J. Int. Intl. Sys. Soft, Vol. 1, No. 1, 2022
ejournal.akakom.ac.id/index.php/JISS/index

JISS Journal of Intelligent Software Systems

Journal of Intelligent Software System – JISS

Building a Knowledge Graph on Video Transcript Text Data

Original Article doi: 10.26798/jiss.v1l1.585

submit:2022-06-14, Accepted:2022-07-06, Publish:2022-07-23

Bagas Triaji[1,2*], Widyastuti Andriyani[2†], Bambang Purnomosidi DP[2‡], and Faizal Makhrus[3§]

1 Student, Master in Information Technology Universitas Teknologi Digital Indonesia, Yogyakarta, Indonesia

2 Master in Information Technology Universitas Teknologi Digital Indonesia, Yogyakarta, Indonesia

3 Department of Computer Science and Electronics Faculty of Mathematics and Natural Sciences

Abstract: Youtube is a video platform which not only provides entertainment but also education in which knowledge can be dug based on video transcripts. The results of this knowledge can be formed as a knowledge graph to build a knowledge base that saves storage space. Moreover, it can be used for other purposes such as recommendation systems and search engines. Prosen built a knowledge graph using NLP to extract the text by identifying the subject-verb-object (SVO) and stored in the graph database. The construction of a knowledge graph on a Youtube video transcript was successfully carried out. However, there are still obstacles in the process of extracting text using NLP which is less optimal so it is possible that there is still a lot of knowledge that has failed to be obtained.

Keywords: knowledge graph, graph, neo4j, nlp

* *E-mail: student.bagastriaji@mti.akakom.ac.id*
† *E-mail: widya@utdi.ac.id*
‡ *E-mail: bpdp@utdi.ac.id*
§ *E-mail: faizal_makhrus@ugm.ac.id*

1. Introduction

Currently, the internet contributes to the development of education globally with educational content on various platforms. YouTube is a video platform that provides not only entertainment but also educational videos which assist students to get learning materials [1]. The videos can be extracted knowledge based on video transcripts or subtitles. The results of this knowledge extract can be figured a knowledge graph. It is done so knowledge base can be built to save storage space and can be used for other purposes such as recommendation systems and search engines.

A knowledge graph is a directed labeled graph where the labels have a well-defined meaning. A directed labeled graph consists of nodes, edges, and labels. A knowledge graph (KG) can represent semantic data arranged with 3 components such as subject (s), predicate (p), object (o) [2]. Another opinion is that knowledge graphs are a compilation of interrelated facts which describe real-world entities, events, or objects and their relationships in a set-up that humans and machines can understand.

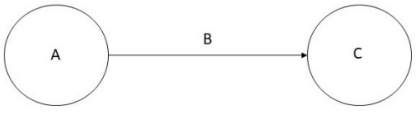

Figure 1. Knowledge in the form of graph

A knowledge graph consists of three main components: nodes, edges, and labels. Any object, place or person can be a node. An edge defines the relationship between nodes. As shown in Figure 2, node A represents the subject, edge B represents the predicate and node C represents the object. Semantic data is stored with graph models in graph databases such as Neo4J, JanusGraph, TigerGraph. This database is designed to store graph data and perform

data query operations. A graph is a discrete structure formed from a tuple, which is a set of vertices and a set of edges that connect the vertices of the graph. Graph notation is $G = (V, E)$, in which V is the set of vertices and E is the set of edges. In other terms, it is also called a node and connecting between nodes is called a relationship [3, 4]. A graph whose edge (E) has a direction orientation. The directed side is referred to as the arch. In a directed graph, (u, v) and (v, u) represent two different arcs. For vertices (u, v), the vertex u is called the origin vertex and the vertex v is called the terminal vertex.

