The Generation and Visualization of Solutions for Improving Yokohama City's Government and Citizens' Living Quality by Machine Learning Methods and LLM

氏名:薛丁銘

所属:東京大学　工学系研究科　システム創成学専攻

学籍番号:37-237256

1. Introduction

In today's society, the abundance of social data has reached a point where individuals often struggle to discern the relationships and essence among the data when addressing complex societal issues. This lack of clarity impedes the generation of effective solutions. There is a pressing need for a methodology that can unveil the connections between data points, thereby facilitating the identification of viable solutions to these problems.

In this topic, a network was created and visualized for demonstrating the solutions for improving Yokohama City's government and citizen's living quality. The datasets of Yokohama City regarding to social issues were collected and a discussion meeting was held to summarize the potential problems of Yokohama City's problem and expectations from the citizens. Through machine learning methods such as clustering and similarity calculation of sentence vectors along with the meaning extraction by LLM (large language model), the latent relationship between datasets of Yokohama is revealed and visualized. Some experiments were also conducted to select the most suitable clustering model for the system.

The work of this topic mainly lies on the selection of machine learning models in order to better extract the essence of the meta data of the datasets in order to creating a high quality solution network. Besides, with the help from LLM, the essence extraction of sentences can be smooth and the latent relationship between datasets can be better revealed.

2. Background

2.1 Sentence Vectors Clustering

Sentence vector clustering is a pivotal task in natural language processing that involves grouping sentences based on their semantic similarities. Various methods have been developed to achieve effective sentence vector clustering, each employing distinctive techniques. Traditional approaches often rely on methods like K-means clustering or hierarchical clustering [1]. However, recent advancements in deep learning have led to the emergence of more sophisticated techniques, such as using pre-trained language models to generate sentence embeddings [2].

One prevalent method employs models like BERT (Bidirectional Encoder Representations from Transformers) [3] or Universal Sentence Encoder [4] to convert sentences into dense vector representations, capturing semantic nuances. These embeddings are then fed into clustering algorithms, like K-means or DBSCAN, to group similar sentences together. Alternatively, methods like Doc2Vec [5] leverage paragraph vectors to represent sentences in a continuous vector space, facilitating meaningful clustering.

In recent research, attention mechanisms have been incorporated into neural network architectures to enhance the performance of sentence vector clustering. Models like Transformer-based architectures, which utilize self-attention mechanisms, have demonstrated superior capabilities in capturing contextual information for improved clustering accuracy [6].

These advancements showcase the evolving landscape of sentence vector clustering, combining classical methods with state-of-the-art deep learning approaches, aiming to uncover nuanced relationships within text data. As the field continues to progress, the integration of innovative techniques promises further improvements in the accuracy and efficiency of sentence vector clustering.

2.2 Large Lange Model for Essence Extraction

Large language models have become instrumental in distilling the essence of textual content, revolutionizing natural language processing and understanding. These models, such as GPT-3 (Generative Pre-trained Transformer 3), have demonstrated remarkable capabilities in summarizing and capturing the central themes of diverse pieces of text. With their deep learning architectures and vast pre-training on extensive datasets, these models excel in contextual understanding and generating coherent summaries.

One notable application of large language models in summarization involves extractive summarization techniques. GPT-3, for instance, has showcased its ability to identify key sentences or phrases within a document and generate concise summaries that encapsulate the essential information [7]. The model's proficiency in comprehending context and extracting salient points has paved the way for more advanced and accurate summarization approaches.

Furthermore, large language models have been employed in abstractive summarization, where they not only extract information but also generate new, coherent sentences to convey the main ideas of a text. GPT-3, with its extensive pre-training, has demonstrated proficiency in this regard, producing human-like summaries that go beyond mere extraction.

Dingming Xue, The University of Tokyo, Graduate School of Engineering, Department of System Innovation

These advancements underscore the transformative impact of large language models in distilling the central themes of text, offering a glimpse into the potential of such models in enhancing the efficiency and effectiveness of automated summarization tasks.

