

Using LLMs for Ontology generation by video summarization

Khaled Omar

kh.om.mail@gmail.com

Damascus University

Article

Keywords: LLM, Gemini, video summarization, ontology, RDF

Posted Date: September 25th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10802097/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Using LLMs for Ontology generation by video summarization

Dr. Khaled Omar

Syria, Damascus

Damascus University

Information technology engineering college

Email : kh.om.mail@gmail.com

Abstract—Ontology defined as abstraction of concepts and presents the relations between these concepts, it is a domain oriented, the construction and validation of ontologies is a hard work and it needs a lot of resources from domain experts to represent the concept of a specific domain in a computer understandable format, many algorithms have been developed to automatic construction of ontologies, almost all of these developed algorithms takes textual input and convert it in any of ontology representation languages such as RDF, or OWL, in this paper we are presenting a new algorithm for automatic ontology construction from video, the input of this algorithm is online YouTube video URL, and the output is an ontology which describes input video, the proposed algorithms utilizes large language models in its phases, the proposed algorithms contains two main phases, the first one converts the input video to a textual summary contains the domain of the video and the main detected entities in this video, the second phase takes the video textual summary as input and converts this summary into ontology which representing this summary , the output of the second phase is RDF format, we have tested our proposed algorithm on a public video dataset available on YouTube, this video series belong to a sport domain, the evaluation of our proposed algorithm accomplish 90% accuracy rate.

Keyword: LLM, Gemini, video summarization, ontology, RDF.

I. INTERODUCTION

Ontology is the core element in the semantic web , an ontology encompasses the following components [1]:

- Concepts
- Relations
- Objects

Ontologies offer a structured approach to knowledge representation, providing several advantages:

- Standardized communication and data exchange: by modeling the knowledge into ontologies, this helps in data sharing for these ontologies.
- decision-making support: for example, in healthcare sector ontologies help in medical diagnosis.
- Efficient knowledge organization and discovery: by converting textual data into ontologies this help in the retrieval process for finding the relevant topics and relevant terms in data retrieval.
- Domain-specific modeling: for example, ontologies in healthcare are specific to this domain and can be used to develop new systems based on these ontologies.

Constructing ontologies is a complex and resource-intensive process, requiring expert human resources for both development and validation. Based on this, research efforts have increasingly focused on automating ontology construction. many algorithms have been developed in this domain, most of the developed methods take textual input and produce a domain ontology expressed in languages such as RDF [2] or OWL [3].

In recent years, large language models have been utilized across various research domains. Their primary added value lies in their training on extensive datasets and their capacity to perform a wide range of tasks. This capability has made them especially useful in natural language processing applications.

Video summarization is the process of generating a concise summary of video content. Traditionally, this involved splitting videos into frames, retaining only the most significant ones by discarding duplicates, extracting captions from the unique frames, and producing a summary based on those captions. More recently, large language models have been increasingly employed for summarization tasks, owing to their effectiveness in generating coherent and meaningful text.

II. Related work:

Building ontologies manually is time-consuming and often impractical given the vast amounts of data in fields like Generative AI. To address this, researchers have developed automated and semi-automated methods for ontology construction. These algorithms analyze textual data—such as research papers, technical documents, and domain-specific corpora—to identify concepts, hierarchies, and relationships with minimal human intervention. Below, we explore some of the most influential algorithms in this space, highlighting their methodologies, advantages, and drawbacks.

1. Text2Onto[4][5]: Adaptive Probabilistic Ontology Learning

One of the earliest and most widely cited approaches is Text2Onto, introduced by Cimiano and Völker in 2005. Unlike rigid rule-based systems, Text2Onto employs statistical methods to detect terms and relationships dynamically. It processes raw text using natural language techniques like part-of-speech tagging and dependency parsing, then applies machine learning to refine concept extraction.

2. OntoLearn[6]: Semantic Filtering for Specialized Domains

Developed by Navigli and Velardi, OntoLearn (2004) focuses on extracting domain-specific terminology by leveraging semantic networks such as WordNet. The algorithm filters out generic terms, retaining only those relevant to the target field. This makes it particularly effective for technical disciplines where precision is critical.