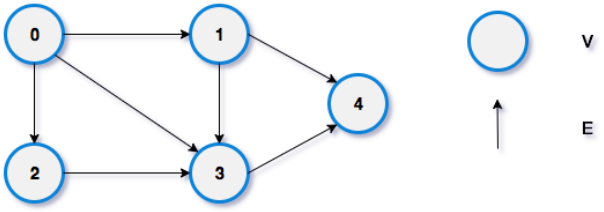

Figure 2. Directed Graph

In Figure 2, if a set is formed, it will be formed as follow:

$$\begin{aligned} V &= 0, 1, 2, 3, 4 \\ E &= [(0,1), (0,2), (0,3), (1,3), (1,4), (2,3), (3,4)] \end{aligned} \tag{1}$$

In this study, Natural Language Processing (NLP) was applied to extract knowledge from the text. NLP is a computer way of processing human language in meaningful way. NLP is generally used to derive the semantic pattern of a text or speech and convert it to a more structured format so that it can be processed by a computer [2]. The video transcript text data employ the engineering data search keywords and the top 10 videos from the search results will be used.

2. Theoretical Review

Research on building the knowledge graphs was carried out by several previous researchers with similar focus on the problem, which is building knowledge graphs manually will take a lot of time. In order that the concept of forming knowledge graphs automatically minimizes manual annotations by humans, such as Application of NER and path-ranking prediction of two-entity relations [5]. A knowledge graph built on entity relationships and a visual analysis platform [6]. Building a knowledge graph automatically applies Named Entity Recognition (NER) to extract material concepts and association rule mining to obtain material prerequisites [7, 8].

3. Methods

The data taken from YouTube is the transcript/subtitle data on the video. This transcript is in the form of conversational text either made by the video maker or auto-generated text by the YouTube feature. In Figure 3, it describes the flow of data retrieval and extract.

Figure 3. Youtube transcript data extraction flow

Both YouTube-search and YouTube-transcript-api provide text data results in JSON (Javascipt Object Notation) format which are then converted to data structures list and dictionary in python. From the dictionary, people can choose which attributes are required. In this case, the required attribute is text and will be saved as a text file.

The transcript text file will be processed using Natural Language Processing (NLP) to obtain entities and their relation and then the data is stored in the form of graphs. Text processing using NLP requires several stages so that it will construct data in the form of entities and their relation. The stages of text processing are represented in Figure 4 in which the collected text

will be processed by NLP in eight stages until it becomes a data graph.

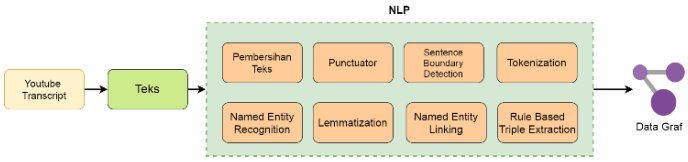

Figure 4. **Text processing stages**

The extracted text of the YouTube document and transcript contains several punctuation marks, spaces, white space and unreadable characters. Text cleaning uses a regular expression (Regex) to detect certain characters. Punctuator is a process of returning the missing punctuation marks, especially the dot at the end of the sentence [9]. This process is very much needed on YouTube transcript data because the transcript text is not composed in complete sentences because the text is based on the time speech of each video frame. With punctuation marks, the sentences that are arranged will be clearer so that it can be easier in the next process, called detecting sentences.

Table 1. **Punctuator process result**

Raw Youtube Transcript	Punctuation using the Punctuator
in this video we will take a look at how you can run spark inside of a cluster now in this specific example i be use spark in in a cluster that have a standalone cluster so it have not use your own reason so again it should basically work the same regardless of the underlie schedule that you have set up for the resource manager ...	In this video, we will take a look at how you can run spark inside of a cluster. Now, in this specific example, i be use spark in in a cluster that have a standalone cluster, so it have not use your own reason. So again, it should basically work the same regardless of the underlie schedule that you have set up for the resource manager. ...

Table 1 illustrates the comparison between text from YouTube which has not undergone punctuation restoration and after restoration of punctuation using a punctuator; it will be

easier to detect sentences in the next process. Sentence Boundary Detection (SBD) serves to identify sentences in a collection of text or documents. If it is not in the form of a sentence, it will be difficult in the process of getting information or parsing by Natural Language Processing (NLP). [10]. The success rate of SBD will be higher if the text is complete with punctuation marks, especially commas and periods, so that a set of text or paragraphs can be separated into each sentence.

