice, passive-voice, and the interaction between two entities. After such construction of triplets, the algorithms could add one more step for filtering the relation through negation check, effector-effectee detection, enumeration resolution, and restricting candidate relations to focus domain for a better precision [8].

Supervised Learning-based Methods Compared to rule-based algorithms, supervised learning-based methods

[1] https://spacy.io/

rely less on human. However, the triplets which the model can extract are constrained by the training data, which means that they cannot recognize the sentence words that do not exist in the dataset. Supervised learning-based methods mainly use the following two common approaches:

- Feature-based method. This approach mostly designs grammatical features for entity pairs. For example, some model measure entity mentions and entity pairs in the context and perform pre-defined target relation on the new sentence for prediction [4][12].

- Kernel functions. This approach designs kernel functions for Support Vector Machine to detect entities and classify relations, also measures the similarities between relations and context [6][24].

Deep Learning Methods Specially, deep learning methods are supervised learning methods but take advantages of neural networks to automatically extract grammatical features from text and highly rely on large-scale datasets and intensive annotation. Through customized construction of neural network architecture, the deep learning models are trained for accurately capture the semantics within the context [23, 2]. To capture the relational information [9], there are multiple partical network architectures, including Recursive Neural Networks, Convolutional Neural Networks, Recurrent Neural Networks, Graph Neural Networks, and Attention-based Neural Networks. Instead of using the raw words or hand-crafted features of words as input, deep learning models use specific techniques, such as word embedding [21] to vectorize the sentence words. As the results, the deep learning methods obtain the state-of-the-art performance in relation extraction.

Rule-based Methods vs. Deep Learning Methods Compared to traditional machine learning methods (e.g., support vector machine), deep learning methods have showed remarkable performance and are more tolerant to the noise in the datasets. They heavily rely on better feature extraction, much more model parameters to be learned, much more training data and more intensive annotation (e.g., annotating each triplet in the sentences). However, these deep learning methods usually have domain constraints. They are limited by the concepts or categories which were pre-defined in the training data (e.g., the fixed relation types), and thus can't generalize to the free-form content, such as arbitrary text in our daily life. What's more, the deep learning methods are generally hard to interpret since the decision making is performed in the implicit semantic space.

On the contrary, despite of relatively lower performance, the rule-based methods focus on designing the universal rules which are easily to be understood by human. To obtain such rules, it generally require more sophisticated design of the algorithm, instead of the large datasets as deep learning methods. Most importantly, they can be directly applied to the text in open world, such as those in our daily life. To this end, we decided to investigate the rule-based methods in this project.

ABC News	
Collected Sentences	Annotated Triplets
Public health officials need to clearly explain the relative risks to avoid public panic.	officials,need to clearly explain,risks
The CDC recommends that adults partake in aerobic activity for at least 150 minutes at a moderate intensity.	CDC,recommends,that; adults,partake,activity

Table 1. The samples of collected sentences (left column) and annotated triplets (right column) from ABC News.

Novel	
Collected Sentences	Annotated Triplets
But Mr. Brunner, our Latin teacher, was leading this trip, so I had hopes.	Mr.Brunner,is,Latin teacher; Mr.Brunner,was leading,trip; I,had,hopes
Mr. Brunner told us about the carvings on the sides.	Mr.Brunner,told,us; Mr.Brunner,told us about,carvings

Table 2. The samples of collected sentences (left column) and annotated triplets (right column) from the novel (Percy Jackson & the Olympians: The Lightning Thief).

4. Dataset

We collected the sentence data from 3 sources: news (ABC News),the novel (Percy Jackson & the Olympians: The Lightning Thief), and human annotated captions of images [5]. For each data source, we have 100 sentences. After data collection, we manually annotate the triplets in each sentence. More details about our collected sentences and annotated triplets are presented in the Table 1, Table 2, Table 3.

