Chat-GPT Based Entities Classification - ESWA 2023

Taliya Shitreet, Renana Rimon, Eran Levy, Daniel Zafrir,Or Haim Anidjar

[1]School of Computer Science, Ariel University, Israel

orhaim@ariel.ac.il, Taliyashitreet@gmail.com, renana1414@gmail.com, eranlevy9@gmail.com, danieltzafrir96@gmail.com

Abstract

renana This research is a collaborative endeavor with ProBogotá aimed at developing Bogotá savanna in Colombia. The Bogotá Savanna spans a wide expanse and is inhabited by a significant population, a substantial portion of which grapples with socio-economic challenges. The objective is to plan a public infrastructure in collaboration with the residents, utilizing the collective intelligence approach [1] to address the specific needs of the inhabitants accurately. Data collection is facilitated by a Chatbot [2] to reach a wide audience at a minimal cost. The collected data requires labeling for optimal utilization. The labeling process involves determining the relevance of each sentence to the architecture topic and assigning it to the appropriate sub-topic. The research employs two learning approaches: unsupervised [3] [4] and supervised [5] learning. In the unsupervised method,combined with metric learning [6] approach, sentences embedding are computed using Language-agnostic BERT [7], and sentence pairs are selected based on high and low cosine similarity. In the supervised approach, initial labeling of the data is required. The article proposes an innovative approach using ChatGPT for labeling. ChatGPT [8] [9] receives the unlabelled sentences along with a list of possible labels and provides multi-labeling for each sentence. Following the tagging process, two deep learning models are trained, incorporating sentences embedding. The first model utilizes Language-agnostic BERT, while the second model employs TF-IDF [10] [11]. Experimental results indicate that ChatGPT's initial sentence labeling is consistent, as the model achieves favorable outcomes across various metrics. The research contributes to improving data collection and processing with deep learning [12], [13] techniques, avoiding human intervention by utilizing ChatGPT, ensuring consistent data labeling using ChatGPT, and offering a replicable model for urban planning with global potential, Thanks to the use of Language-agnostic BERT.

1. Introduction

Eran Urban planning has become increasingly complex in recent years. The city planning process today is overseen by a group of professionals with experience in the field. However, those responsible for planning do not necessarily live in the same area or will live in the same area in the future. Urban planners are limited in their ability to fully understand the everyday needs and preferences of the individuals who will ultimately live within the planned communities.

They can only guess or rely on projects they have done in the past in similar areas. But what if it's a new area for a new population? The paper focuses on the city of Bogota, the capital city of Colombia. But in particular in small municipalities Colombia is the largest and most populated city in Colombia. Its population is estimated at 8 million people.

Bogota is surrounded by several small municipalities that have grown organically over the years without a long-term vision.

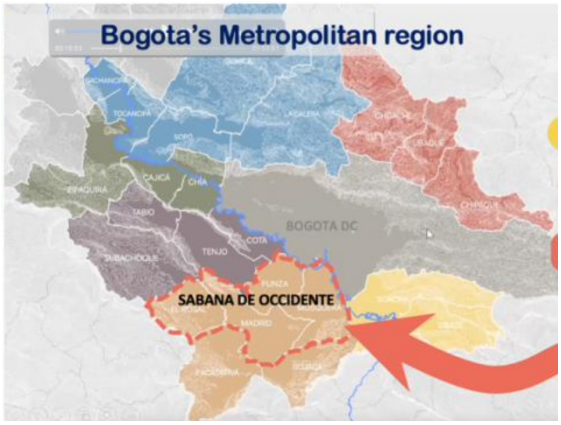

Figure 1: *Bogota's 'Sabana de Occidente planning area comprises the municipalities of Funza, Mosquera, Madrid, Facatativá and El Rosal (only part of the district).*

Taliya While the city is known for its vibrant culture and burgeoning business scene, it is also grappling with high levels of poverty, unemployment, and income inequality. A significant portion of the population lives in informal settlements, where access to basic services such as clean water, sanitation, and elec-

tricity can be limited. These disparities are further exacerbated by the limited public transportation options available to residents, which can hinder their ability to access education, healthcare, and employment opportunities.

Eran These municipalities attracted mostly populations who could not afford to live in Bogota and would commute to the capital daily. Still, the region has also developed its own industries, among them 'export-quality' flowers - the biggest crop is roses. This also means that a large percentage of the population living in the area are seasonal workers, one of the strongest explanations as to why the region did not develop a strong identity and has been so shortsighted when it comes to designing the future living conditions in the area.

Taliy ProBogotá is a non-profit organization composed of 40 different companies that donate funds to the organization every month. The organization is staffed by young, visionary planners with a high degree of motivation and a strong desire to make a difference and involve the community in the process of ultimately designing the city while representing the interests of the population and contributing to the community.

Eran The challenge of better city planning, particularly in areas heavily reliant on seasonal workers such as the region under consideration, highlights the need for public participation in order to gather diverse perspectives and harness the collective wisdom of the crowd to develop more effective strategic plans.

However, large cities like Bogota can present challenges when it comes to the logistics of organizing, collecting, and interpreting the required information from everybody.

Daniel When we talked about the challenges faced in better city planning, it is important to recognize that in a country such as Colombia with a vast array of businesses and industries, engaging a diverse range of stakeholders is essential to the success of any strategic plan. However, the reality is that many citizens, particularly those in settlements surrounding major cities like Bogotá, face significant barriers to participation in physical meetings or online surveys, be it due to limited transportation options or lower levels of technological literacy. Therefore, finding alternative ways to engage with citizens and encourage participation is crucial in order to effectively gather and incorporate a range of perspectives into the city planning process.

Potential solutions to the challenge of engaging citizens in city planning could be to utilize platforms that are already widely used and accessible to the public. For example, in Colombia, WhatsApp is prevalent due to its accessibility through free social network packages offered by smartphone providers. Leveraging such platforms for communication and engagement could increase the likelihood of citizen participation and help ensure a more representative range of perspectives is incorporated into the city planning process.

Taliya ProBogotá decided to use the WhatsApp platform, utilizing a chatbot that employs ChatGPT technology to gather various data from community members regarding urban planning. The chatbot is used to ask residents what they would like to see in the city to make them want to settle down and raise their children there. The invitation to participate is open to everyone, but it specifically targets those living in the area, property owners, renters, or anyone else who might be directly affected by the strategic plan. Special emphasis is placed on groups and individuals who would typically not have access to such participatory processes due to limitations, long working hours, cultural or gender barriers, or illiteracy.

By leveraging the WhatsApp chatbot and incorporating innovative technologies like ChatGPT, ProBogotá can gather diverse perspectives from a broader range of residents, addressing the challenges posed by Colombia's social disparities and limited mobility. The incorporation of digital tools helps create a more inclusive and representative strategic planning process, empowering communities to have a direct impact on the future development of their city. The insights gathered from this process will be invaluable in shaping a strategic plan that truly reflects the needs and aspirations of the people of Bogotá.

Eran But the question arises, how can the capabilities of the chatbot be used to generate concrete insights from the people that the planners can use? Before beginning the planning process, it is essential for the planners to have a thorough understanding of the area they are going to work on. This requires conducting a statutory study, which involves obtaining information such as building permits, land use patterns, and legal ownership in the region. In addition, geophysical research is crucial, including geography, geology, hydrology, and aspects related to environmental hazards. The quality of built and open areas, road infrastructures, and service infrastructures are also studied alongside other relevant investigations.

To proceed further, the chatbot must gather data and classify it into five key themes. These themes are as follows:

- Environment and climate resilience
- Mobility (transport)
- Local identity
- Future of work
- Land use

When we talk about environment and climate resilience and Mobility (transport) there is a strong connection between them.

The interest in this aspect, when we approach urban planning, is largely driven by the increasing awareness and concern about climate change. Urban planning that takes into account environment and climate resilience will focus on measures to ease the effects of climate change, such as increased flooding or heat waves, and reduce the city's contribution to global warming.

According to [14] urban development needs to actively deal with environmental and climate resilience due to the combined impacts of urbanization and climate change, which seriously threaten the stability of the environment, economy, and society.

In addition to directly affecting the quality of life in cities, efficient transportation infrastructure can have a considerable impact on a city's environmental footprint. Cities are estimated to contribute between 40% to 70% of greenhouse gas emissions, a significant portion of which is related to the consumption of fossil fuels for transportation.

Urban planning can shape the local identity of a city by preserving cultural heritage, promoting local economic development, and designing public spaces that foster community interactions. Local identity can also influence the acceptance of urban projects by the local community

In [15] the paper goes on to examine the implications of place identity for planning. Jones argues that place identity can be a valuable tool for planners, as it can help them to understand how people perceive and identify with their cities. He also argues that place identity can be used to help planners to create more sustainable and livable cities.