Table 2. Punctuator process result

In this video, we will take a look at how you can run spark inside of a cluster. Now, in this specific example, i be use spark in in a cluster that have a standalone cluster, so it have not use your own reason. So again, it should basically work the same regardless of the underlie schedule that you have set up for the resource manager. ...	0 In this video, we will take a look at how you can run spark inside of a cluster. 1 Now, in this specific example, i be use spark in in a cluster that have a standalone cluster, so it have not use your own reason. 2 So again, it should basically work the same regardless of the underlie schedule that you have set up for the resource manager.

Table 2 in the Text column, illustrates the differences the text before the SBD process is carried out, while the results are in the SBD column and it seems it has been split into several sentences.Tokenization is the process of separating sentences into small parts in the form of words, numbers and punctuation marks. The tokenization process is carried out in stages, starting from splitting based on whitespace, then checking each result of the separation. If a punctuation mark or number is found, the part will be separated into small parts. NER in this study is used to identify entities in text data extracted from YouTube. Entities can be people, places, organizations, times, brands and so on [11]. The text that has been extracted from the YouTube transcript using NER produces various entity labels. However, only a few entities were successfully recognized by the SpacyNLP model. Named entity linking is the process of linking

entities in a document to knowledge bases such as Wikipedia and DBPedia. However, NEL can also be used to identify entities by matching words with entities in the knowledge base [12]. In the previous NER process, there were very few entity identifications, so the NEL process was carried out to map the wikidata or called as wikification, so that more entities were obtained as shown in Table 3.

Table 3. Entity data identified using the NEL process with wikidata

wikidata_url	wikidata_id	wikidata_label_entity
https://www.wikidata.org/wiki/Q34508	34508	video
https://www.wikidata.org/wiki/Q206637	206637	computer cluster
https://www.wikidata.org/wiki/Q42253	42253	Uniform Resource Locator
https://www.wikidata.org/wiki/Q460584	460584	Scala
https://www.wikidata.org/wiki/Q28865	28865	Python
https://www.wikidata.org/wiki/Q18109	18109	operating system shell
https://www.wikidata.org/wiki/Q47146	47146	user interface
https://www.wikidata.org/wiki/Q7573619	7573619	Apache Spark
...	...	...
https://www.wikidata.org/wiki/Q865746	865746	metric function
https://www.wikidata.org/wiki/Q35140	35140	performance

he process of obtaining the knowledge in the text can be seen from the sentence structure. The arrangement of words in the form of a subject, verb and object can represent knowledge in a sentence. Then a pattern or rule base is required when identifying SVO. In this stage, the researches use dependency parsing and part-of-speech (POS) tagging to find out the position of words and their relationship to other words in a complete sentence. Identification of the subject uses the dependency parsing process with the nsubj tag, while the object uses the dobj tag. Objects can be more than one word by including a compound tag in the word before the dobj tag. Meanwhile, the relation between subject and object is connected using a verb which is between nsubj and dobj.

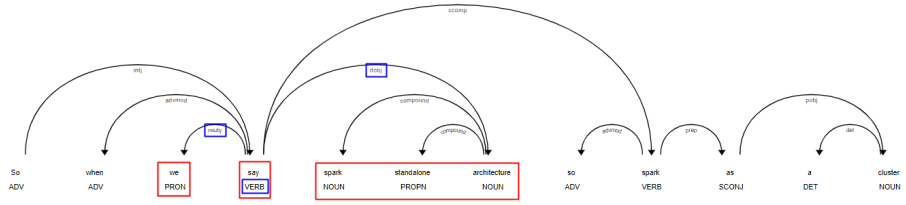

Figure 5. **SVO identification process using Dependency Parsing and POS Tagging**

Identification the verb applies the POS tagging with the tag name VERB. Therefore, a sentence is shown in Figure 5 when SVO is extracted, the result is (subject=[we], verb=[say], object=[spark, standalone, architecture]).