3. LExO :Logic-Based Relation Extraction[7]

LExO takes a different approach by combining lexico-syntactic patterns with formal logic. Instead of relying solely on statistics, it identifies relationships like "is-a" or "part-of" through predefined grammatical structures. This method yields highly accurate ontologies with clean hierarchical mappings—ideal for applications demanding strict formalism, such as biomedical knowledge graphs.

4. DeepOnto[8]: Modern Deep Learning for Ontology Construction

Recent advances in neural networks have enabled more sophisticated solutions like DeepOnto (Chen et al., 2020), which uses transformer models (e.g., BERT[9]) to infer contextual relationships. By training on large corpora, DeepOnto captures subtle semantic connections that traditional methods miss—for example, linking "diffusion models" to "image generation" based on latent patterns in research literature. While powerful, this approach demands substantial computational resources and annotated training data, posing challenges for smaller organizations.

Each algorithm serves distinct needs:

- Text2Onto excels in dynamic environments where data evolves rapidly.
- OntoLearn is preferable for niche domains with well-defined vocabularies.
- LExO suits projects requiring rigorous logical consistency.
- DeepOnto leads in scalability and contextual awareness but requires significant infrastructure.

Hybrid systems that merge rule-based and machine learning techniques are gaining traction, offering a balance between precision and adaptability. Future advancements may focus on reducing dependency on labeled data and improving multilingual support to make automated ontology construction more accessible.

III. METHODOLOGY

The following workflow contains the main steps on our proposed algorithm:

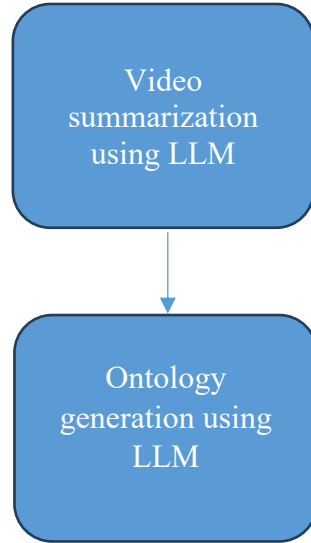


Figure1.proposed workflow

1- **Video Summarization using LLM:**

in this step we will summarize the input video, the algorithm input is a video URL from YouTube, and the output of this step is a structured summary which contains the following points:

- Main Topic/Theme
- Key Points
- Important Details
- Conclusions/Takeaways

The algorithm makes the video summarization by using LLama 3.2 [10] as large language model and YouTube Transcript API [11]to get the transcript of the input video, following workflow describes the steps of video summarization:

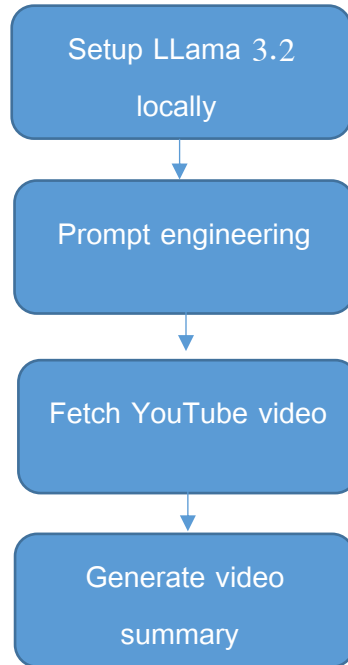


Figure 2. Video Summarization using LLM

1-1 Setup Setup LLama 3.2 locally : the aim of this step is to be able to run Llama 3.2 large language model locally(not on the cloud), so to initiate using Llama 3.2 locally Ollama application installed in the local system and then the LLama 3.2 model is downloaded in local System.

1-2 Prompt engineering : to be able to use any large language model to fit the purpose of using it we need to setup a suitable prompt to the model, that this prompt will be as a guideline to the model to generate the expected output, so in our developed system we will use the following prompt templates:

- One_chunk_prompt_template :

"""

Summarize the following part of a YouTube video transcript:

"{text}"

KEY POINTS AND TAKEAWAYS:

"""

"{text}" is parameter for the prompt, and it holds the string of one YouTube video transcript.