During labeling the sentence triplets, we found that extracting the triplets from free-form text is actually a challenging task even for human. We identified several common challenges and summarized them as follows:

- The processing of common phrase/compound words. For example, "She is good at drawing" will be easily processed by tools as "she, is, good" if we do not perform preprocessing on sentences. Some complicated common phrase/compound words could also be con-

Image Caption	
Collected Sentences	Annotated Triplets
a man in tan shirt sitting at a table with food.	man,in,shirt; man,sitting at,table; table,with,food
a man who is sitting at a table with a plate of food in front of him.	man,sitting at,table; food,in front of,him; tabel,with,food

Table 3. The samples of collected sentences (left column) and annotated triplets (right column) from the image captions.

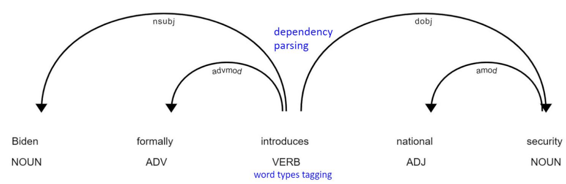

Figure 3. Visualized structure of the sentence ”Biden formally introduces national security” parsed by spaCy.

fused for human, which make both human and proposed model hard to comprehend the semantics of sentences.

- Sentence that contains conjunctions is also a challenge. The logic of human and program may be different in processing the conjunction connected words. For example, ”I like spring and winter” could be analyzed as ”I, like, spring and winter”, or ”I, like, spring; I, like, winter”. Such different labeling will result as a lower predicted precision.

- Multiple triplet extraction. Sometimes, people may ignore some triplets inside the sentence easily, which may be detected by tools. Thus, our members-oriented, whom are not grammar experts, labeled triplet may limit the performance of the proposed model.

5. Method description

After investigating the methods for relation extraction, we selected the three most widely-used and practical tools for relation extraction. In this section, we will introduce the key ideas behind each method and how they work.

5.1. spaCy

spaCy is an open-source software library for advanced natural language processing, which supports multiple languages. spaCy provides lots of linguistic annotations and tagging, including word types, dependencies, and relations between words. In this way, user could get an overview into a text’s grammatical structure (See Fig. 3).

The key idea of relation extraction in spaCy is to find the most-related words through analyzing the word types and dependency between words. The method is including three steps: finding all entities (noun), determining the subject of verbs, and extracting relations among entities.

The first step is to find all entities. This is achieved by selecting all noun words through the analysis on word types in spaCy. After we get all noun chunks, it is significant to decide which words should be the subjects. Such process is done in the second step. Through dependency parsing analysis, we select the ”nominal subject” and clausal modifier of noun (adjectival clause) as our subject, and at the same time, check word types and dependencies of corresponding verbs with our identified subject. Similarly, the last step of determining relations is also achieved by spaCy’s own analysis on words types and dependencies. We have classified the structure of sentence (active voice, passive voice, preposition-lead, etc), so that the relations can be extracted accurately.

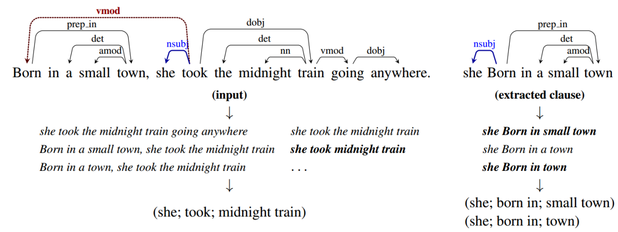

Figure 4. Visualized the procedures of Stanford OpenIE for extracting the triplets in sentences.

5.2. Stanford OpenIE

Stanford OpenIE [3] is written in JAVA and is one of the many annotators in Stanford Core NLP group which supports 6 languages, including Arabic, Chinese, English, French, German, and Spanish. It has two functions, which are to extract relation triples labeled with subject, verb, and relation, and returns extracted sentence fragments (clauses) from a given sentence. The most useful part of it comes in extracting triples without training set data and processing around 100 sentences per second per CPU core.