The future of work is shaping urban planning because changes in the way we work, particularly the increase in remote working, have implications for the design of residential areas, transportation needs, and the provision of services in the city. This can also influence the spatial distribution of jobs in the city.

According to [16] Richard Florida and Adam Ozimek argue that the rise of remote work has the potential to fundamentally change the way we live and work in cities. Florida argues that remote work can help to reduce traffic congestion, improve air quality, and increase productivity. It can also help to create more opportunities for people to live in affordable housing and to work in places that are more in line with their values.

Land use planning is a key aspect of urban planning because it determines the location of housing, businesses, and services in the city. Land use decisions can influence traffic patterns, environmental quality, and the livability of neighborhoods

According to [17] the following are some of the key principles of good urban land use planning:

- Sustainability: Urban land use planning should be guided by the principles of sustainability. This means that planning should take into account the environmental, social, and economic impacts of land use decisions.
- Livability: Urban land use planning should aim to create a more livable city. This means that planning should take into account the needs of residents, such as access to jobs, housing, and services.
- Participation: Urban land use planning should be a participatory process. This means that all stakeholders, including government, businesses, and residents, should be involved in the planning process.

Taliya The chatbot gathers all this information, but the question arises, how will this clustering be performed? How can insights be extracted from this data? We have data from numerous conversations with various clients, and our research deals with classifying these conversations into the five main topics while considering the extraction of entities from the sentences. We argue that there is an influence of an entity in a sentence on its classification into a specific topic.

Since we are dealing with unlabeled data, we are talking about unsupervised learning using an entity extraction model. In this context, we used the language-agnostic BERT Sentence Embedding (LaBSE) [18] technique to generate the embedding vectors for the sentences and categories. LaBSE is an advanced natural language processing (NLP) method that allows for the creation of vector representations of sentences in a way that is agnostic to the input language, enabling semantically meaningful comparisons and analyzes of sentences across different languages. By using these word embeddings, each sentence and category will be represented by a vector, and we will employ mathematical metrics such as cosine similarity and Euclidean distance to estimate the distance of each sentence from each of the categories, taking into account the nouns in the sentence.

Additionally, we can use supervised learning technology by labeling the data using ChatGPT. By giving the chat a sentence, he will have to tag it into one of our 5 categories that we mention above. Upon obtaining the labeled data from ChatGPT, we pursued two distinct methods for generating sentence embeddings, aimed at making the dataset suitable for Multi-Layer Perceptron (MLP) model training [19]. These two methods were the LaBSE model and Term Frequency-Inverse Document Frequency (TF-IDF) via the TextVectorization layer in TensorFlow, marking two distinct approaches for text classification in our work.MLP

TF-IDF is a numerical statistic intended to reflect how important a word is to a document in a collection

or corpus. It measures the significance of each word by considering its frequency in the specific sentence and its scarcity in the entire corpus. This emphasizes words that are frequent in specific sentences but are generally rare in the corpus, thus highlighting the more informative words in the text.

In the first approach, we transformed the labeled sentences into vector representations using the LaBSE model. This yielded embeddings capable of capturing the context-aware and language-agnostic characteristics of the sentences.

In the second approach, we utilized TF-IDF for generating embeddings, focusing on the word frequency within the sentences, thus underlining the most informative and distinctive words in the text.

In both approaches, the vectors derived from these embeddings laid the foundation for training the supervised MLP models. This process enabled the models to predict categories based on the syntactic, semantic, and word importance characteristics of the sentences, as captured by the LaBSE and TF-IDF embeddings, respectively.

In summary, the Bogotá metropolitan region is a dynamic urban area facing significant socioeconomic challenges that need to be addressed through inclusive and representative urban planning processes. By leveraging innovative technologies like ChatGPT, LaBSE and TF-IDF ProBogotá can gather valuable insights from a broader range of residents, ultimately leading to more effective strategic plans that better reflect the diverse needs and aspirations of the region's population.

1.1. Our contribution

Eran & Daniel This paper presents an innovative approach to tagging data using ChatGPT. In addition, a chatbot is used to collect and organize resident feedback to inform the urban development team. The contributions in this paper are organized as follows:

- **Improving data collection and processing with deep learning technique.** The implementation of the chatBot eliminates the need for manual data collection from seven million residents. Instead, it leverages deep learning techniques to aggregate, analyze, and summarize this information efficiently, without the need for a human team to carefully analyze and summarize the responses.
- **Avoiding human intervention by using ChatGPT.** Using ChatGPT, the researchers skip the labor-intensive process of human data tagging. This advancement overcomes the logistical challenges and potential errors that humans would have made in tagging data.
- **Ensuring consistent data tagging using ChatGPT.** This approach eliminates the potential differences and inconsistencies that might have arisen from employing different taggers for the data, thereby improving the quality and reliability of the project's data.
- **Adopting a dual approach consisting of supervised and unsupervised methods for scalability.** The proposed solution combines supervised and unsupervised approaches to facilitate seamless expansion to additional categories. This two-way approach allows for immediate expansion to new categories in the unsupervised approach, while the supervised technique allows for easy adjustments to expand categories.
- **Offering a replicable model for urban planning with global potential.** This research adds a lot to making cities more sustainable worldwide. It figures out the best way to develop cities in the long run, and it uses Bogotá as an example. This approach is flexible in that it can be used in different parts of the world, making it a handy tool for improving urban planning strategies everywhere.

1.2. Paper structure

The remainder of this paper is structured as follows: Section 2 mainly includes a focus on ChatBot in combination with NER [20] and, embedding sentences with BERT, and also planning urban cities. Section 3 discusses the dataset used in this paper and the pre-processing procedure. Section 4 presents the approach employed in this paper which is an unsupervised and supervised method with different embedding techniques. Section 5 presents the results of the unsupervised and supervised methods with BERT and TF-IDF embedding and their conclusions. Finally, Section 6 provides a fundamental discussion, and concludes and summarizes this paper.

For ease of reading, Table 1 provides a list of abbreviations that are commonly used in this paper.

2. Related Work

Eran The topic of chatbots has been covered in quite a few works. In [21] the authors suggest a new way to create smart chat-bots that can better understand what the user is asking for. They do this by identifying key information in the user's input and using it to improve the chatbot's ability to recognize the user's intention. The method uses a combination of RASA NLU and neural network techniques. RASA NLU is a library for natural language processing. It extracts entities from user input by using techniques like tokenization [22],

Table 1: *List of abbreviations*

Abbreviation	# Meaning
BERT	Bidirectional Encoder Representations from Transformers
LaBSE	language-agnostic bert sentence Embedding
NLP	Natural Language Processing
NER	Named Entity Recognition
RASA	Real-time Application Support and Automation
NLU	Natural Language Understanding
LLM	Language Learning Model
TF-IDF	Term Frequency-Inverse Document Frequency
NLI	Natural Language Inference
STSb	Semantic Textual Similarity Benchmark
IS-BERT	Info Sentence BERT
STS	Semantic Textual Similarity
GPT	Generative Pre-training Transformer
MLP	Multi-Layer Perceptron
ReLU	Rectified Linear Unit
POS	part-of-speech

POS tagging [23], and NER. In our work, we use these similar methods to extract entities and keywords that relate to our 5 categories.

In [24] describes the development of an academic chatbot that uses natural language processing (NLP) and entity extraction to provide students with access to academic information. The chatbot was developed at ISB Atma Luhur, a university in Indonesia, and is designed to be used by students on mobile devices. The chatbot uses NLP to understand the intent of a student's query and to extract entities from the query. For example, if a student asks 'What is the course schedule for next semester?' The chatbot will use NLP to understand that the student is looking for information about course schedules. The chatbot will then use entity extraction to identify the entities in the query, such as "course schedule" and "next semester". When users will use our chatbot we will need to understand what they say and in which category to put it among the 5 we have. Understanding chatbot users is critical. If the users feed the chatbot with unrelated things we must filter them accordingly or alternatively give them an appropriate response based on what they write to the chatbot.

We can see another approach in [25] where the authors discuss the different approaches to knowledge entity extraction and text mining. One approach is to use supervised learning. Supervised learning is a machine learning technique that requires labeled data. Labeled data is data that has been manually annotated with the correct answers. Supervised learning can be used to train a model to extract knowledge from text. Another approach is to use unsupervised learning. Unsupervised learning is a machine learning technique that does not require labeled data. Unsupervised learning can be used to find patterns in unlabeled data. These patterns can be used to extract knowledge from text. In this article, both of the techniques will be used.

When we talk about the benefits of crowdsourcing in the context of urban places there are also possibilities like in [26] when they used to collect data in smart cities with mobile crowdsourcing. Mobile crowdsourcing is a type of crowdsourcing that uses mobile devices to collect data from a large number of users. Just like in our work. This data can be used to improve a variety of urban services, such as traffic management, public safety, and environmental monitoring and conditions, such as air quality and water quality. In contrast to our research that the collection of information with crowdsourcing using mobile is before the construction of the city.