2.3 Feature Concept & Data Leaves

The idea of feature concept and data leaves was firstly proposed by Prof.Yukio Ohsawa from The University of Tokyo. The Feature Collection (FC) serves as a conceptualization of the anticipated knowledge to be gained from the dataset(s). It can be manifested in diverse forms such as text, images, or logical trees [4], aligning with the specified requirements. Deep Learning (DL) acts as metadata encapsulating an underlying event structure within the dataset, corresponding to the FC for an individual dataset. The DLs are chosen and gathered from a pre-established DL base, and they are linked to the FC. Encompassing the FC through DLs involves envisioning the utilization of data to fulfill a particular requirement. This process highlights the connection between the FC, which represents the expected knowledge, and the DLs, which encapsulate the latent event structure within the dataset.

2.4 Meta Data

Metadata, in the context of information systems, refers to structured and descriptive information that provides insights into the characteristics, attributes, or properties of data. It serves as a critical component for organizing, managing, and understanding diverse datasets. Metadata encapsulates essential details about the origin, structure, format, and context of data, offering a comprehensive framework for data management and interpretation.

There are various types of metadata, each serving distinct purposes. Descriptive metadata focuses on providing information about the content, including titles, keywords, and abstracts, facilitating the discovery and retrieval of specific data. Structural metadata outlines the organization and relationships within a dataset, defining how different elements are interconnected. Administrative metadata encompasses details related to data ownership, access permissions, and versioning, contributing to effective data governance.

The role of metadata extends beyond individual datasets to support interoperability across systems. Standardized metadata formats and taxonomies enhance data sharing and integration, fostering collaboration within diverse domains. In the era of big data and artificial intelligence, metadata becomes increasingly crucial for ensuring data quality, provenance, and compliance with regulatory standards.

In summary, metadata acts as a foundational element in the realm of data management, offering a structured framework that enhances data discoverability, usability, and reliability across various applications and disciplines. Its importance continues to grow as data ecosystems become more complex and interconnected.

3. Methodologies and Experiments

The whole works is divided into three parts. The first part was data preprocessing, including data acquiring, data transformation and sentence vectors embedding for deep learning. The second part was model selection for better clustering the generated sentence vectors. The final part was to sort the sentences according to their similarity and visualize the solution network.

3.1 Data Preprocessing

This paragraph mainly talks about the procedure of data preprocessing. After importing necessary packages such as transformers and pytorch, the global variables were announced such as the prompt for LLM, the API keys, the configuration of generation methods of LLMS and the settings of the similarity calculations as shown in Figure 3.1.1.

```
# API KKey Settings
apis = pd.read_csv("D:\\code\\FCDL\\apis.csv")
openai.api_key = apis["openai"][0]
API_KEY = apis['googlesearch'][0]
CUSTOM_SEARCH_ENGINE = 'a76a2783d10a44be6'
page_limit = 1
PROMPT_TEXT_KADAI_CORP = '''
    この文章の主題を15文字以上30文字未満で要約してください。
'''

PROMPT_TEXT_FC = '''
    この文章の主題を要約すると次の5個です。それぞれ15文字以上30文字未満で要約してください。
'''

PROMPT_TEXT_OTR = '''
    この文章の主題を要約すると次の3個です。それぞれ15文字以上30文字未満で要約してください。
'''

model = SentenceTransformer('stsb-xlm-r-multilingual')

OUTPUT_TOKENS = 1000

MAX_DISTANCE = 3.0
EMPTY_TEXT = ''
SIMILARITY_UPPER_THRESHOLD = 3.00
SIMILARITY_LOWER_THRESHOLD = 2.65
RESCALED_MAX_SIMILARITY_VALUE = 3.0
RESCALED_MIN_SIMILARITY_VALUE = 0.1

distance_fc_list = []
distance_otr_list = []
distance_fc_and_otr_list = []
```