This tool works in the following procedure (Fig. 4):

- Split the given sentence into clauses by parsing through the dependency tree generated from the sentence. Since the clauses can be incomplete, the subject of the original sentence will be appended to this phrase to construct a clause.

- Clauses extracted will be shortened to its basic form by utilizing natural logic inference. For example, ”all cs students code in JAVA” can be seen as ”cs students code in JAVA”. But ”all UW Madison cs students code in JAVA” does not yield ”cs students code in JAVA”.

- Having shortened clauses, the system parses them into triplets by using 6 dependency patterns. If there are

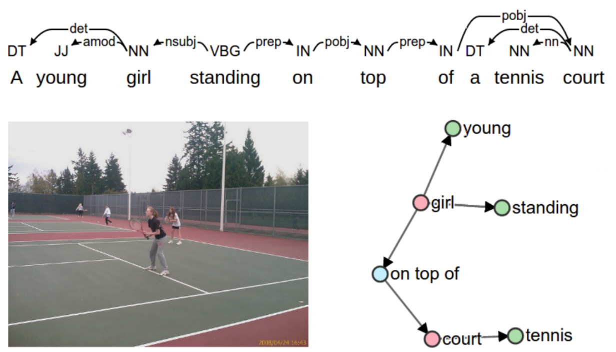

Figure 5. Visualized SPICE pipeline for extracting the triplets from a given sentence. The first stage (the top part) is to parse the dependency tree between the sentence words. The second stage (the bottom-right part) is to convert the tree into a set of triplets according to a set of pre-defined linguistic rules.

substructures that relation extraction is still possible, then Stanford openIE implements nominal relations 5 dependecy and 3 TokensRegex surface form patterns.

5.3. SPICE

SPICE [1] was designed for image caption evaluation. It's a widely-adopted metric for measuring the similarity between the model generated sentences and human annotated sentences, as the other language metrics (e.g., BLEU [18], CIDEr [22]). Its main idea is to extract the key concepts in the sentences, and compare the triplets between model generated captions and human annotated captions.

In our project, we only use SPICE to extract the sentence triplets. The procedure of extracting triplets from given sentence can be viewed in Fig. 5. SPICE uses a two-stage approach to extract triplets:

- In the first stage, given a sentence, SPICE uses a dependency parser [15] to identify the part-of-speech tags for each word, and construct the dependency tree between the word pairs, followed by 3 post-processing steps that simplify quantification modifiers, resolve pronouns and handle plural nouns.
- In the second stage, the resulting tree structure is parsed into the triplets (lemmatized objects, relations and attributes) according to 9 simple linguistic rules. Since the parsed triplets are already lemmatized, they can be directly merged to construct a graph. Hopefully, such graph is able to represent the main meaning of the given sentence.

6. Experiments and Results

In this section, we describe how we conducted experiments to select the best method, how we evaluated the predicted triplets versus the human-annotated triplets as well as the evaluation metrics and the evaluation results.

Methods \Data Sources	News
spaCy	**18.60 / 16.49 / 17.49**
Stanford OpenIE	8.02 / 19.59 / 11.38
SPICE	13.08 / 14.43 / 13.73

Table 4. The evaluation results of 3 different methods on ABC News. In each cell, the numbers are precision / recall / F1 score, respectively. The best method for this data source is highlighted.

Methods \Data Sources	Novel
spaCy	19.53 / 17.01 / 18.18
Stanford OpenIE	**23.81 / 47.62 / 31.75**
SPICE	1.50 / 1.36 / 1.43

Table 5. The evaluation results of 3 different methods on the novel. In each cell, the numbers are precision / recall / F1 score, respectively. The best method for this data source is highlighted.

