Using Semantic Networks for Text Classification in Education:

“Generating Tailored Questions for Students"

Badr Touis[1], Souhaib Aammou[1], Oussama EL Warraki[1] and Jalal Lahiassi[1]

[1] Abdelmalek Essaadi University, S2IPU, Morocco
badrtouis@gmail.com

Abstract. This paper discusses how semantic revolution can be used to represent textual data for text classification purposes. Text classification includes automatically classifying text data into predefined classes or categories, such as positive or negative sentiment, or articles categorized into different topics. The paper describes the method of using semantic networks to classify knowledge, including steps such as collecting and preprocessing text data sets, representing text data in the form of semantic networks, and training algorithms. Machine learning on a semantic network, which uses algorithms to classify new textual data and generate questions based on categorical output. The article also includes Python examples for some of the steps involved in the methodology. The paper highlights the power of semantic networks as a tool for knowledge representation and manipulation in AI and related fields.

Keywords: Semantic networks, Text classification, knowledge representation, Machine learning (ML)

1 Introduction

Semantic networks are one of the most commonly used approaches in knowledge representation. They are graphical representations of knowledge that use nodes to represent concepts or objects, and edges to represent the relationships between them. The edges can be labeled with predicates or verbs to indicate the type of relationship between the nodes.

The main objective of this paper is about using semantic networks in text classification tasks. Semantic networks are a type of knowledge representation that captures relationships between concepts in a graph-like structure. Text classification, on the other hand, is a subfield of natural language processing (NLP) that involves automatically categorizing text documents into predefined classes or categories. The methodology of using semantic networks in text classification tasks involves collecting and preprocessing a dataset of text data, representing the text data as a semantic network, training a ML algorithm on the semantic network, using the trained algorithm to classify new text data, and using the text classification output to generate a set of questions.

M. Khaldi et al. (eds.), *Proceedings of the International Conference on E-Learning and Smart Engineering Systems (ELSES 2023)*, Atlantis Highlights in Social Sciences, Education and Humanities 14,
https://doi.org/10.2991/978-94-6463-360-3_5

2 Theoretical framework

2.1 Semantic network

A semantic network is a type of knowledge representation that captures relationships between concepts in a graph-like structure. In a semantic network, nodes represent concepts and edges represent relationships between these concepts. These relationships can include "is-one" relationships, such as "cat is an animal" or "part of" relationships, such as "tail is part of" cat". Semantic networks can be used to represent many different areas of knowledge, from simple taxonomies to complex networks of interrelated concepts [1].

Semantic networks have been used in many areas of artificial intelligence and related fields, such as natural language processing, expert systems, and cognitive psychology [2]. They provide a powerful and flexible way to represent knowledge, allowing easy manipulation and inference. In addition, they can be used to aid natural language understanding by providing a structured way of expressing the meaning of words and concepts [3].

Some common examples of semantic networks include WordNet, a lexical database of English words and their semantic relationships [4], and Cyc, a large-scale general knowledge database including millions of concepts and relationships. Semantic networks continue to be an active area of research in artificial intelligence, as researchers discover new ways to represent and manipulate knowledge in more powerful and expressive ways.

2.2 Text classification

Text classification is a subfield of natural language processing (NLP) that involves automatically categorizing text documents into predefined classes or categories [5]. It is a supervised learning task that requires a labeled dataset to train a model, which can then classify new, unlabeled text data into the predetermined categories.

Several machine learning algorithms can be used for text classification, such as Naive Bayes, decision trees, and support vector machines. More recently, deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have also shown promising results in text classification tasks [6].

3 Methodology

Semantic networks are indeed a powerful tool for knowledge representation, and they can be particularly useful in text classification tasks:

1. Collect and preprocess a dataset of text data: This step involves gathering a dataset of text data that is related to the subject matter you want to classify. This can include textbooks, articles, lecture notes, and any other relevant sources. It is important to preprocess the data to remove any irrelevant information and standardize the format of the text.