Daniel It's not the first time that the topic of integrating citizen input into the urban planning process has been talked about. In [27] The authors suggest a different method for designing cities that involves taking into account the opinions and suggestions of the people who live in them. They believe that the old way of planning cities from the top down is outdated and ineffective for meeting the demands of contemporary urban living and that a more inclusive approach is required. In another study [28], the authors utilized an LLM chatbot to gather input from citizens about their preferences and ideas for the park's design, which was found to be effective, well-received, and user-friendly.

Renana Rimon In [29] the focus is on classifying words and expressions into data categories. That is, the classification is not in the context of a sentence but rather for a specific word. The classification is done using the following methods: Term Frequency - Inverse Document Frequency (TF-IDF) [30], and Linear Support Vector Machine [31]. The results of the research showed that the linear support vector model achieved the highest accuracy rate of up to 96.82%.

Sentence-BERT (SBERT) [32] is a modification of the BERT (Bidirectional Encoder Representations from Transformers) model [33] that is designed to generate fixed-length sentence embedding. The main idea behind SBERT is to fine-tune the pre-trained BERT model on a sentence-level task, such as sentence similarity, instead of a word-level task, as done in BERT. SBERT generates sentence embeddings by fine-tuning a pre-trained BERT model on a sentence-level task, using a Siamese or triplet network architecture, various pooling strategies, and data augmentation techniques. The SBERT sentences embedding were eval-

uated using the SentEval toolkit [34], which is a popular tool for assessing the quality of sentence embeddings. The evaluation yielded an impressive score of 87.69% for the SBERT-NLI-large model. This model was pre-trained on NLI datasets and then fine-tuned on the STSb (Semantic Textual Similarity Benchmark) dataset [35]. The model has good results, but its limitation is the lack of multilingualism. Each model is trained for only one language.

Taliya Shitreet The paper [36] proposes a method called Info-Sentence BERT (IS-BERT) for unsupervised sentence representation learning. The goal is to learn meaningful sentence embeddings without relying on labeled data. The method builds on top of BERT and uses a combination of 1-D convolutional neural networks (CNNs) and a new self-supervised learning objective based on mutual information maximization to derive meaningful sentence embeddings. The approach maximizes the mutual information between sentence-level global representations and token-level local representations, using a discriminator to decide whether pairs of sentence and token embeddings are from the same sentence or not. Experimental results show that IS-BERT significantly outperforms other unsupervised sentence embedding baselines on common semantic textual similarity (STS) tasks and downstream supervised tasks. The paper evaluated the performance of their proposed IS-BERT model using the SentEval toolkit. The evaluation showed that the IS-BERT model achieved an accuracy of 85.91% .

3. Datasets

Renana Rimon The research aims to reach a diverse and broad audience to obtain the most comprehensive and genuine public opinion. When working on a project based on collective intelligence, it is crucial to gather abundant data that represents the true target audience. However, collecting data, especially in a domain lacking tagged data, is not a straightforward task. The data needed consists of conversational sentences, such as WhatsApp messages, in the architecture topic. To collect this data, the researchers utilized a chatbot based on chatGPT, designed to engage in conversations with residents and gather as much relevant information as possible regarding the topic of constructing a public structure that suits the real needs of the residents. The initial dataset consisted of several dialogues conducted through the chatbot. The chatbot conversed with the client as an architect, asking targeted questions. The initial dataset comprised approximately 850 sentences from both the architect and the customer. The focus was on the responses received from the customer, totaling 360 sentences. The initial dataset size was relatively small, posing a challenge for the model to learn and accurately label the data. The conversations were held with a wide spectrum of people - women, men, young and old, as well as a broad range of socio-economic statuses, ranging from academics to illiterates. Individuals with low socio-economic status constitute the majority of the population in Bogotá Savanna, with low levels of education or no education at all. Consequently, it was not uncommon to receive irrelevant answers, answers with spelling errors, or incomplete sentences. The goal was to obtain clean data, containing only the relevant sentences. Furthermore, the field of architecture is extensive and encompasses various topics. Each sentence was to be labeled with the appropriate category it belongs to, including environment and climate resilience, mobility (transport), local identity, future of work, or land use. A sentence may be related to multiple categories or none of them, in which case it would be labeled as ”other”. Since the data was collected in an unlabeled format, suitable approaches were required to deal with it. Both unsupervised and supervised approaches were considered. The supervised approach involved data labeling. For the initial 300 sentences, could be considered to label the data manually, which was feasible for the initial small dataset, but would require a more efficient method as more data was accumulated. Therefore, an innovative approach for data labeling was proposed. The labeling process would be performed by utilizing chatGPT. Each sentence and the five categories would be inputted to chatGPT, which would then assign the sentence to the relevant categories. It was emphasized to ChatGPT that each sentence could belong to more than one category or none of them. If chatGPT did not assign any category to a sentence, indicating its irrelevance, it would be labeled as ”other” [Table 2].

Given the multi-labeled data, despite having 360 examples, the number of labels was 405. The data was divided into train, validation, and test sets, resulting in a very small number of examples in each set. This small dataset size could potentially hinder the model's ability to learn effectively. Additionally, the limited number of examples in the test set made it challenging to measure the results accurately, although some indicative results could still be observed. The dataset prepared for the research included 360 client sentences with labels generated by chatGPT, which were considered the ground truth for the research purposes [Table 3], [Table 4].

The research utilizes an additional dataset for training a Named Entity Recognition (NER) model to extract entities. The dataset used is the GMB (Groningen Meaning Bank) corpus [37], which has been annotated and tagged for entity classification. Natural Language Processing techniques are applied to this dataset, incorporating enhanced and popular features. The GMB corpus provides context and serves as a valuable resource for training the classifier to predict

Table 2: *Sentences and Categories*
Part of the dataset consists of sentences labeled by the chatGPT.

Sentence	**Category**
”thank you and bye”	[’other’]
”Where will the building be located?”	[’Mobility (transport)’]
”safe”	[’Environment and climate resilience’]
”to create a public space in front of the building”	[’Land use’, ’Local identity’]
”Perhaps more accessible entrances larger public spaces more communal areas”	[’Environment and climate resilience’, ’Land use’]
”Could you describe in your own words the example I provided?”	[’Future of work’]
”There should be many student gathering areas.”	[’Land use’]
”I hope there will be some green plants in the building, plants always make people feel more comfortable”	[’Environment and climate resilience’]
”I want a garden on the roof”	[’Land use’]
”The building should be made of recyclable materials”	[’Environment and climate resilience’]

named entities such as names, locations, and more. The dataset contains essential information about different types of entities, including geographical entities, organizations, persons, geopolitical entities, time indicators, artifacts, events, and natural phenomena. The total word count in the dataset is 1,354,149.

4. Framework

Renana Rimon In order to train a deep learning model to classify sentences into different categories, there are two approaches to solving the problem. One approach is the unsupervised method, where untagged data can be used to train the model. In the unsupervised method, a smaller variant of the LaBSE (Language-agnostic BERT Sentence Embedding) model was employed to generate sentence embeddings. To prepare the input for the model, we used an NER (Named Entity Recognition) model to extract entities from the sentence. For each sentence, the input to the LaBSE model consisted of three versions: 1. The original sentence. 2. The list of entities extracted by NER. 3. A sentence that includes only the entities. These embeddings capture the semantic meaning of the sentences in a numerical representation. By calculating the cosine similarity between the sentence embeddings and predefined categories, the method determines the relevance or similarity of the sentences to each category.

Table 3: *Data Distribution*
Statistics of number of sentences & number of labels in the data. The data is multi-labeled, so #labels >#sentences. The statistic show in the table is 1) all data. 2) run with split 80% train, 10% validation, 10% test.

Data	Total # of sentences	Total # of labels
All data	360	405
Train (80%)	288	325
Validation (10%)	36	40
Test (10%)	36	40

The second approach is the supervised method, which requires labeled data. The research proposes an innovative approach to labeling the data using the ChatGPT model. The input provided to the model is a sentence and the five categories that represent subcategories in the architecture domain. It is emphasized that the labeling is multi-label, meaning each sentence can be classified into multiple categories or none of them. If ChatGPT does not assign the sentence to any of the categories, the label will be ”other,” indicating it is not relevant to the architecture domain.

When labeled data is available, various deep learning models can be trained. The research utilizes two approaches for sentence embedding: the first is Language-agnostic BERT, and the second is TF-IDF. After embedding the sentences using these approaches, the data enters a fully-connected neural network for training.

To evaluate the model’s results using different metrics such as accuracy, precision, recall, and F1 score, the data is initially divided into train and test sets. After training the model, the test data can be fed into the model for prediction and evaluation of the results. On the other hand, in the unsupervised approach, there are seemingly no labeled data. To evaluate the results, the ChatGPT’s labeling is considered as the ground truth and compares the results.