We have collected and annotated 300 sentences in total with each data source 100 sentences. We further divide them into validation set (80 sentences for each source) and test set (20 sentences for each source). To select the best method, we applied 3 method candidates on the validation set and evaluated them using precision, recall and F1 score as the previous works. After evaluation, we selected the best method for each data source and used the predicted triplets of test set to construct the knowledge graph. We now introduce the details for evaluation processing and metrics.

Given the model generated triplets and human-annotated triplets, our goal is to measure how well they can be matched to each other. Before we match the triplet strings, we lemmatized each word in the triplets first (e.g., "holds" and "holding" are both converted into "hold"). After that, we used string matching to check whether the given two triplets can be matched. The extracted triplets are treated as correct cases if they are exactly matched to the labeled triplets as a whole, including subject strings, relation strings and object strings.

Following previous work [23, 1], the metrics include: precision, recall and F1 score. Precision is calculated as the ratio of correctly predicted triplets over all predicted triplets. Recall is calculated as the ratio of correctly predicted triplets over all human-annotated triplets. F1 score is calculated based on precision and recall. Generally, the higher the scores, the better performance the model has.

The results are shown in Table 4, Table 5, and Table 6. The best method for news, novel and image caption is spaCy, Stanford OpenIE and spaCy, respectively. We will analyze the results in Section 7 and visualize the constructed knowledge graph based on the predicted triplets on the test set data in Section 8.

7. Analysis and Discussion

According to the results in Table 4, Table 5, and Table 6, spaCy is the most robust method across different data

Methods \Data Sources	Caption
spaCy	**37.68 / 32.30 / 34.78**
Stanford OpenIE	19.87 / 19.25 / 19.56
SPICE	34.51 / 30.43 / 32.34

Table 6. The evaluation results of 3 different methods on the image captions. In each cell, the numbers are precision / recall / F1 score, respectively. The best method for this data source is highlighted.

sources. Specially, for the data source Novel, Stanford OpenIE outperforms spaCy significantly. We have 2 main observation and the analysis is presented as follows.

We found that Stanford OpenIE performed well on the novel data, but performed badly on the captions. We conjectured that OpenIE was designed for extracting triplets from the open domain sentences. Therefore it performs badly on captions which involves a more confined format of sentence structures. For example, the sentence "kitchen setting is with plotted plants" will be extracted into "kitchen setting; is with; plotted plants". However, the true label is "kitchen; with; plants". That's because OpenIE implements natural logic, that the adjective describing the object is included to describe the object as a unique entity. Another possible reason why the recall score was low is that OpenIE extracts clauses from sentences and adds the words. For example, it adds the word "is" to the word "with" in relations to make phrases into clauses. This would decrease the score based on the word matching during our evaluation.

On the contrary, SPICE performance was opposite to Stanford OpenIE. It performs particularly bad on the novel data set but performs closely with spaCy on caption data. The reason of this observation may be attributed to the original design goal of SPICE. SPICE was originally designed to capture the key concepts in the image captions. As the results, the entities in the extracted triplets tend to be short words and mostly are single words. This characteristic fits with the image caption data but not the novels. For instance, a sentence from the novel dataset, "Mr. Brunner was leading this trip", was extracted into "brunner; lead; trip" by SPICE. The extracted subject was shortened too much to match the correct subject "Mr. Brunner". However, SPICE was good at parsing confined sentence structures in the caption sentences which uses preposition descriptions a lot. Since the caption sentences are not overly complicated like the novel sentences, SPICE was able to correctly identify the triples.

8. Visualization

We constructed the knowledge graph using the predicted triplets after word lemmatization. The visualization of the knowledge graph is achieved through directed graph, where subject points to relation, and relation points to object. We used NetworkX[2] as our tool for the visualization of the knowledge graph. See Fig. 6, Fig. 7, and Fig. 8 for details.