2. Represent the text data as a semantic network: In this step, we will create a semantic network that represents the relationships between the different concepts in the text data. The nodes in the network represent the concepts, and the edges represent the relationships between them.
3. Train a ML algorithm on the semantic network: This step involves training a ML algorithm, such as a neural network, on the semantic network representation of the text data. The algorithm will learn to recognize patterns in the relationships between concepts and use this knowledge to classify new text data.
4. Use the trained algorithm to classify new text data: Once the algorithm has been trained, it can be used to classify new text data into different levels of difficulty, such as easy, medium, or hard. This can help teachers and educators to identify areas where students are struggling and tailor their teaching accordingly.
5. Use the text classification output to generate a set of questions: Finally, the text classification output can be used to generate a set of questions that are tailored to the students' level of understanding of the subject matter. This can help to reinforce their learning and improve their overall comprehension of the material.

Essentially, semantic networks provide a powerful way to represent knowledge in a structured and meaningful way, and they can be used to improve the accuracy and effectiveness of text classification tasks.

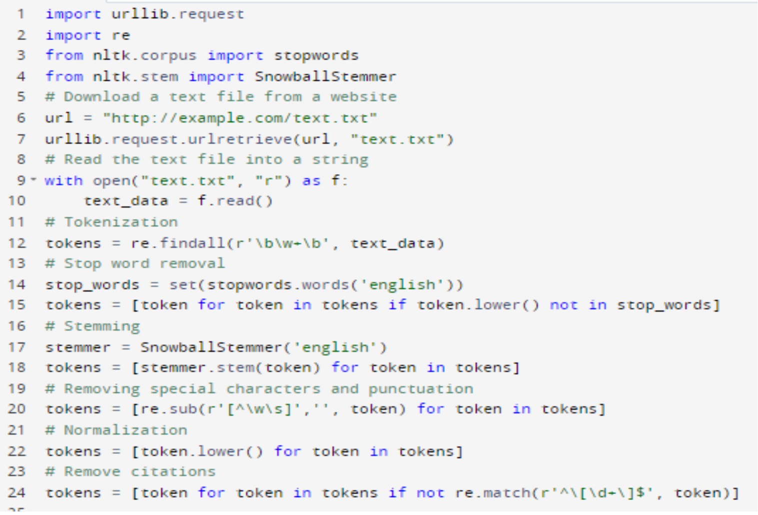
4 Implementation

4.1 Collect and preprocess a dataset

Collecting and preprocessing a dataset of text data is an important first step in using semantic networks for text classification (figure 1):

1. Gather a dataset of text data: The first step is to gather a dataset of text data that is related to the subject matter you want to classify. This dataset can come from a variety of sources, including textbooks, articles, lecture notes, and any other relevant sources. It is important to ensure that the dataset is representative of the subject matter and covers a broad range of topics and concepts.
2. Preprocess the data: Once you have collected the text data, the next step is to preprocess it. This involves several steps, including:
   a. Tokenization: The text data is broken down into individual words, phrases, or sentences, depending on the level of granularity required for the classification task.
   b. Stop word removal: Common words such as "the," "and," and "a" are removed from the text data as they do not provide meaningful information for the classification task.
   c. Stemming or lemmatization: Words are transformed into their base forms to standardize the text data and reduce the number of unique words in the dataset.

   d. Removing special characters and punctuation: Special characters and punctuation marks are removed from the text data as they do not provide any meaningful information for the classification task.
   e. Normalization: The text data is standardized to ensure that it is in a consistent format, such as lowercase or uppercase, to reduce the number of unique words in the dataset.
3. Remove irrelevant information: Finally, any irrelevant information is removed from the text data. This can include advertisements, footnotes, citations, and other information that is not relevant to the classification task.



```python
import urllib.request
import re
from nltk.corpus import stopwords
from nltk.stem import SnowballStemmer
# Download a text file from a website
url = "http://example.com/text.txt"
urllib.request.urlretrieve(url, "text.txt")
# Read the text file into a string
with open("text.txt", "r") as f:
    text_data = f.read()
# Tokenization
tokens = re.findall(r'\b\w+\b', text_data)
# Stop word removal
stop_words = set(stopwords.words('english'))
tokens = [token for token in tokens if token.lower() not in stop_words]
# Stemming
stemmer = SnowballStemmer('english')
tokens = [stemmer.stem(token) for token in tokens]
# Removing special characters and punctuation
tokens = [re.sub(r'[^\w\s]','', token) for token in tokens]
# Normalization
tokens = [token.lower() for token in tokens]
# Remove citations
tokens = [token for token in tokens if not re.match(r'^\[\d+\]$', token)]
```