Figure 3.1.1 Global Variables and Settings

After that, similar code was used to extract, load and transform the city problem data, the dataset of the companies in Yokohama City, the information dataset of government grants in Yokohama City and other previous collected datasets of public meta data. Regarding to the city problem, in a previous meeting in the University of Tokyo, the guests were gathered to discuss the problem of Yokohama City's problem from the various perspectives. One example code is as shown in Figure 3.1.2.

```
#########################################################################
######################## 企業データセットの取得 ########################
#########################################################################

corp_df = pd.read_csv("D:\\code\\FCDL\\dataleaves\\yokohama_chiikikouken.csv")
corp_df["fulltext"] = [f"{pr}, {jg}, {tk}" for pr, jg, tk in zip(corp_df["PR"], corp
corp_df["fulltext"] = [st.replace("nan, ", "") for st in corp_df["fulltext"]]
corp_df["fulltext"] = [st.replace(", nan", "") for st in corp_df["fulltext"]]

sentences_corp = []
for add_s in corp_df["fulltext"]:
    sentences_corp.append(add_s.strip())

sentences_corp_df = pd.DataFrame({
    "sentence": sentences_corp
})
sentences_corp_df["counterpart"] = [st.strip().replace("株式会社", "").replace("ホー

#Compute embeddings
if False:
    print("Start calc embeddings")
    print(datetime.datetime.now())
    embeddings_corp = model.encode(sentences_corp, convert_to_tensor=True)
    print(datetime.datetime.now())

    # pickle化してファイルに書き込み
    with open('D:\\code\\FCDL\\dataleaves\\dataleaves_corp.pkl', 'wb') as f:
        pickle.dump(embeddings_corp, f)
else:
    with open('D:\\code\\FCDL\\dataleaves\\dataleaves_corp.pkl', 'rb') as f:
        embeddings_corp = pickle.load(f)
print(len(embeddings_corp))
print(len(sentences_corp_df))
print(sentences_corp_df.tail())
```

Figure 3.1.2 Dataset Transformation

This part reads the data from the format of csv file and transform the dataset into a suitable structure. Besides, it also embeds the essence of the data to sentence vectors using 'stsb-xlm-r-multilingual' model, which is commonly used for embedding work in NLP field.

While dealing with the query dataset in the meeting, the result of the answers sometime could be very long, hard to embed and hard

to be demonstrated through the network. In that case, ChatGPT was used for extracting the essence of the answers within 30 words. The code of using OpenAI API is demonstrated as in Figure 3.1.3.

```
def get_results_for_one_text(in_text, prompt):
    response = openai.Completion.create(
        # model="gpt-4",
        # model="gpt-3.5-turbo",
        # model="text-davinci-003",
        model="gpt-3.5-turbo-instruct",
        prompt=_get_cleaned_text(in_text) + prompt,
        temperature=0,  #temperature=0.5,
        max_tokens=OUTPUT_TOKENS,
        top_p=1.0,
        frequency_penalty=0.8,
        presence_penalty=0.0
    )
    return _get_numbering_removed_keyword(
        _get_parsed_result_by_return(
            response["choices"][0]["text"]
        ))
```

Figure 3.1.3 The Code of Using OpenAI API

There are two kind of request for OpenAI APIs regarding to the endpoints you would like to request. In other words, it depends on whether the user would like to inquire in the formation of a chat or just simple texture. Figure 3.1.4 shows the two different APIs from the OpenAI website and for each kind of API there's some recommended models.

Model endpoint compatibility

ENDPOINT	LATEST MODELS
/v1/assistants	All models except gpt-3.5-turbo-0301 supported. retrieval tool requires gpt-4-1106-preview or gpt-3.5-turbo-1106.
/v1/audio/transcriptions	whisper-1
/v1/audio/translations	whisper-1
/v1/audio/speech	tts-1, tts-1-hd
/v1/chat/completions	gpt-4 and dated model releases, gpt-4-1106-preview, gpt-4-vision-preview, gpt-4-32k and dated model releases, gpt-3.5-turbo and dated model releases, gpt-3.5-turbo-16k and dated model releases, fine-tuned versions of gpt-3.5-turbo
/v1/completions (Legacy)	gpt-3.5-turbo-instruct, babbage-002, davinci-002
/v1/embeddings	text-embedding-ada-002
/v1/fine_tuning/jobs	gpt-3.5-turbo, babbage-002, davinci-002
/v1/moderations	text-moderation-stable, text-moderation-latest
/v1/images/generations	dall-e-2, dall-e-3