4. Results

The data from text processing is remaining in the form of a table, and then it will be converted into graph data consisting of nodes and edges with a directed graph type.

The YouTube transcript data is presented in Table 4. There is also data that there is only edge_1 and node_1, while edge_2 and node_2 are empty. The data is data from entity identification using entity linking (NEL) while other data with complete columns, is the result of SVO extract. Then the existing table will be formed as a data graph as shown in Figure 6.

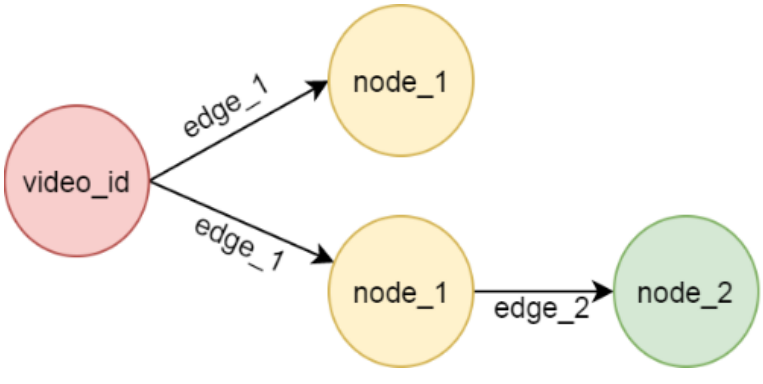

Figure 6. **YouTube transcript graph data model**

Table 4. **The data of extracted result from YouTube video knowledge**

video_id	edge_1	node_1	edge_2	node_2
VApvm6llNcY	has entity	person	go	to discuss about
VApvm6llNcY	has entity	spark	have	deployment modes
VApvm6llNcY	has entity	person	say	spark standalone architecture
VApvm6llNcY	has entity	person	instal	spark
VApvm6llNcY	has entity	person	do	activities
VApvm6llNcY	has entity	person	use	hadoop technology
VApvm6llNcY	has entity	which	mean	hadoop technology
VApvm6llNcY	has entity	person	call	that
VApvm6llNcY	has entity	person	build	clusters
VApvm6llNcY	has entity	person	distribute	processing jobs execution things
VApvm6llNcY	has entity	person	use	to integrate with hadoop
VApvm6llNcY	has entity	person	call	that that
VApvm6llNcY	has entity	apache hadoop		
VApvm6llNcY	has entity	computer cluster		
VApvm6llNcY	has entity	daemon		

From the data model in Figure 6, it will turn out a data graph like Figure 7. The red nodes are YouTube videos connected to several entities with blue nodes connected to the has_entity edge. Entities resulting from the SVO extract process will then be connected to orange nodes with various edge names. The transformation process from tabular data until it is entered into the graph database is shown in Figure 8. Storage in the graph database is practical so that the knowledge graph is stored and can be updated. The knowledge graph of the videos that have been formed can be calculated the similarity level of a video with another. It can use Jaccard Similarity by measuring the similarity of two sets. Then calculating the size of the intersection divided by the union of the set [13].

The set that will be used in calculating the Jaccard algorithm is a node connected by an edge. The way is by removing the edge name so that the similarity value is higher. Therefore,