In Section 4.1, we present the unsupervised solution, while in Section 4.2, we demonstrate the supervised solution, which includes the two embedding methods. Section 4.3 describes how we simulate a scenario where the unsupervised solution seemingly received ground truth labeling to evaluate the results.

4.1. Unsupervised Method

Eran When we talk about the unsupervised method in the context of our research, we will first explain 2 main things that are important for the rest of the article. The first one is Named Entity Recognition (NER) which is a sub-task of information extraction that identifies and classifies named entities in text into predefined cate-

Table 4: *Data Distribution by Category*
Statistics of number of labels For each label category, The statistic shown in the table is 1) all data. 2) run with split 80% train, 10% validation, 10% test.

	Other	**Local Identity**	**Future of Work**	**Environment and Climate Resilience**	**Land Use**	**Mobility (Transport)**
All data	88	73	48	76	108	12
Train (80%)	70	59	39	61	86	10
Validation (10%)	7	5	5	8	14	1
Test (10%)	11	9	4	7	8	1

gories such as person names, organizations, locations, etc.

Daniel The NER model employed in this project follows a straightforward process. It begins by taking a sentence as input and proceeds with tokenization, breaking the text into individual words and phrases known as tokens. These tokens are then fed into the BERT model, which generates a series of hidden states representing the model's understanding of the input text. The subsequent step involves passing the hidden states through a classifier component, responsible for predicting the entity type for each token. This classifier is trained on a labeled dataset containing the correct entity types for each token. Taking advantage of its extensively trained classifier, the NER model accurately identifies named entities within the given text, demonstrating exceptional accuracy in classifying these entities.

In this project, after assigning each word to its respective POS, the model selectively extracts and saves the identified nouns for further analysis and processing. Importantly, the original sentences are retained alongside the extracted nouns, enabling a comprehensive understanding of the context in which the nouns are found. During the training phase, the NER model utilizes a technique known as fine-tuning, which involves customizing a pre-trained BERT model specifically for the task of named entity recognition. The fine-tuning procedure initiates by tokenizing the input text, breaking it into individual words and phrases. The BERT model processes the resulting tokens, generating hidden states that capture the model's understanding of the input text. These hidden states are then fed into a classifier, which predicts the entity type for each token. To train the classifier, a labeled dataset is used, containing the correct entity types for each token. Supervised learning is utilized to train the classifier. In this process, the model is presented with input-output pairs, where the input pairs represent the text tokens, and the output pairs indicate the correct entity types. The model is then trained to predict the accurate entity type for each input token.

The second main topic is Language-Agnostic BERT Sentence Embeddings (LaBSE)[5.1.2]. ¡replacement command¿

The unsupervised algorithm consists of several steps. In the initial stage, NER is utilized to extract nouns from the dataset. Next, a smaller-LaBSE model is employed to compute the proximity between sentences and pre-selected categories.

This stage involves three parts:

- Assessing the sentence's proximity to each category.
- Evaluating the proximity of the noun sequence to each category.
- Determining the average proximity of each noun in the sentence to each category.

Each part aims to determine the degree of sentence-category association. To achieve this, the sentences are encoded using a pre-trained model, generating vector representations. These vectors are then compared to the category vectors using cosine similarity, which measures their similarity based on orientation. A manually defined threshold is applied to determine the results, ensuring optimal performance of the unsupervised models. Note that the threshold plays a crucial role in optimizing the unsupervised models by controlling the level of association between sentences and categories. The category "other" was assigned to each sentence that did not receive any category during labeling.

The selection of an appropriate distance calculation technique is of utmost importance as it significantly impacts the outcomes of clustering. Depending on the nature of the data and the research objectives, different dissimilarity measures may be more suitable. For example, when dealing with numerical data and aiming to assess the distance between points in a coordinate system, Euclidean distance serves as a valuable tool, providing insights into the similarity or dissimilarity based on numerical attributes. However, Euclidean distance may not be well-suited for

high-dimensional data, such as text data represented as high-dimensional vectors with each dimension corresponding to a unique word or feature. In contrast, cosine similarity measures similarity by considering the angle between vectors, which remains stable and meaningful in high-dimensional spaces.

Cosine similarity offers several advantages over Euclidean distance. Firstly, it is not influenced by the magnitude or length of vectors being compared, focusing solely on their direction or orientation. This property is particularly useful in text classification tasks where the frequency of words may vary widely, but their importance for classification is not necessarily related to their frequency. Cosine similarity enables effective comparisons based on the relevance of shared words, irrespective of their frequency.

Moreover, cosine similarity has been shown to capture semantic similarity better than Euclidean distance for text data. By focusing on the orientation of vectors, cosine similarity can identify similarities in terms of the meaning or context of words, even if their actual representations differ. This characteristic makes it well-suited for tasks such as document similarity, information retrieval, and recommendation systems.

Additionally, cosine similarity overcomes a challenge encountered in text data, which often exhibits sparsity where most dimensions or features have zero values. In such cases, Euclidean distance may exhibit bias towards documents with more non-zero values, whereas cosine similarity remains unaffected by sparsity. It allows for effective comparisons even when a majority of dimensions are zero, as it considers only the non-zero dimensions.

In contrast, Manhattan distance, another distance calculation technique, focuses on the absolute differences between coordinates rather than capturing the semantic relationship between text documents accurately. Hence, cosine similarity is preferred over Manhattan distance in text-related tasks. The cosine similarity metric is computed using the following equation:

$$\text{Cosine Similarity} = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

Here, $(A \cdot B)$ represents the dot product of vectors A and B, while $||A||$ and $||B||$ represent the Euclidean norms or magnitudes of vectors A and B, respectively. Cosine similarity values range from -1 to 1, with 1 indicating perfect similarity, 0 indicating orthogonality or no similarity, and -1 indicating complete dissimilarity.

Eran In our implementation of the unsupervised method we have 3 approaches that we mention above. For each option, we will check to which category the sentence belongs (by calculating the computational distance between the vector representing the sentence and the vectors representing the categories). And In the end, we will get a .csv file that will assign each sentence to a certain category or more.

After we get an output we need to check how close we were to the tag that the ChatGPT outputted using the supervised method.

In order to do this we will compare the results of the 2 methods. In other words, we will compare the categories we received in each option we mentioned above using the metrics F1 score, Precision, Accuracy, and Recall.

4.2. Supervised Method

Taliya [Figure2] Catering to a multi-class [38] and multi-label [39] classification problem. Data, labeled using ChatGPT technology, served as the basis for the supervised solution. The primary challenge was assigning appropriate categories to an incoming, unseen sentence, a classic text classification problem. To manage this problem, an algorithm under supervised learning methods was employed. Data preprocessing was an essential first step. The rows representing sentences without any labeling were removed, as these null values could distort the learning process and impact the normalization level of machine learning algorithms. Further, to prevent the biasing of the learning process due to duplicate sentences, all duplicates were carefully identified and removed. A method to category analysis revealed 14 categories appearing only once and a total of 31 unique categories in the dataset. Instead of discarding these rare categories, which could have led to potential information loss, the categories were split into different rows, each becoming associated with a particular sentence. This augmented the dataset, addressing the challenge of an uneven category distribution. Following preprocessing, the refined dataset included 360 non-duplicate, well-prepared samples, each associated with their respective categories.

For the next stage of processing, the data was split into training and testing sets. The data was encoded to become compatible with deep learning models. The model used TensorFlow's StringLookup function to transform the categories into a multi-hot format - an ideal choice for a multi-label classification task. The process also included an inversion function to decode the multi-hot encoding back into the original categories, which aided model interpretability and validation. Sentence embedding is a crucial step in transforming human language into numerical representations that can be processed by machine learning models. In this framework, the transformation was achieved using three distinct text vectorization techniques: TextVectorization TF-IDF 4.2.1 and LaBSE 4.2.2. The final stage involved constructing a deep learning model based on a Multi-Layer Perceptron (MLP) architecture . The MLP model, also known

as a feed-forward neural network. In this model, four hidden layers were used, consisting of 512, 256, 128, and 64 neurons respectively. Each hidden layer used a Rectified Linear Unit (ReLU) activation function [40]. ReLU activation function introduces non-linearity into the network, enabling the model to learn and solve complex problems and is often a default choice for hidden layers. The output layer, on the other hand, used a sigmoid activation function [41]. The sigmoid function, also known as the logistic function, maps real-valued numbers into the range between 0 and 1, which can be used to represent probabilities for binary and multi-label classification problems. In the case of this multi-label text classification task, the sigmoid function allowed the model to assign a separate probability to each category, indicating the likelihood of an input sentence belonging to that specific category. The model's performance was evaluated during the training and testing phases using several metrics: Binary Accuracy, Precision, Recall, and F1 score. [42] The model was trained for 16 epochs, with a loss function set to "binary crossentropy" [43], considering the multi-label nature of the classification task. Adam Optimizer [44], a popular adaptive learning rate optimization algorithm, was applied.5

4.2.1. TF-IDF

The TextVectorization layer from TensorFlow was employed to transform text into a vector representation using the Term Frequency-Inverse Document Frequency (TF-IDF) method. In this process, each sentence in the dataset was tokenized into terms (words or 2-grams), and a TF-IDF score was calculated for each term within the sentence. The score evaluates the importance of a term within a sentence in the context of the whole dataset, thereby reflecting the significance of the term in representing the sentence's content.