Figure 3.1.4 GPT Model Selection

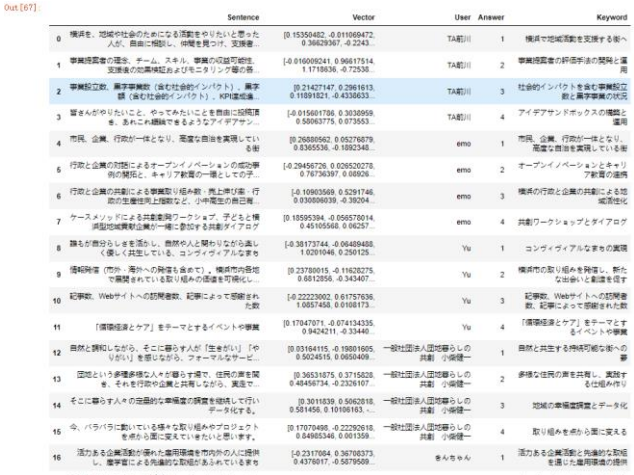
However, a lot of GPT models were deprecated from 2024-01-04, the 'text-davinci-003' model is on longer be supported and I have to switch the model to 'gpt-3.5-turbo-instruct' so that I could complete this work as shown in Figure 3.1.5.

InstructGPT models

SHUTDOWN DATE	LEGACY MODEL	LEGACY MODEL PRICE	RECOMMENDED REPLACEMENT
2024-01-04	text-ada-001	$0.0004 / 1K tokens	gpt-3.5-turbo-instruct
2024-01-04	text-babbage-001	$0.0005 / 1K tokens	gpt-3.5-turbo-instruct
2024-01-04	text-curie-001	$0.0020 / 1K tokens	gpt-3.5-turbo-instruct
2024-01-04	text-davinci-001	$0.0200 / 1K tokens	gpt-3.5-turbo-instruct
2024-01-04	text-davinci-002	$0.0200 / 1K tokens	gpt-3.5-turbo-instruct
2024-01-04	text-davinci-003	$0.0200 / 1K tokens	gpt-3.5-turbo-instruct

Figure 3.1.5 Model Deprecations

After the data preprocessing, the dataframe consisting the answer , it's essence and embedding vectors is generated as shown in Figure 3.1.5 for clustering and similarity calculations.