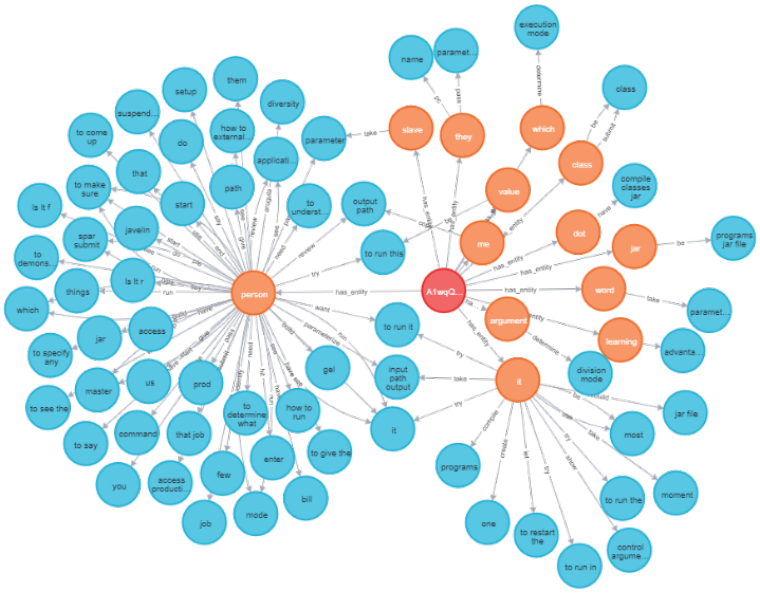

Figure 7. The results of the transformation of the Youtube transcript table data into a graph

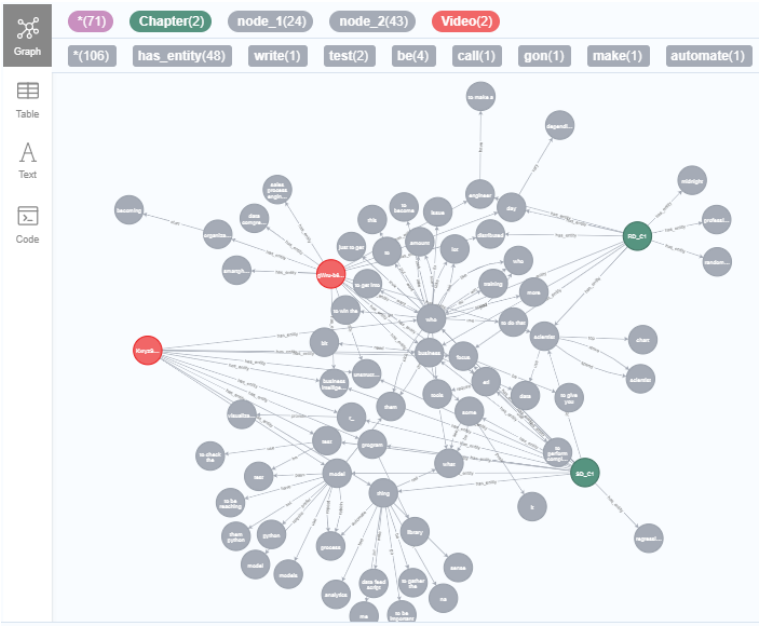

Figure 8. Knowledge graph on Neo4J

two sets of graphs by taking video samples with Id=VApvm6llNcY denoted $G_1(E)$ and compared

to video Id=VYNsaR-gOsA denoted $G_2(E)$. Formula :

$$
\begin{aligned}
G_1(E) &= \begin{bmatrix} (agenda, to\ fetch\ top), (asking, ram), (availability, to\ ask\ this), \\ (block, b0\ copy), (cluster, resource),\ \dots, (cluster, that) \end{bmatrix} \\
G_2(E) &= \begin{bmatrix} (application, to\ package\ this), (bin, win), (bin, what), (change, effect), \\ (command, svt),\ \dots, (file, checking) \end{bmatrix}
\end{aligned} \tag{2}
$$

Furthermore, these two sets will be compared using the Jaccard algorithm with the following mathematical calculation steps: Formula:

$$
\begin{aligned}
J(G_1(E), G_2(E)) &= \frac{[G_1(E) \cap G_2(E)]}{[G_1(E) \cup G_2(E)]} = \frac{[G_1(E) \cap G_2(E)]}{[G_1(E)] + [G_2(E)] - [G_1(E) \cap G_2(E)]} \\
J(G_1(E), G_2(E)) &= \frac{4}{242 + 61 - 4} \\
Jaccard\ Score &= 0.03040816326530612
\end{aligned} \tag{3}
$$

The score from the calculation of the Jaccard value of the two graphs $J(G_1, G_2)$ is used as the basis for determining the level of similarity between videos. The higher the value with a maximum values of 1, the more similar the videos that have been compared. It can be useful to recommend other related videos based on the context of the video content.