Mathematically, TF-IDF is calculated as follows:

Term Frequency (TF) is calculated as the count of a term 't' in a document 'd' (a sentence, in this case) divided by the total number of terms in the document 'd':

$$TF(t,d) = \frac{\text{count of term } t \text{ in document } d}{\text{number of terms in document } d}$$

Inverse Document Frequency (IDF) is the logarithmically scaled inverse fraction of the documents that contain the word, obtained by dividing the total number of documents by the number of documents containing the term 't':

$$IDF(t) = \log_e \left(\frac{\text{Total number of documents}}{\text{Number of documents with term } t \text{ in it}} \right)$$

The **TF-IDF** score is simply the product of the TF and IDF scores of the term:

$$TF - IDF(t,d) = TF(t,d) \times IDF(t)$$

This TF-IDF score is assigned to each term in each sentence of the dataset, thereby converting each sentence into a vector of TF-IDF scores.

The strength of TF-IDF lies in its simplicity and its ability to highlight document-specific significant words. It excels in reducing the influence of common words, thereby allowing the model to focus on more unique and category-specific words.

However, TF-IDF does have limitations. It doesn't consider semantic relationships and the context of words, which could impact the classification of complex texts where understanding sentence semantics is crucial. It treats all words equally and ignores the order of words in a sentence, potentially missing important language nuances.

Moreover, TF-IDF struggles with multilingual data as it is language-specific. This is a crucial limitation in this work with a multilingual dataset; it may not fully capture the complexities and variations of multilingual text data. Despite these, TF-IDF serves as a reliable foundational approach for text vectorization.

4.2.2. Language-Agnostic BERT (LaBSE)

Another approach was adopted to generate sentence embeddings using the Language-Agnostic BERT Sentence Embedding (LaBSE) model. This model, designed to yield robust sentence embeddings for various languages, offers vector representations of sentences that encapsulate their semantic meaning and structural characteristics.

The conversion of a sentence into its vector representation using LaBSE is a multi-step process. Firstly, the input sentence s is tokenized into a sequence of subword units utilizing the WordPiece tokenization strategy [45], denoted as $T(s)$. The pre-trained LaBSE model, built on the transformer-based neural network architecture, then encodes this tokenized sequence into a sequence of hidden states, denoted as $h = B(T(s))$, where B stands for the LaBSE model.

The size of the output vector hinges on the model's hyperparameters. The variant of LaBSE used in this work provides an output vector of a fixed length of 768 dimensions. The vector representation v is obtained from the final hidden state of the CLS token, a distinct token located at the beginning of the input sequence. This vector is denoted as $v = S(h)$, where S is the selection operation extracting the CLS token hidden state.

This vector representation captures the syntactic and semantic properties of the sentence, making it valuable for diverse Natural Language Processing (NLP) tasks including text classification, sentiment analysis, and information retrieval.

After the sentence embeddings are generated, they are normalized using the L2-normalization technique [46]. Denoting the sentence representation as v, the

L2-normalization can be expressed as:

$$N(v) = \frac{v}{||v||} \tag{1}$$

Here, N denotes the normalization function, and $||v||$ represents the L2 norm of the vector v. This normalized sentence representation $N(v)$ ensures uniformity of scale across all embeddings, thereby enhancing their effectiveness during model training.

These generated sentence embeddings, enriched with semantic and contextual information, can be employed to measure semantic similarity between sentences, facilitate multilingual translations, or assist in any NLP task that requires a comprehensive representation of the input text.

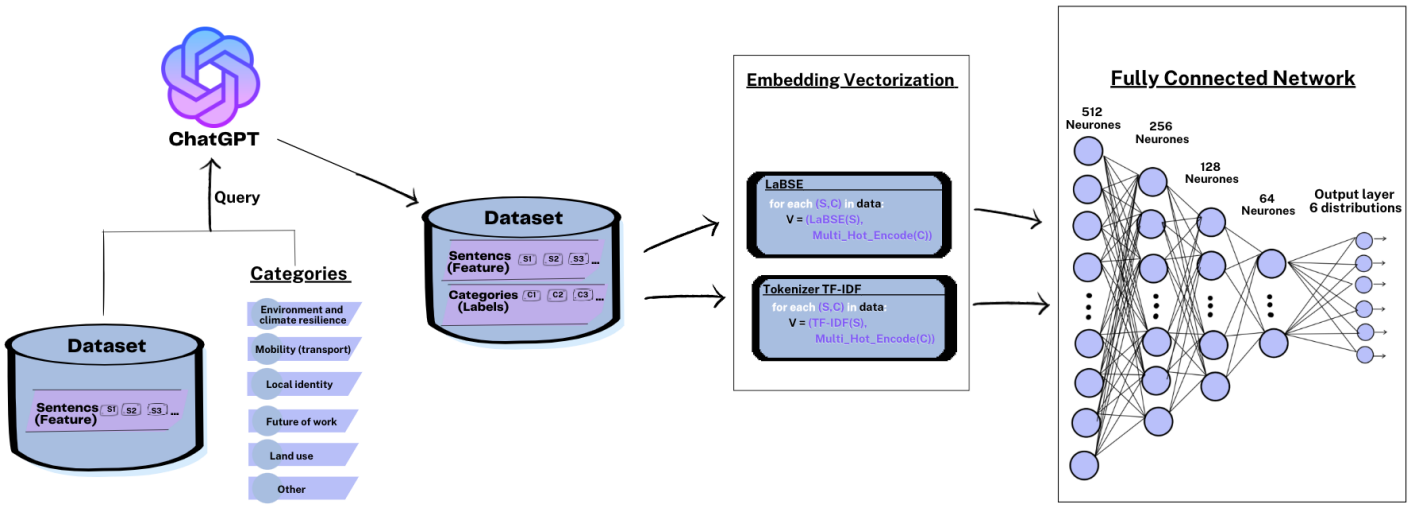

Figure 2: *This figure provides an illustration of the* ***Supervised learning process****. Each original sentence 's' from the dataset (sourced from chat conversations by Pro-Bogota) undergoes a query with six categories for tagging by ChatGPT, resulting in a tagged dataset. This dataset, consisting of a feature column of sentences and a corresponding multi-label category column, undergoes embedding via two distinct methods: Language-agnostic BERT Sentence Embedding (LaBSE), which performs multilingual sentence vectorization considering the sentence's context, and Term Frequency-Inverse Document Frequency (TF-IDF), which performs sentence vectorization considering word frequency, without reference to context. In both methods, the labels are vectorized using a multi-hot encoding scheme. Once the numerical data is available, it serves as input for a fully-connected neural network, which undergoes a learning process. The network output consists of six probabilities, each representing the likelihood of a sentence being associated with a specific category.*