	Sentence	Vector	User	Answer	Keyword
0	横浜を、地域や社会のためになる活動をやりたいと思った人が、自由に相談し、仲間を見つけ、支援者...	[0.15350482, -0.011069472, 0.36629367, -0.2243...	TA前川	1	横浜で地域活動を支援する街へ
1	事業提案者の理念、チーム、スキル、事業の収益可能性、支援後の効果検証およびモニタリング等の各...	[-0.016009241, 0.96617514, 1.1718836, -0.72530...	TA前川	2	事業提案者の評価手法の開発と運用
2	事業設立数、雇字事業数（含む社会的インパクト）、雇字額（含む社会的インパクト）、KPI達成進...	[0.21427147, 0.2961613, 0.11891821, -0.4338633...	TA前川	3	社会的インパクトを含む事業設立数と雇字事業の状況
3	皆さんがやりたいこと、やってみたいことを自由に話相談を、あれこれ話題できるようなアイデアサン...	[-0.015601786, 0.3038959, 0.58063775, 0.073553...	TA前川	4	アイデアサンドボックスの構築と運用
4	市民、企業、行政が一体となり、豊富な自治を実現している街	[0.26880562, 0.05276879, 0.8365536, -0.1892346...	emo	1	市民、企業、行政が一体となり、豊富な自治を実現している街
5	行政と企業の対話によるオープンイノベーションの成功事例の開拓と、キャリア教育の一環としての子...	[-0.29456726, 0.026520278, 0.76736397, 0.06826...	emo	2	オープンイノベーションとキャリア教育の連携
6	行政と企業の共創による事業取り組み数・売上伸び率・行政の生産性向上指数など、小中豊生の自己肯...	[-0.10903569, 0.5291746, 0.030899039, -0.39204...	emo	3	横浜の行政と企業の共創による地域活性化
7	ケースメソッドによる共創型ワークショップ、子どもと横浜地域資源企業が一緒に参加する共創ダイアログ	[0.18595394, -0.056576014, 0.45105568, 0.06257...	emo	4	共創ワークショップとダイアログ
8	誰もが自分らしさを活かし、自然や人と関わりながら楽しく優しく共生している、コンヴィヴィアルなまち	[-0.38173744, -0.06489488, 1.0201046, 0.250125...	Yu	1	コンヴィヴィアルなまちの実現
9	情報発信（市外・海外への発信を含めて）、横浜市内各地で展開されている取り組みの価値を可視化し...	[0.23780015, -0.11628275, 0.6812856, -0.343407...	Yu	2	横浜市の取り組みを発信し、新たな出会いと創造を促す
10	記事数、Webサイトへの訪問者数、記事によって感銘された数	[-0.22223002, 0.61757636, 1.0857458, 0.0108173...	Yu	3	記事数、Webサイトへの訪問者数、記事によって感銘された数
11	「循環経済とケア」をテーマとするイベントや事業	[0.17047071, -0.074134335, 0.9424211, -0.33440...	Yu	4	「循環経済とケア」をテーマとするイベントや事業
12	自然と調和しながら、そこに暮らす人が「生きがい」「やりがい」を感じながら、フォーマルなサービ...	[0.03164115, -0.19801605, 0.5024515, 0.0650409...	一般社団法人団地暮らしの共創 小俣健一	1	自然と共生する持続可能な街への歩
13	団地という多様多様な人々が暮らす場で、住民の声を聞き、それを行政や企業と共有しながら、実証で...	[0.36531875, 0.3715828, 0.48456734, -0.2326107...	一般社団法人団地暮らしの共創 小俣健一	2	多様な住民の声を共有し、実践する仕組みづくり
14	そこに暮らす人々の生活の along 幸福度の調査を継続して行い、データ化する。	[0.3011839, 0.5062818, 0.581456, 0.10106163, ...	一般社団法人団地暮らしの共創 小俣健一	3	地域の幸福度調査とデータ化
15	今、バラバラに動いている様々な取り組みやプロジェクトを求める声に変えていきたいと思います。	[0.17070498, -0.22292618, 0.04965346, 0.001359...	一般社団法人団地暮らしの共創 小俣健一	4	取り組みを求める声に変える
16	活力ある企業活動が優れた雇用環境を市内外の人に提供し、雇字による先進的な取組があふれているまち	[-0.2317684, 0.36708373, 0.4376017, -0.5879509...	きんちゃん	1	活力ある企業活動と先進的な取組を通じた雇用環境の提供

Figure 3.1.5 Processed Answer Data Frame

3.2 Sentence Vectors Clustering

As it is introduced in the background chapter, the most popular method of sentence vectors clustering is K-means. Therefore, k-means is the first machine learning model to use in this case. Another clustering method is Agglomerative Clustering model. The comparison of the two clustering results is as shown in Figure 3.2.1.

```
############ Displaying the statistical result of the Kmeans_clusteringwith preprocess method of normalization ############
##
The Silhousette Score:  0.05026147047133942
The Calinski-Harabasz Score:  3.4766760003185
The Davies-Bouldin Score:  2.6418963298313964
############ Displaying the statistical result of the Agglomerative_clusteringwith preprocess method of normalization ####
#########
The Silhousette Score:  0.056583986956275647
The Calinski-Harabasz Score:  3.577797179762023
The Davies-Bouldin Score:  2.6265577379925316
```

Figure 3.2.1 The Comparison of Two Clustering methods

From the figure it can be seen that the Silhousette Score of Agglomerative Clustering model is a slight bit higher, the Calinski-Harabasz Score is bit higher and the Davies-Bouldin Score is a slight bit lower than the result of k-means, indicating that Agglomerative Clustering slightly out-performed K-means in this issue. However, both of the models have very low Silhousette Score nearly to zero, which implies that the clustering result is not good.