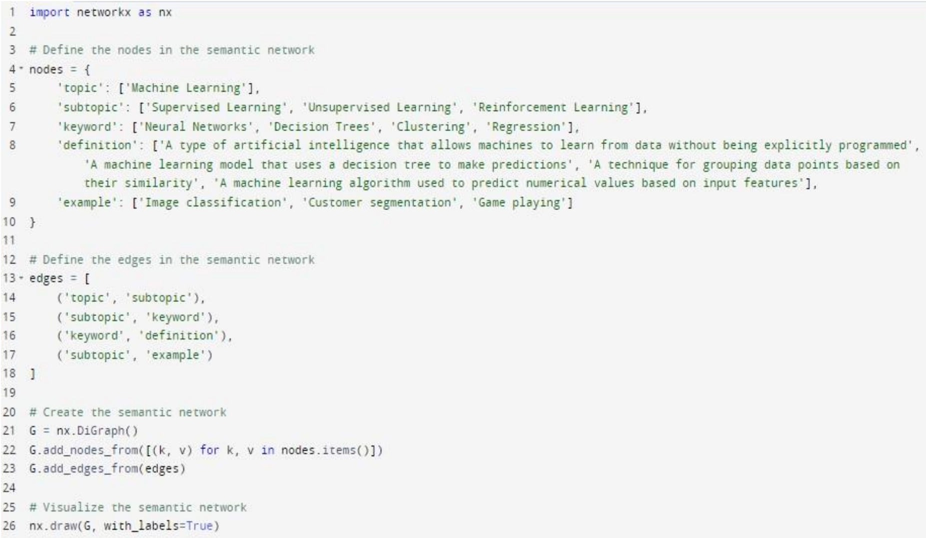
Fig. 1. Python code for Collecting and preprocessing a dataset

By collecting and preprocessing a dataset of text data, we can ensure that the input data is standardized, free of irrelevant information, and ready for use in the semantic network representation and ML algorithms used in the text classification task.

4.2 Represent the text data as a semantic network:

To create a semantic network for text data, you first need to identify the key concepts and topics that are relevant to your text classification task. These concepts might include things like "topic," "subtopic," "keyword," "definition," and "example," among others.

We represent text data as a semantic network in Python using the NetworkX library (see figure 2):



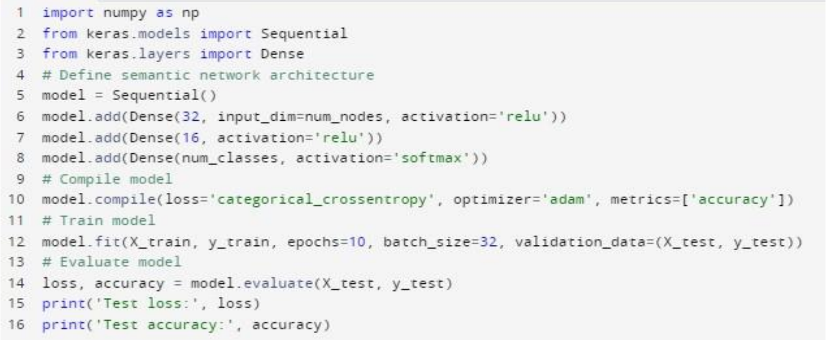
```python
import networkx as nx

# Define the nodes in the semantic network
nodes = {
    'topic': ['Machine Learning'],
    'subtopic': ['Supervised Learning', 'Unsupervised Learning', 'Reinforcement Learning'],
    'keyword': ['Neural Networks', 'Decision Trees', 'Clustering', 'Regression'],
    'definition': ['A type of artificial intelligence that allows machines to learn from data without being explicitly programmed',
        'A machine learning model that uses a decision tree to make predictions', 'A technique for grouping data points based on
        their similarity', 'A machine learning algorithm used to predict numerical values based on input features'],
    'example': ['Image classification', 'Customer segmentation', 'Game playing']
}

# Define the edges in the semantic network
edges = [
    ('topic', 'subtopic'),
    ('subtopic', 'keyword'),
    ('keyword', 'definition'),
    ('subtopic', 'example')
]

# Create the semantic network
G = nx.DiGraph()
G.add_nodes_from([(k, v) for k, v in nodes.items()])
G.add_edges_from(edges)

# Visualize the semantic network
nx.draw(G, with_labels=True)
```