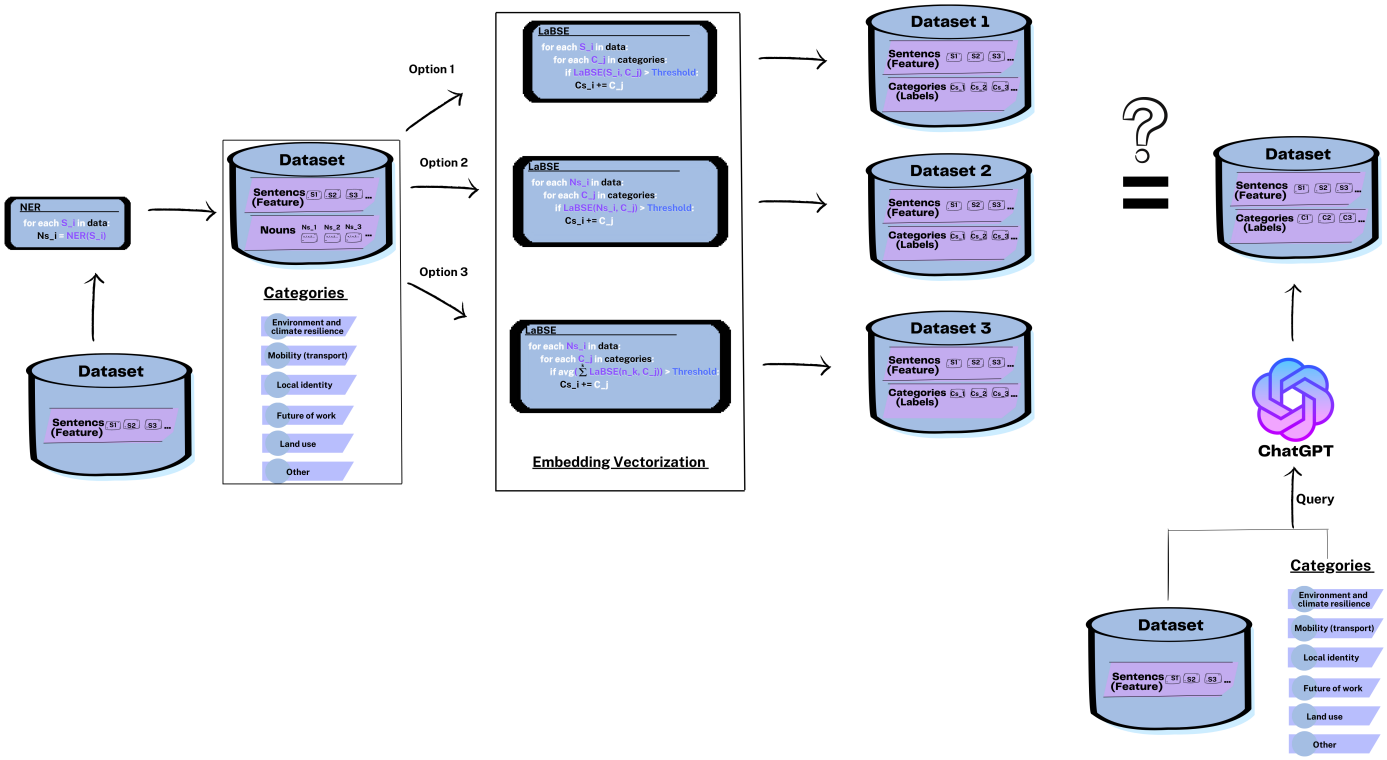

Figure 3: *This figure provides an illustration of the* ***Unsupervised learning process****. In this process, original sentences from a data set are passed through a Named Entity Recognition (NER) model to extract the nouns within each sentence. The LaBSE model is used for the embedding process. The unsupervised learning process offers three options for performing embedding: Option 1: Embedding the complete sentence. Option 2: Embedding a chain of nouns within the sentence. Option 3: Embedding each noun separately and averaging the results. After the embedding step, the cosine similarity is calculated between the vector representing the sentence and the vector representing each category. If the similarity result exceeds a certain threshold, the sentence is associated with that particular category. Consequently, for each option, a data set is generated, comprising sentences as features and their corresponding categories as labels. These data sets can then be compared against a data set labeled by ChatGPT (truth ground) using evaluation metrics such as accuracy, precision, recall, and F1 score.*

5. Experimental Evaluation

Eran There are a few ways to evaluate the metrics, which are the results of the model, that are commonly used:

- **Accuracy** Is a measure of how often the model makes correct predictions. It tells us the percentage of instances that the model classified correctly out of all the instances it predicted. For example, if a model has an accuracy of 85%, it means that it correctly predicted the outcome for 85 out of every 100 instances.

  The accuracy is given by the formula:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

	Actual Positive (1)	Actual Negative (0)
Predicted Positive (1)	TP	FP
Predicted Negative (0)	FN	TN

Figure 4: *Confusion matrix*

  where TP (True Positives) is the number of instances correctly classified as positive, TN (True Negatives) is the number of instances correctly classified as negative, FP (False Positives) is the number of instances incorrectly classified as positive, and FN (False Negatives) is the number of instances incorrectly classified as negative.

- **Precision** Precision is a measure of how accurate a classifier is when it predicts positive instances. It focuses on the proportion of correctly predicted positive instances out of all instances that the classifier labeled as positive. It is given by the formula:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

  For example, if the classifier predicts that 100 emails are spam, and it correctly identifies 90 of them as spam while misclassifying 10 non-spam emails as spam, the precision would be 90%. It means that out of all the emails the classifier labeled as spam, 90% of them were indeed spam.

- **Recall** Recall is a metric that measures the ability of a classifier to find and correctly identify all positive instances. It tells us the proportion of actual positive instances that the classifier correctly detects.

  A high recall means that the classifier is able to find most of the positive instances, while a low recall indicates that the classifier is missing a significant number of positive instances.

  The formula for the recall is:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

- **F1 score** The F1 score takes both precision and recall into account and gives you a balanced measure of the model's performance. It's like finding a middle ground between precision and recall.

  A higher F1 score means the model is doing well in both correctly identifying positive cases and finding all the positive cases. A lower F1 score indicates that the model may be lacking in one or both of these areas.

  F1 is calculated using the harmonic mean of precision and recall. The formula for F1 is as follows:

$$F1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{5}$$

5.1. Unsupervised Experiment and Conclusion

In option 1 [5.2] the result are:

Table 5: *Performance Metrics for Different Categories presents the performance metrics of option 1 when we send to the model all the sentences, including accuracy, precision, recall, and F1 score, for our 6 categories. The data highlights variations in these metrics across the categories.*

Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
71.71	73.68	13	22.04
40	27	80	40.3
80.85	63.63	27.6	38.53
80.5	100	8.10	15
82.57	20	8.17	11.6
92.28	10.52	16.66	12.9

When looking at category number 1 the category has a relatively high accuracy of 71.71% and a precision of 73.68%, meaning that the model is fairly good at correctly predicting this category. However, the recall is quite low (13%), implying that the model is not

capturing all relevant sentences that should be classified under this category. The low F1 score (22.04%) also suggests an imbalance between precision and recall, primarily due to the low recall.

In category 2 the model's performance is mixed. It has a lower accuracy (40%) and precision (27%) but very high recall (80%). This suggests that while the model is good at identifying sentences that relate to mobility, it's likely classifying too many sentences into this category, resulting in lower precision. The balanced F1 score (40.3%) reflects this trade-off.

Category 3 shows high accuracy (80.85%) but lower precision (63.63%) and recall (27.6%). The high accuracy suggests that the model is generally good at identifying whether a sentence is about local identity or not, but it's not as successful in correctly classifying these sentences (reflected by the lower precision) or identifying all relevant sentences (reflected by the lower recall). This is also mirrored in the moderate F1 score (38.53%).

In 4, the category shows high accuracy (80.5%) and perfect precision (100%), but extremely low recall (8.10%), leading to a very low F1 score (15%). This suggests the model is highly precise when it classifies a sentence as relating to the future of work - it's always right - but it's missing a lot of sentences that should be categorized as such.

In 5, the model has high accuracy (82.57%) but low precision (20%), recall (8.17%), and F1 score (11.6%) in this category. This means that while the model can accurately predict whether a sentence is about land use or not, it struggles to correctly classify these sentences (low precision) and capture all relevant sentences (low recall). The last one, category number 6 has the highest accuracy (92.28%) but low precision (10.52%) and moderate recall (16.66%), leading to a low F1 score (12.9%). This suggests that the model is good at determining if a sentence does not fit into the other five categories, but it may be over-classifying sentences into this "other" category, reducing precision.

In conclusion, it seems that the model is generally accurate but struggles with precision and recall in most categories, particularly recall. This indicates that it's often able to correctly identify when sentences are not related to a specific category (high accuracy), but struggles to identify all the relevant sentences that should be classified under that category (low recall), and often misclassified sentences (low precision).

For option 2 [5.2] the results are:

For category 1 **Environment and climate resilience** the accuracy dropped from 71.71% to 65.71%, and precision dropped from 73.68% to 42.85% compared to option 1. However, the recall has increased from 13% to 33.33%, resulting in an increased F1 score of 37.5%. This shows that the model identified more relevant sentences for this category but also made more mistakes in classifying sentences.

Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
65.71	42.85	33.33	37.5
53.71	32.33	73.33	44.89
81.14	70.83	22.36	33
76	32.14	12.16	17.64
75.42	21.5	28.57	24.45
87.14	7.7	25	11.76

Table 6: *Performance Metrics*
presents the performance metrics of option 2 when we send to the model all nouns in a concatenated string. The table includes accuracy, precision, recall, and F1 score, for our 6 categories. The data highlights variations in these metrics across the categories.

For **Mobility (transportation)** there's a significant increase in accuracy (from 40% to 53.71%) and precision (from 27% to 32.33%). Recall is slightly lower (from 80% to 73.33%), but the F1 score increased from 40.3% to 44.89%. This indicates improved model performance for this category.

For **Local identity** accuracy increased slightly from 80.85% to 81.14%. However, precision increased (from 63.63% to 70.83%), while recall dropped significantly (from 27.6% to 22.36%), resulting in a decrease in F1 score to 33%. The model seems to be better at identifying non-relevant sentences but worse at identifying all relevant sentences.

For **Future of work** accuracy decreased slightly (from 80.5% to 76%), and precision significantly dropped from 100% to 32.14%. Recall also decreased from 8.1% to 12.16%. Consequently, the F1 score increased to 17.64%. Despite lower precision, the model is now better at identifying relevant sentences.

