

Code Explanation:-

The provided code is intended to create a Knowledge Base (KB) or Knowledge Graph from textual inputs such as short text, long text, and web articles. The process involves using a language model to generate relations from the input text and then converting these relations into a structured KB format. The KB consists of entities, relations, and information about the sources (articles) from which the relations were extracted.

1- extract relations from model output(text): This function takes a generated text as input and extracts relations from it. The input text is processed to identify triplets using special tokens "<triplet>", "<subj>", and "<obj>". It then extracts the subject, relation, and object components of each triplet and returns a list of dictionaries, each containing the head, type (relation), and tail of the relation.

2- class KB(): This class represents the Knowledge Base and contains methods to manage the entities, relations, and sources. It has the following methods:

- **init (self):** Initializes an empty Knowledge Base with empty lists for entities and relations.
- **are relations equal(self, r1, r2):** Compares two relations (dictionaries) to check if they are equal based on their "head," "type," and "tail" attributes.
- **exists relation(self, r1):** Checks if a relation already exists in the Knowledge Base.
- **add relation(self, r):** Adds a relation to the Knowledge Base if it doesn't already exist. If the relation exists, it merges the spans (meta data) associated with the relation.
- **print(self):** Prints the entities and relations in the Knowledge Base.

3- from small text to kb(text, verbose=False): This function takes short text as input and generates a Knowledge Base from it. It tokenizes the text using a tokenizer, generates relations using a language model, and then adds these relations to the Knowledge Base using the 'KB' class.

4- from text to kb(text, span length=128, verbose=False): This function takes long text as input and splits it into spans of a specified length. It then generates relations for each span and creates a Knowledge Base using the 'KB' class. Spans allow processing long text in smaller chunks, making it possible to handle longer documents.

5- from text to kb(text, article url, span length=128, article title=None, article publish date=None, verbose=False): This function is similar to the previous one but includes additional information about the source article, such as its URL, title, and publish date. This allows the Knowledge Base to store and associate relations with specific sources.

6- get wikipedia data(self, candidate entity): This method attempts to retrieve data from Wikipedia for a given entity (candidate_entity) and returns a dictionary containing the title, URL, and summary of the Wikipedia page for that entity. If the entity is not found on Wikipedia, it returns 'None'.

7- from url to kb(url): This function takes a URL of a web article as input, retrieves the article's content using the 'newspaper' library, and then generates a Knowledge Base for the article using the

'from_text_to_kb' function. It includes the article's title and publish date as part of the source information in the Knowledge Base.

Block Diagram Explanation:-

1- Multimodal Input Sources:

- This represents the sources from which the system gathers multimodal inputs, such as a list of URLs from the open web or an internal SharePoint site. These inputs may contain text data from various documents or web articles.

2- Text Preprocessing:

- The text data obtained from the input sources undergoes preprocessing to clean and format the text. Preprocessing may include tasks such as removing special characters, handling punctuation, lowercasing, and other text-specific processing steps. The goal is to prepare the text data for further analysis.

3- Language Model (LM):

- The preprocessed text is fed into a Language Model, which is a powerful natural language processing (NLP) model capable of understanding and generating human-like text. In this system, the LM is responsible for generating relations between entities mentioned in the text.

4- Relation Extraction Module:

- The Language Model outputs predictions, including relations between entities, in the form of generated text. The Relation Extraction Module processes this generated text to extract meaningful relations, entities, and their associated metadata. The relations are usually structured as "head" (subject), "type" (relation type), "tail" (object), and additional metadata such as the article URL or span boundaries.

5- Knowledge Base:

- The extracted relations and associated metadata are stored in a Knowledge Base (KB) in the form of a Knowledge Graph or Knowledge Base. The KB serves as a structured repository of knowledge, representing entities as nodes and relations as edges in the graph. Each entity may have various attributes, and the graph structure captures the relationships between entities.

The system's overall goal is to create a knowledge graph or ontology that represents the semantic relationships between entities found in the input text. The Knowledge Base becomes a valuable resource for knowledge retrieval and reasoning, supporting various applications, including information retrieval, question-answering systems, and knowledge-driven decision-making processes.