Fig. 2. Python code for representing the text data as a semantic network

We create a directed graph using the DiGraph() function from NetworkX, and add the nodes and edges to it using the add_nodes_from() and add_edges_from() functions. Finally, we visualize the semantic network using the draw() function from NetworkX.

4.3 Train a machine learning algorithm on the semantic network

Once we have prepared the data, we can use a variety of machine learning algorithms to train the model. One popular approach for text classification is to use a neural network, such as a convolutional neural network (CNN) or a recurrent neural network (RNN). These algorithms are designed to handle sequential data, such as text, and can learn to recognize patterns in the semantic network representation.



```python
import numpy as np
from keras.models import Sequential
from keras.layers import Dense
# Define semantic network architecture
model = Sequential()
model.add(Dense(32, input_dim=num_nodes, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))
# Compile model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
# Train model
model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test, y_test))
# Evaluate model
loss, accuracy = model.evaluate(X_test, y_test)
print('Test loss:', loss)
print('Test accuracy:', accuracy)
```

Fig. 3. Python code to train a machine learning algorithm on the semantic network

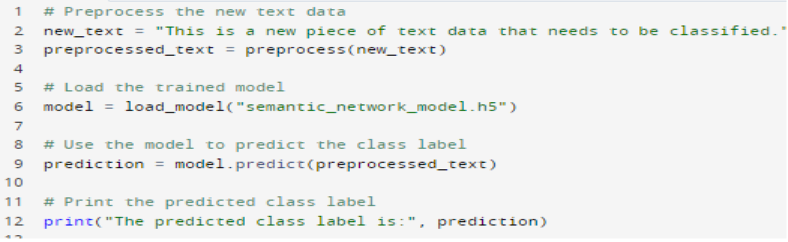
In this code, the Sequential model is used to define the semantic network architecture with several Dense layers. The input_dim of the first layer is set to the number of nodes in the semantic network, and the activation functions are set to 'relu' for hidden layers and 'softmax' for the output layer, which is used for classification.

The model is compiled with the categorical_crossentropy loss function, the adam optimizer, and the 'accuracy' metric for evaluation. It is then trained on the preprocessed dataset using the fit method with a batch size of 32 and for 10 epochs.

After training, the model is evaluated on the test set, and the loss and accuracy are printed. Finally, the trained model can be used to classify new text data.

4.4 Use the trained algorithm to classify new text data

To use the trained algorithm to classify new text data, first we need to preprocess the new text data in the same way you preprocessed the training data. Then, we can input the preprocessed data into the trained model and use it to predict the class label.