For **Land use** accuracy dropped from 82.57% to 75.42%. Precision slightly increased from 20% to 21.5%, but recall saw a significant increase from 8.17% to 28.57%, leading to an increased F1 score of 24.45%. The model is better at identifying relevant sentences but makes more mistakes in classifying sentences.

And for the last category, **Other** accuracy dropped from 92.28% to 87.14%. Precision significantly decreased from 10.52% to 7.7%, and recall increased from 16.66% to 25%, resulting in a slight decrease in the F1 score to 11.76%. The model is better at identifying relevant sentences but makes more mistakes in classifying sentences.

In conclusion, option 2 seems to have improved recall in most categories but often at the expense of precision, except for Category 2 (Mobility), where all metrics improved. This suggests that focusing on

nouns in sentences allows the model to capture more instances of each category, but it also leads to more misclassifications. The overall trade-off depends on the specific importance of precision and recalls for your task. If finding all relevant sentences (even at the risk of including irrelevant ones) is the goal, then option 2 may be a good choice.

For option 3 [5.2] the results are:

Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
70.57	53.96	31.48	39.76
42.57	30.52	96	46.4
78	45.45	6	11.5
76.57	36.66	14.86	21.15
77.42	17.4	16.32	16.8
83.42	4	16.6	6.45

Table 7: *Performance Metrics*
Presents the performance metrics of option 3 when we send to the model each noun from the sentence and calculate the avg for each category. The table includes accuracy, precision, recall, and F1 score, for our 6 categories. The data highlights variations in these metrics across the categories.

For category 1 **Environment and climate resilience**, the accuracy slightly decreased from 71.71% (option 1) to 70.57%, but is higher than option 2. Precision is significantly better than both previous options. Recall improved over option 1, but is lower than option 2. The F1 score is the highest among the three options, suggesting a better balance between precision and recall.

For **Mobility (transportation)**, accuracy is slightly higher than option 1, but lower than option 2. Precision is higher than both previous options. Remarkably, recall is extremely high at 96%, significantly better than both previous options. The F1 score is also the highest among the three options, indicating an improved balance between precision and recall.

For **Local identity**, the accuracy decreased slightly compared to the two previous options. Precision is lower than in option 1 but higher than in option 2. The recall is significantly lower than in both previous options. The F1 score is the lowest among the three options.

For **Future of work**, the accuracy is slightly lower than in both previous options. Precision is higher than in option 2 but lower than in option 1. Recall is higher than in option 1 but lower than in option 2. The F1 score is higher than in option 1 but lower than in option 2.

For **Land use**, the accuracy is slightly lower than in both previous options. Precision is lower than in option 2 but higher than in option 1. The recall is lower than in both previous options. The F1 score is slightly higher than in option 1 but lower than in option 2.

And for the last category **Other**, the accuracy is lower than in both previous options. Precision is significantly lower than in both previous options. The recall is similar to option 2 but lower than option 1. The F1 score is the lowest among the three options.

In conclusion, option 3 resulted in some improvements over the two previous options, particularly for Categories 1 and 2. However, performance in other categories generally decreased. The high recall for Category 2 is impressive and suggests that this option may be particularly effective for categories where certain key nouns are very indicative of the category. The lower performance in other categories could be due to the key context being lost when individual nouns are considered separately.

5.2. Supervised Experiment and Conclusion

Eran We saw an interesting trend when splitting the training and testing system. By performing two sets of runs, one with an 80-20 split and another with a 70-30 split, we emphasize the importance of data volume in training the model. A larger training set (80-20 split) generally produced superior results compared to a smaller training set (70-30 split). This aligns with the general understanding in machine learning that more data often leads to better model performance. However, it's also crucial to remember that the quality and diversity of data are just as important.

The approach of evaluating the model separately for each category has provided granular insights into the model's performance across various labels. This approach has enabled us to identify which categories the model is strong in and where it struggles, which can guide future improvements.

In the testing phase, we evaluated the model separately for each category, providing detailed insights into its performance across different labels. We generated predictions on the test set, converted them to binary format, and calculated individual metrics for each category.

5.3. TF-IDF

5.3.1. TF-IDF - Train: 80, Test: 20

5.3.2. TF-IDF - Train: 70, Test: 30

Upon comparing the two results for the TF-IDF approach (80/20 vs 70/30 train/test split) In all categories, apart from 'Future of Work' (Category 4) and 'Other' (Category 6), show a decrease in performance metrics when the split changes from 80/20 to 70/30. This could suggest that the model is benefiting from more training data, and thus the reduced size of the

Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
85.18	80	80	80
88.89	100	57.14	72.72
81.48	50	60	54.55
85.86	66.67	40	50
92.59	100	33.33	50
96.3	0	0	0

Table 8: *Performance Metrics*
presents the performance metrics when we use the TF-IDF embedding by splitting the train and test to 80 and 20. The table includes accuracy, precision, recall, and F1 score, for our 6 categories. The data highlights variations in these metrics across the categories.

Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
63.89	50	69.23	58.06
72.22	41.67	62.5	50
61.11	0	0	0
80.56	0	0	0
91.67	0	0	0
94.44	0	0	0

Table 10: *Performance Metrics*
presents the performance metrics when we use the BERT embedding by splitting the train and test to 80 and 20. The table includes accuracy, precision, recall, and F1 score, for our 6 categories. The data highlights variations in these metrics across the categories.

Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
75.93	61.11	64.71	62.86
72.22	42.86	46.15	44.44
72.22	37.5	23.08	28.57
83.33	16.67	20	18.18
74.07	22.22	22.22	22.22
92.59	0	0	0

Table 9: *Performance Metrics*
presents the performance metrics when we use the TF-IDF embedding by splitting the train and test to 70 and 30. The table includes accuracy, precision, recall, and F1 score, for our 6 categories. The data highlights variations in these metrics across the categories.

Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
66.67	46.15	75	57.14
55.56	35.71	62.5	45.45
79.63	0	0	0
83.33	0	0	0
88.89	0	0	0
96.3	0	0	0

Table 11: *Performance Metrics*
presents the performance metrics when we use the BERT embedding by splitting the train and test to 70 and 30. The table includes accuracy, precision, recall, and F1 score, for our 6 categories. The data highlights variations in these metrics across the categories.

training set in the 70/30 split might be negatively affecting performance.

5.4. BERT

5.4.1. BERT - Train: 80, Test: 20

5.4.2. BERT - Train: 70, Test: 30

When observing the results for Environment and Climate Resilience (Category 1) despite relatively lower accuracy compared to the TF-IDF method, BERT shows higher recall in both splits. This means the model identifies a higher portion of relevant instances. However, precision suffers, indicating the model often incorrectly labels sentences as belonging to this category. The F1 score slightly decreases as well. In Mobility (Category 2) Similar to Category 1, the model with BERT has a lower precision but higher recall than the one with TF-IDF. The accuracy, precision, and F1 score have reduced considerably in both splits compared to the TF-IDF approach, suggesting that BERT is less capable of handling this category.

In summary, the model with BERT embeddings shows a substantial decline in performance compared to the TF-IDF method. This could be due to various reasons such as the higher complexity of BERT, which may require more data to effectively learn, or it might indicate that the BERT model parameters need fine-tuning to better suit the task. Furthermore, the complete failure in predicting categories 3, 4, 5, and 6 suggests that these categories might be more challenging to distinguish, require more training examples, or need additional features that BERT does not capture.

6. Conclusion and Future Work

Eran The research aimed to explore and compare different supervised and unsupervised techniques for categorizing text data into predefined categories. The text data in this study was separated into six categories. The study focused on two types of word embeddings – Term Frequency-Inverse Document Frequency (TF-IDF) and BERT (Bidirectional Encoder Representations from Transformers), applying these embeddings in both supervised models, and in the unsupervised method using only BERT.

In the initial phase of the research, an unsupervised approach was implemented with a clustering algorithm. Here, the data was not labeled. Instead, the algorithm was used to cluster the data based on similarity in word embeddings. The primary aim was to see how well the natural structure of the data could be discovered by the algorithm. The performances of the algorithm with BERT embeddings were compared between 3 options. [5.2]

The results indicated that the unsupervised learning method had varying levels of success, largely dependent on the chosen word embeddings and the specific category. In general, it was found that while the unsupervised technique could cluster the data reasonably well, it struggled to capture some of the nuances of certain categories, particularly when the categories were more closely related or overlapping.

Next, the research delved into a supervised learning approach. This time, the models had access to labeled data. They were trained to predict the predefined categories using the TF-IDF and BERT embeddings. To evaluate the model performance, different data split ratios were used, and metrics like accuracy, precision, recall, and F1-score were calculated for each category.

For TF-IDF, the models generally performed well with the 80/20 train/test split and showed some performance decline when the split was adjusted to 70/30. This suggested that the models benefited from a larger training set and that the performance could be negatively affected when the size of the training set was reduced.