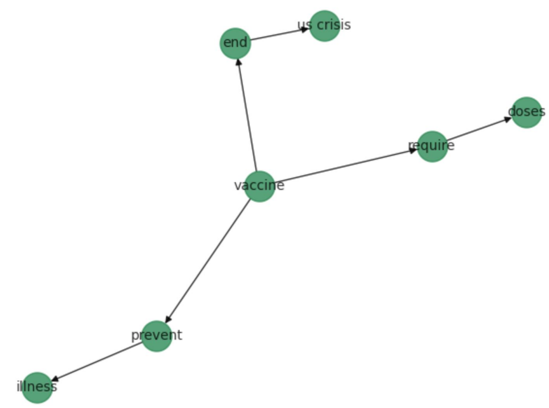

Figure 6. The visualization of a knowledge graph generated by ABC News with triplet extracted by the best method: spaCy

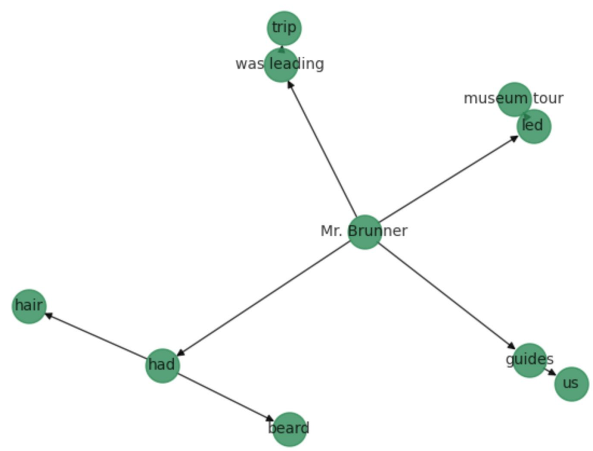

Figure 7. The visualization of a knowledge graph generated by novel (Percy Jackson & the Olympians: The Lightning Thief) with triplet extracted by the best method: Stanford OpenIE

The knowledge graphs using triplets extracted from different sentences helps people to quickly digest the overview of the sentences, and get the key relations between entities. The characteristics of directed graph also present people a clear view on the logic of sentences. With the help of such knowledge graphs, even several sentences with complicated structure can become straightforward to people and easy to comprehend the key idea or logic inside within seconds.

9. Conclusion

Our project goal is to identify the best relation extraction method in practice. To this end, we collected and annotated the daily life data, including news, novel and image captions. We have studied and evaluated 3 rule-based methods,

[2] https://networkx.org/

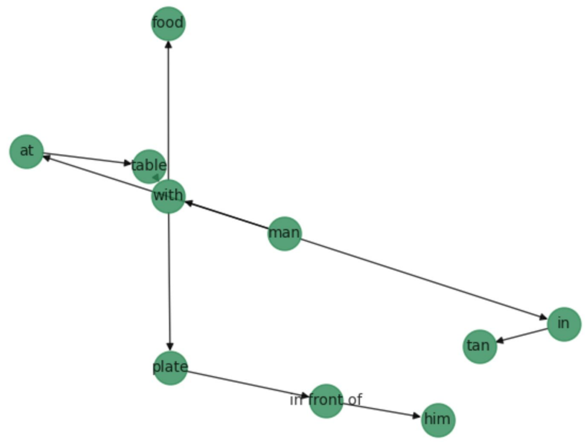

Figure 8. The visualization of a knowledge graph generated by human annotated captions of images with triplet extracted by the best method: spaCy

including spaCy, Stanford OpenIE, and SPICE, on our collected data. The results showed that spaCy performed the most stable across different data sources. Stanford OpenIE performed well on the data from novel, but bad on image captions. SPICE performed closely with spaCy across data sources but particularly bad on the sentences from novel. The main reason might be what the model was designed for and the target area where the models are used in. The analysis provided us more insights on how the variety of sentences can damage performances of models and what types of data the models can fit into. For future researches, we can change evaluation method, execute more model comparisons, or test the methods on more diverse datasets.