The above methods used regular normalization methods to do the clustering of the sentence vectors. There's another way of vector preprocess, which is calculating the cosine matrix of all sentence vectors. The comparison is as shown in Figure 3.2.2

```
############ Displaying the statistical result of the Kmeans_clusteringwith preprocess method of cosine ############
The Silhousette Score:  0.04791624169589978
The Calinski-Harabasz Score:  3.574451415513821
The Davies-Bouldin Score:  2.736412291821706
############ Displaying the statistical result of the Agglomerative_clusteringwith preprocess method of cosine ##########
##
The Silhousette Score:  0.07201064095054989
The Calinski-Harabasz Score:  3.2334275592221 41
The Davies-Bouldin Score:  2.6706460217822654
```

Figure 3.2.1 The Comparison of Two Clustering methods with Cosine Matrix

From this result we can find out that with cosine matrix as input, the clustering result has bad performance for K-means. However, for the Agglomerative Clustering model the performance was better. Considering that although the performance was improved, the Silhousette Score is still near to zero, I finally decided to use k-means and ordinary normalization for clustering.

3.3 Visulation

After the calculation of similarity and rankings between vectors, the most similar vectors were extracted and used for generating the solution network. Each clustered topic will be shown as a node and it would be connected to the related companies/grants and other potential useful datasets. The whole image of the network generation is as shown in Figure 3.3.1.

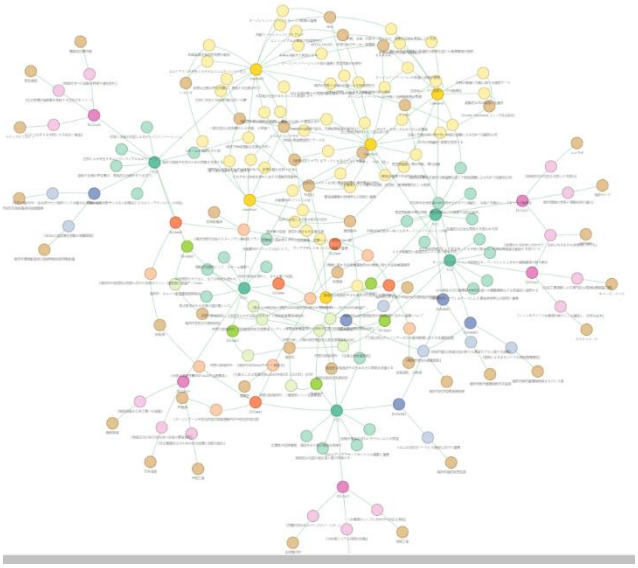


Figure 3.3.1 The Visualization Results

4. Conclusion

In this topic, in order to demonstrate the hidden relationship or solution of Yokohama City's problem, a network was created and visualized for improving Yokohama City's government and citizen's living quality. The work consists of data processing, NLP techniques such as sentence vectors clustering and using LLM for texture essence extraction. Some experiments were also conducted to select the most suitable clustering model for the system. For the future work, a propose of applying LLM for clustering can be used for generating better quality results.

Reference Paper

1. Jain, A. (2010). Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 31(8), 651-666.
2. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. arXiv preprint arXiv:1908.10084.
3. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
4. Cer, D., Yang, Y., Kong, S. Y., Hua, N., Limtiaco, N., John, R. S., ... & Kurzweil, R. (2018). Universal Sentence Encoder. arXiv preprint arXiv:1803.11175.
5. Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In International conference on machine learning (pp. 1188-1196).
6. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
7. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165.
8. Radford, A., & Wu, J. (2019). Language models are few-shot learners. OpenAI Blog.