- Combine_chunks_prompt_template:

Create a detailed summary of the YouTube video based on these transcript summaries:

"{text}"

Please structure the summary as follows:

1. Main Topic/Theme
2. Important Named Entities

- 3. Important Details
- 4. Conclusions/Takeaways

DETAILED SUMMARY:

""

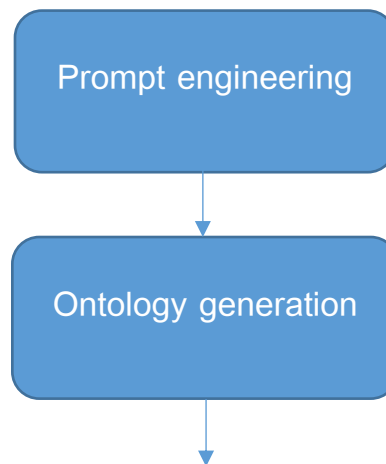
"{text}" is parameter for the prompt, and it holds all string chunks of YouTube video transcript
Fetch YouTube video

1-3 Fetch YouTube video : in this step the video is fetched from YouTube by passing the URL of the video and also by using YouTube transcript API the description of the video is also downloaded.

1-4 Generate video summary : in this step the summary of the video is generated, this summary is generated based on the large language model that we are using LLama 3.2, and based on the prompts that we have set to this model, the first step the algorithm generates a summary for each transcript chunk, then the model generate a full summary based on chunks summaries, we have used chunks to divide the full transcript of the video to a substring.

2- Ontology generation using LLM

In this step the algorithm takes the video summary which generated from step 1 (**Video Summarization using LLM**) and **base on this summary the ontology of the video is generated**, in this step we will use Gemini Pro 002[12] as large language model, the workflow of ontology generation explained in the following figure:



Ontology refinements

Figure 3. Video Summarization using LLM

2-1 Prompt engineering:

In this step the prompt is set to the used large language model which is Gemini Pro 002, the prompt takes two inputs the first one is the domain (**{videoDomain}**) and the second one is the list of the detected named entities of the video summary(**{concepts}**)

The full prompt for the ontology creation is as the following :

“””

Task: Generate an ontology representing the knowledge domain of **{videoDomain}**

Constraints:

Domain Analysis:

Carry out a comprehensive examination of the **{videoDomain}** domain to detect and specify:

Fundamental **{concepts}** and areas of expertise

The connections between these **{concepts}**

Ontology Population:

Fill the ontology with occurrences of specific **{concepts}**, guaranteeing correct categorization and allocation of property values.

Label each of the **{concepts}** with meanings, explanations, and pertinent cases.

Point to external sources where relevant (e.g., official manuals, instructional guides, sector norms).

Generate the output as a RDF (Turtle) format.

“””

2-2 Ontology generation:

In this step the Gemini Pro 002 model is called after feeding it with the prompt with parameters from previous step, and ontology output will be in RDF Turtle format.

2-3 Ontology refinements:

The goal of this step is to enhance the generated ontology, this enhancement is done as the following steps:

- Iterative prompting to generate parts of the ontology and then combining it, these can be done by generate a small ontology for sub sets of concepts and then combine the generated ontologies.

- Validating the generated ontologies relationships, by using prompts to validate the relationships between concepts.

IV. Evaluation

Data set

We have tested our developed algorithm on a dataset contains about 100 videos, these videos belong to a sport domain, and these videos are from YouTube videos, for each video the algorithm takes video URL as input and it get the transcript of it and then it generates the summary of input transcript and then it generated the RDF ontology format of the video.

Evaluation metrics

We have chosen the following metric for evaluation our proposed algorithm.

For evaluate the generated summary, we have used the cosine similarity formular to calculate the similarity between the video title and the generated summary of the input video, that this formula usually used for calculate the similarity between texts[13], as the following[14]:

$$\text{Precision of summary} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}},$$

Where:

A: is the vector of the generated video summary

B: is the vector of the video title

The **Precision** of the generated summaries was about 95%.

For evaluation the generated ontologies based on the video's summaries, we have used the following formula:

$$\text{Precision} = (\text{automatic ontologies} \cap \text{manual ontologies}) / \text{automatic ontologies}$$

Where :

automatic ontologies: is the ontologies that generated automatically by the algorithm.

manual ontologies: is the ontologies that generated by human expert.