10. Contributions

This project can be divided into 7 parts: method survey, data preparation, model setup, method evaluation, result analysis, visualization and report writing. (1) Method survey was mostly done by Yiwu. (2) Data was collected and annotated by all team members. (3) Each team member set up a method and applied it on the validation data from 3 sources (Yuxin: spaCy, Ching-Wen: Stanford OpenIE, Yiwu: SPICE). (4) Method evaluation was conducted by Yiwu. (5) Result analysis was mostly done by Ching-Wen. (6) Visualization was mostly done by Yuxin. (7) The final report was written by all team members.

References

[1] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016.

[2] M. Andrews, Y. K. Chia, and S. Witteveen. Scene graph parsing by attention graph. 2019.

[3] G. Angeli, M. J. J. Premkumar, and C. D. Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, 2015.

[4] Y. S. Chan and D. Roth. Exploiting background knowledge for relation extraction. 2011.

[5] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[6] A. Culotta and J. Sorensen. Dependency tree kernels for relation extraction. 2004.

[7] B. Distiawan, G. Weikum, J. Qi, and R. Zhang. Neural relation extraction for knowledge base enrichment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 229–240, 2019.

[8] K. Fundel, R. Kuffner, and R. Zimmer. Relex-relation extraction using dependency parse trees. Bioinformatics, 23(3):365–371, 2006.

[9] X. Han, T. Gao, Y. Lin, H. Peng, Y. Yang, C. Xiao, Z. Liu, P. Li, M. Sun, and J. Zhou. More data, more relations, more context and more openness: A review and outlook for relation extraction. 2004.

[10] T.-K. Jenssen, A. Lægreid, J. Komorowski, and E. Hovig. A literature network of human genes for high-throughput analysis of gene expression. Nature Genetics volume 28, pages21–28, 2001.

[11] M. Jiang and J. Diesner. A constituency parsing tree based method for relation extraction from abstracts of scholarly publications. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, page 186–191. Association for Computational Linguistics, 2019.

[12] N. Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. IBM T. J. Watson Research Center, 2011.

[13] S. H. Kassani and P. H. Kassani. Building an ontology for the domain of plant science using protégé. *CoRR*, abs/1810.04606, 2018.

[14] H. Kilicoglu and S. Bergler. Effective bio-event extraction using trigger words and syntactic dependencies. Computational Intelligence, Volume 27, Number 4, 2011.

[15] D. Klein and C. D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st annual meeting of the association for computational linguistics*, pages 423–430, 2003.

[16] J. Kuang, Y. Cao, J. Zheng, X. He, M. Gao, and A. Zhou. Improving neural relation extraction with implicit mutual relations. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 1021–1032. IEEE, 2020.

[17] J. L. Martinez-Rodriguez, I. Lopez-Arevalo, and A. B. Rios-Alvarado. Openie-based approach for knowledge graph construction from text. *Expert Systems with Applications*, 113:339–355, 2018.

[18] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In

Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pages 311–318, 2002.

[19] H. A. M. G. S. R. M. S. P. Rebholz-Schuhmann, Dietrich; Kirsch. Ebimed-text crunching to gather facts for proteins from medline. Bioinformatics (Oxford, England), 2007.

[20] S. Schuster, R. Krishna, A. Chang, L. Fei-Fei, and C. D. Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, pages 70–80, 2015.

[21] S. Vashishth, P. Jain, and P. Talukdar. Cesi: Canonicalizing open knowledge bases using embeddings and side information. 2018.

[22] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[23] Y.-S. Wang, C. Liu, X. Zeng, and A. Yuille. Scene graph parsing as dependency parsing. 2018.

[24] D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. Journal of Machine Learning Research, 3(Feb):1083–1106, 2003.

[25] D. Zhou and Y. He. Extracting interactions between proteins from the literature. Journal of Biomedical Informatics 41 (2008) 393–407, 2007.