The performance of BERT embeddings was quite different. In some categories, BERT showed higher recall compared to TF-IDF but suffered from lower precision. The accuracy and F1-score were also lower with BERT. Most notably, BERT completely failed to predict some categories in both splits, possibly indicating a need for more training examples or fine-tuning of model parameters.

In conclusion, this paper presented an extensive comparison of supervised and unsupervised methods for text categorization with two different types of word embeddings. The findings suggested that the choice of word embeddings and the amount of training data can significantly impact model performance. While TF-IDF provided strong results in a supervised setting, the more complex BERT embeddings did not perform as well, likely due to the need for more extensive training or parameter tuning.

Future research could delve deeper into understanding why certain categories were more challenging to predict, potentially looking at the unique characteristics of these categories or exploring different model architectures. It would also be interesting to experiment with other word embeddings and learning approaches.

This study noticed substantial differences in performance across different categories. It could be that the dataset was imbalanced, i.e., some categories had much more data than others. Future studies could look into techniques for handling imbalanced data.

7. References

[1] J. M. Leimeister, "Collective intelligence," *Business & Information Systems Engineering*, vol. 2, 2010.

[2] E. Adamopoulou and L. Moussiades, "An overview of chatbot technology," in *Artificial Intelligence Applications and Innovations*, I. Maglogiannis, L. Iliadis, and E. Pimenidis, Eds. Cham: Springer International Publishing, 2020, pp. 373–383.

[3] H. B. Barlow, "Unsupervised learning," *Neural computation*, vol. 1, no. 3, pp. 295–311, 1989.

[4] Z. Cao, X. Li, Y. Feng, S. Chen, C. Xia, and L. Zhao, "Contrastnet: Unsupervised feature learning by autoencoder and prototypical contrastive learning for hyperspectral imagery classification," *Neurocomputing*, vol. 460, pp. 71–83, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231221010493

[5] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, R. Tibshirani, and J. Friedman, "Overview of supervised learning," *The elements of statistical learning: Data mining, inference, and prediction*, pp. 9–41, 2009.

[6] C. Wang, G. Peng, and B. De Baets, "Deep feature fusion through adaptive discriminative metric learning for scene recognition," *Information Fusion*, vol. 63, pp. 1–12, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253520302682

[7] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang, "Language-agnostic bert sentence embedding," *arXiv preprint arXiv:2007.01852*, 2020.

[8] B. Lund and T. Wang, "Chatting about chatgpt: how may ai and gpt impact academia and libraries?" *Library Hi Tech News*, vol. 40, no. 3, pp. 26–29, 2023.

[9] P. P. Ray, "Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope," *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 121–154, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S266734522300024X

[10] I. Arroyo-Fernández, C.-F. Méndez-Cruz, G. Sierra, J.-M. Torres-Moreno, and G. Sidorov, "Unsupervised sentence representations as word information series: Revisiting tf–idf," *Computer Speech &Language*, vol. 56, pp. 107–129, 2019.

[11] I. Arroyo-Fernández, C.-F. Méndez-Cruz, G. Sierra, J.-M. Torres-Moreno, and G. Sidorov, "Unsupervised sentence representations as word information series: Revisiting tf–idf," *Computer Speech & Language*, vol. 56, pp. 107–129, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0885230817302887

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[13] A. Mohammed and R. Kora, "An effective ensemble deep learning framework for text classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, Part A, pp. 8825–8837, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157821003013

[14] N. D. Mutizwa-Mangiza, B. C. Arimah, I. Jensen, E. A. Yemeru, and M. K. Kinyanjui, "Global report on human settlements - cities and climate change," *'Earthscan'*, 2011.

[15] M. Sepe, "Planning and place in the city: Mapping place identity," *Planning and Place in the City: Mapping Place Identity*, pp. 1–333, 03 2013.

[16] A. O. Richard Florida, "The work from home revolution and the future of cities," *The Wall Street Journal*, 05 2021.

[17] U. E. Chigbu, "Tenure-responsive land use planning a practical guide for country-level intervention," *UN-Habitat*, pp. 1–82, 2021.

[18] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang, "Language-agnostic bert sentence embedding," 2022.

[19] B. Lavine and T. Blank, "3.18 - feed-forward neural networks," in *Comprehensive Chemometrics*, S. D. Brown, R. Tauler, and B. Walczak, Eds. Oxford: Elsevier, 2009, pp. 571–586. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780444527011000260

[20] C. Park, S. Jeong, and J. Kim, "Admit: Improving ner in automotive domain with domain adversarial training and multi-task learning," *Expert Systems with Applications*, vol. 225, p. 120007, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423005092

[21] A. Jiao, "An intelligent chatbot system based on entity extraction using rasa nlu and neural network," *Journal of Physics*, vol. 1487, 2020.

[22] Y. Mohammadi, F. Ghasemian, J. Varshosaz, and M. Sattari, "Classifying referring/non-referring adr in biomedical text using deep learning," *Informatics in Medicine Unlocked*, vol. 39, p. 101246, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352914823000886

[23] K. Yalcin, I. Cicekli, and G. Ercan, "An external plagiarism detection system based on part-of-speech (pos) tag n-grams and word embedding," *Expert Systems with Applications*, vol. 197, p. 116677, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422001610

[24] L. R. Lukas Tommy, Chandra Kirana, "The combination of natural language processing and entity extraction for academic chatbot," *CITSM*, 2020.

[25] C. Zhang, P. Mayr, W. Lu, and Y. Zhang, "Knowledge entity extraction and text mining in the era of big data," *Data and Information Management*, 2021.

[26] X. Kong, X. Liu, B. Jedari, M. Li, L. Wan, and F. Xia, "Mobile crowdsourcing in smart cities: Technologies, applications, and future challenges," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8095–8113, 2019.

[27] J. Mueller, H. Lu, A. Chirkin, B. Klein, and G. Schmitt, "Citizen design science: A strategy for crowd-creative urban design," *Cities*, vol. 72, pp. 181–188, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0264275117304365

[28] JingWei, SungdongKim, HyunhoonJung, and Young-HoKim, "Leveraging large language models to power chatbots for collecting user self-reported data," *"*, 2023.

[29] M. V. Dominika Krasňanská, Silvia Komara, "Keyword categorization using statistical methods," *TEM Journal*, vol. 10, p. 1377-1384, 2021.

[30] B. Das and S. Chakraborty, "An improved text sentiment classification model using tf-idf and next word negation," 2018.

[31] Y. Tang, "Deep learning using linear support vector machines," *arXiv preprint arXiv:1306.0239*, 2013.

[32] M. A. Rashid and H. Amirkhani, "Improving edit-based unsupervised sentence simplification using fine-tuned bert," *Pattern Recognition Letters*, vol. 166, pp. 112–118, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865523000168

[33] K. Kaur and P. Kaur, "Improving bert model for requirements classification by bidirectional lstm-cnn deep model," *Computers and Electrical Engineering*, vol. 108, p. 108699, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790623001234

[34] A. Conneau and D. Kiela, "Senteval: An evaluation toolkit for universal sentence representations," *arXiv preprint arXiv:1803.05449*, 2018.

[35] H. Li, W. Wang, Z. Liu, Y. Niu, H. Wang, S. Zhao, Y. Liao, W. Yang, and X. Liu, "A novel locality-sensitive hashing relational graph matching network for semantic textual similarity measurement," *Expert Systems with Applications*, vol. 207, p. 117832, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422010910

[36] Y. Zhang, R. He, Z. Liu, K. H. Lim, and L. Bing, "An unsupervised sentence embedding method by mutual information maximization," *ACL Anthology*, 2021.

[37] J. Bos, V. Basile, K. Evang, N. J. Venhuizen, and J. Bjerva, "The groningen meaning bank," *Handbook of linguistic annotation*, pp. 463–496, 2017.

[38] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview," 2020.

[39] J. Read and F. Perez-Cruz, "Deep learning for multi-label classification," 2014.

[40] A. F. Agarap, "Deep learning using rectified linear units (relu)," *CoRR*, vol. abs/1803.08375, 2018. [Online]. Available: http://arxiv.org/abs/1803.08375

[41] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," 2022.

[42] M. Vakili, M. Ghamsari, and M. Rezaei, "Performance analysis and comparison of machine and deep learning algorithms for iot data classification," 2020.

[43] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling," *IEEE Access*, vol. 8, pp. 4806–4813, 2020.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[45] X. Song, A. Salcianu, Y. Song, D. Dopson, and D. Zhou, "Fast WordPiece tokenization," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 2089–2103. [Online]. Available: https://aclanthology.org/2021.emnlp-main.160

[46] M. Yang, M. K. Lim, Y. Qu, X. Li, and D. Ni, "Deep neural networks with l1 and l2 regularization for high dimensional corporate credit risk prediction," *Expert Systems with Applications*, vol. 213, p. 118873, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422018917