Based on the precision formula above, our developed algorithm reached 85% accuracy rate.

I. DISCUSSION AND CONCLUSIONS:

In this paper, we have developed a new algorithm for automatic ontology construction based on video input, this algorithm contains mainly two phases the first one is the video summarization phase, in this phase the algorithm depends on new technology for generating the summary of the video, it utilized large language models in this task, LLama 3.2 has been used to generate the summary, also this model was launched locally to avoid any fees of using large language models on cloud, the generated summary was focused on the video domain and also it contains the named entities, the needed format of the summary was adopted by the prompt that was set to LLama 3.2 model, the second phase takes the video textual summary and generates the represented ontology in RFD format, for implementing this task a Gemini Pro 002 large language model was used after setting a detailed prompt to this model, the new proposed algorithm invest new technology for ontology generation and enhancement, and by usage of this algorithm will be minimize needed resources for ontology creation, and this will make the integration between domain ontologies more quicker and smother, the proposed algorithm has been tested on a public video dataset from YouTube (dataset size is about 100 video in sport domain), the validation of our proposed algorithm reach to 90% in accuracy rate, the developed algorithm can be utilized in any domain such as the medical domain.

The future work will focus on expand the ability for automatic ontology integration between domains ontologies, and we will utilize large language models to achieve this goal

Funding Declaration: no funding was received for implementing this research.

Data availability: The datasets used and/or analyzed during the current study available from the corresponding author on reasonable request.

References

- [1]-Muñoz, Ana & Aguilar, Jose. (2007). A general Ontology for Intelligent database. 10.13140/2.1.5182.5600.
- [2]-RDF - Semantic Web Standards. (n.d.). <https://www.w3.org/RDF/>
- [3]-OWL 2 Web Ontology Language Document Overview (Second Edition). (n.d.). <https://www.w3.org/TR/owl2-overview/>
- [4]-Cimiano, P., Völker, J. (2005). Text2Onto. In: Montoyo, A., Muñoz, R., Métais, E. (eds) Natural Language Processing and Information Systems. NLDB 2005. Lecture Notes in Computer Science, vol 3513. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11428817_21
- [5]-Cimiano, Philipp & Völker, Johanna. (2005). Text2onto. Nat Lang Proces Inform Syst. 3513. 227-238. 10.1007/11428817_21.
- [6]-Paola Velardi, Stefano Faralli, Roberto Navigli; OntoLearn Reloaded: A Graph-Based Algorithm for Taxonomy Induction. Computational Linguistics 2013; 39 (3): 665–707. doi: https://doi.org/10.1162/COLI_a_00146
- [7]-Bellandi, A. LexO: an open-source system for managing OntoLex-Lemon resources. Lang Resources & Evaluation 55, 1093–1126 (2021). <https://doi.org/10.1007/>
- [8]-He, Yuan & Chen, Jiaoyan & Dong, Hang & Horrocks, Ian & Allocca, Carlo & Kim, Taehun & Sapkota, Brahmananda. (2024). DeepOnto: A Python package for ontology engineering with deep learning. Semantic Web. 15. 1-14. 10.3233/SW-243568.
- [9]-Devlin, Jacob & Chang, Ming-Wei & Lee, Kenton & Toutanova, Kristina. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 10.48550/arXiv.1810.04805.
- [10]-meta-llama/Llama-3.2-1B · Hugging Face. (2024, December 6). <https://huggingface.co/meta-llama/Llama-3.2-1B>
- [11]-Captions | YouTube Data API | Google for Developers. (n.d.). Google for Developers. <https://developers.google.com/youtube/v3/docs/captions>
- [12]-Gemini. (n.d.). Google DeepMind. <https://deepmind.google/models/gemini/>
- [13]- Omar, Khaled & Kawas, Mohamad & Alkhatib, Bassel. (2024). Multi-Layer Algorithm for ICD prediction. 20. 394-404.
- [14]-Alake, R. (2025, March 5). Understanding Cosine similarity and its applications. Built In. <https://builtin.com/machine-learning/cosine-similarity>