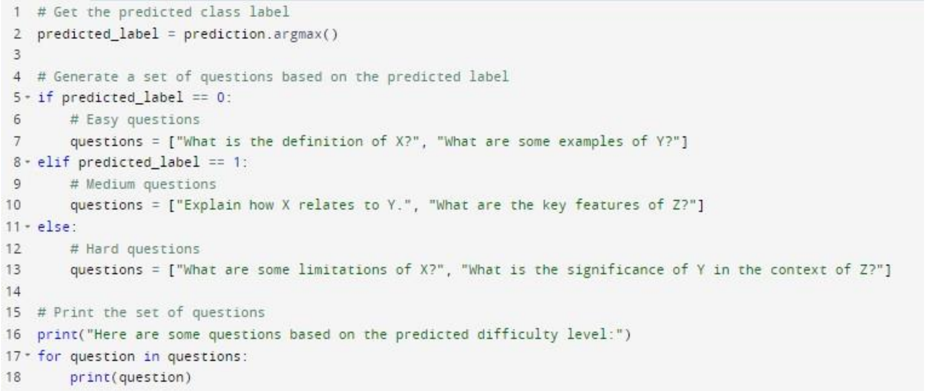


```
# Preprocess the new text data
new_text = "This is a new piece of text data that needs to be classified."
preprocessed_text = preprocess(new_text)

# Load the trained model
model = load_model("semantic_network_model.h5")

# Use the model to predict the class label
prediction = model.predict(preprocessed_text)

# Print the predicted class label
print("The predicted class label is:", prediction)
```

Fig. 4. Use a trained model to classify new text data in Python

4.5 Use the text classification output to generate a set of questions

To generate a set of questions based on the text classification output, we can use the predicted class label to determine the level of difficulty of the text. For example, if the predicted class label is "easy," we can generate a set of questions that test basic knowledge and understanding of the subject matter. If the predicted class label is "hard," then we can generate a set of questions that test more advanced concepts and require deeper understanding of the material.



```
# Get the predicted class label
predicted_label = prediction.argmax()

# Generate a set of questions based on the predicted label
if predicted_label == 0:
    # Easy questions
    questions = ["What is the definition of X?", "What are some examples of Y?"]
elif predicted_label == 1:
    # Medium questions
    questions = ["Explain how X relates to Y.", "What are the key features of Z?"]
else:
    # Hard questions
    questions = ["What are some limitations of X?", "What is the significance of Y in the context of Z?"]

# Print the set of questions
print("Here are some questions based on the predicted difficulty level:")
for question in questions:
    print(question)
```

Fig. 5. Text classification output to generate a set of questions in Python

In this code, the argmax method is used to get the index of the highest predicted value in the prediction array, which corresponds to the predicted class label. Then, based on the predicted label, a set of questions is generated and printed to the console.

5 Conclusion

In conclusion, semantic networks offer a powerful and flexible tool for representing knowledge in a structured way, and text classification is a valuable application of this technique. By using semantic networks to represent the relationships between concepts in text data, ML algorithms can be trained to classify new text data into predetermined categories. This approach has many potential uses in education and research, including identifying areas where students are struggling and generating tailored questions to reinforce learning. With further research and development, semantic networks and text classification have the potential to revolutionize the way we represent and understand knowledge.

References

1. Bonatti, P. A., Decker, S., Polleres, A., & Presutti, V. (2019). Knowledge graphs: New directions for knowledge representation on the semantic web (dagstuhl seminar 18371). In *Dagstuhl reports* (Vol. 8, No. 9). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
2. Vijayakumar, S., & Sheshadri, K. N. (2019). Applications of artificial intelligence in academic libraries. *International Journal of Computer Sciences and Engineering*, *7*(1), 136-140.
3. Kang, Y., Cai, Z., Tan, C. W., Huang, Q., & Liu, H. (2020). Natural language processing (NLP) in management research: A literature review. *Journal of Management Analytics*, *7*(2), 139-172.
4. Sarica, S., Han, J., & Luo, J. (2023). Design representation as semantic networks. *Computers in Industry*, *144*, 103791.
5. Basha, S. M., & Fathima, A. S. (2023). *Natural Language Processing: Practical Approach*. MileStone Research Publications.
6. Zhong, B., Xing, X., Love, P., Wang, X., & Luo, H. (2019). Convolutional neural network: Deep learning-based classification of building quality problems. *Advanced Engineering Informatics*, *40*, 46-57.