5. Conclusions

The process of extracting unstructured text into a knowledge graph is a big challenge. Although the construction of a knowledge graph on the YouTube video transcript was successfully carried out, there are still problems with the text extract process using NLP which is less optimal. It is possible that there are many texts that fail to be built in the form of knowledge.

Exactly, this research still has shortcomings and should be much improved in the future, especially in creating knowledge graphs using NLP, which is in the SVO extraction section using a limited rule based technique, resulting in many texts failing to extract.

References

[1] E. T. Maziriri, P. Gapa, and T. Chuchu, "Student Perceptions Towards the use of YouTube as An Educational Tool for Learning and Tutorials," Int. J. Instr., vol. 13, no. 2, pp. 119–138, Apr. 2020, doi: 10.29333/iji.2020.1329a.

[2] T. Al-Moslmi, M. Gallofre Ocana, A. L. Opdahl, and C. Veres, "Named Entity Extraction for Knowledge Graphs: A Literature Overview," IEEE Access, vol. 8, no. February, pp. 32862–32881, 2020, doi: 10.1109/ACCESS.2020.2973928.

[3] F. Daniel and P. N. L. Taneo, Teori Graf. Sleman: Deepublish, 2019.

[4] R. J. Wilson, Introduction to Graph Theory, Fourth., vol. 148. Harlow: Oliver & Boyd, 1972.

[5] Y. Jia, Y. Qi, H. Shang, R. Jiang, and A. Li, "A Practical Approach to Constructing a Knowledge Graph for Cybersecurity," Engineering, vol. 4, no. 1, pp. 53–60, Feb. 2018, doi: 10.1016/j.eng.2018.01.004.

[6] R. J. Wilson, Introduction to Graph Theory. 1996.

[7] P. Chen, Y. Lu, V. W. Zheng, X. Chen, and B. Yang, "KnowEdu: A System to Construct Knowledge Graph for Education," IEEE Access, vol. 6, pp. 31553–31563, 2018, doi: 10.1109/ACCESS.2018.2839607.

[8] P. Chen, Y. Lu, V. W. Zheng, X. Chen, and X. Li, "An automatic knowledge graph construction system for K-12 education," in Proceedings of the Fifth Annual ACM Conference

on Learning at Scale - L@S '18, 2018, pp. 1–4, doi: 10.1145/3231644.3231698.

[9] O. Tilk and T. Alumäe, "Bidirectional Recurrent Neural Network with Attention Mechanism for Punctuation Restoration," in Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, Sep. 2016, vol. 08-12-Sept, pp. 3047–3051, doi: 10.21437/Interspeech.2016-1517.

[10] G. Donabauer, U. Kruschwitz, and D. Corney, "Making Sense of Subtitles: Sentence Boundary Detection and Speaker Change Detection in Unpunctuated Texts," in Companion Proceedings of the Web Conference 2021, Apr. 2021, pp. 357–362, doi: 10.1145/3442442.3451894.

[11] A. Goyal, V. Gupta, and M. Kumar, "Recent Named Entity Recognition and Classification techniques: A systematic review," Comput. Sci. Rev., vol. 29, pp. 21–43, Aug. 2018, doi: 10.1016/j.cosrev.2018.06.001.

[12] E. Yaman and K. Krdzalic-Koric, "Address entities extraction using named entity recognition," Proc. - 2019 Int. Conf. Futur. Internet Things Cloud Work. FiCloudW 2019, vol. 13, pp. 13–17, 2019, doi: 10.1109/FiCloudW.2019.00016.

[13] S. Fletcher and M. Z. Islam, "Comparing sets of patterns with the Jaccard index," Australas. J. Inf. Syst., vol. 22, pp. 1–17, Mar. 2018, doi: 10.3127/ajis.v22i0.1538.