# Workflow

## Documents are gathered and categorized

The PDF documents were downloaded from the Prevention Web and stored on the local machine. There are 528 documents in total ready for this research.

Each document within the Prevention Web database is categorized under at least one “theme” (e.g., governance). The downloaded documents were categorized and analyzed according to these themes for processing efficiency and providing detailed results. However, the documents’ category is based on the domain knowledge rather than solely on the database’s decision.

The categories involve Health and health facilities[[1]](#footnote-1) (108 documents), Socioeconomic impacts and resilience[[2]](#footnote-2) (105 documents), Urban risk and planning[[3]](#footnote-3) (79 documents), Risk identification and assessment[[4]](#footnote-4) (115 documents), and Disaster risk management [[5]](#footnote-5)(121 documents).

Each preprocessing, training, and evaluation is conducted specifically for each category.

## Texts are gathered from PDF documents

The documents are in different formats (reports, research articles, discussion papers, policy briefs, policy recommendations, guidelines, reviews, technical notes, briefings, etc.). Therefore, they are unstructured, which means that their format (text font & alignment, existence of pictures & alignment, etc.) differ. They are all in PDF format and in English[[6]](#footnote-6). However, some of the documents in this dataset are corrupt, cannot be downloaded from the website, cannot be found on the web, or are unrelated.

“PyMuPDF” library was used to get textual data from the documents.

## Preprocessing is conducted

Preprocessing was the most tedious and problematic part of this study.

First, “WordNet POS” tags for lemmatization were loaded. Then, a preprocessing pipeline was defined.

The preprocessing pipeline involves,

* Removing multiple white spaces and line breaks
* Applying “nltk.word\_tokenize” on clean text
* Loading NLTK stopword list and defining a custom stopword list[[7]](#footnote-7)
* Standardizing the text and removing non-alphabetic or stop words
* Defining lemmatizer
* Conducting POS tagging
* Lemmatizing word-tokens via assigned POS tags

An iteration function was defined afterward. The purpose of the iteration function is to go over each page of each document to confirm that they are in the desired format (PDF), apply the preprocessing pipeline, and store tokens generated for each file in a dictionary. The iteration function involves,

* Creating an empty dictionary to display all documents and tokens together
* Defining a counter to track the number of processed texts
* Defining a for loop to iterate over each file to
* check whether the files are PDFs,
* create a path to the PDF files, extract the content of files,
* apply the preprocessing pipeline,
* generate a unique name for each document[[8]](#footnote-8),
* assign the preprocessed text content of the current document in the docs dictionary with the document name as the key
* Defining the pdf folder path
* Printing each document's tokens as a list

## Token dictionary is defined

Defining a token dictionary is important for preparing tokens for Bag of Words (BoW) representation and related applications by assigning each token a unique ID and collecting them under a single variable. The gensim library was utilized for this purpose.

## BoW representation is defined

BoW representation is important for converting tokens into numerical vectors, where each dimension corresponds to a unique token in the dictionary. It retrieves the frequency of each token in a document. This step makes the data ready for TF-IDF and n-grams applications.

## TF-IDF model is defined

TF-IDF is a model that captures the importance of tokens by weighing the frequency of a token (TF) against its frequency in the entire corpus (IDF). Tokens that appear frequently in a specific document but rarely in other documents are considered more important. Although TF-IDF is mainly used as input for LDA, it can be useful for providing inputs for n-gram analysis. Moreover, TF-IDF can be useful for identifying stop words.

Identifying stop words with TF-IDF can be tricky. It basically involves defining a probability threshold (0.05) and defining the words that are below this threshold as stop words. This method was not used fearing that there would not be many words to work with and certain domain-specific words would be removed.

## N-grams model is defined

N-grams is a model that involves a collection of words that appear together according to their frequency of co-occurrence. Therefore, n-grams is important for examining the context in which the tokens appear. For this study, trigram (with 3 words) n-gram was used because it can provide a better idea regarding the context.

## BERT topic model is defined

As a statistical model, LDA fails to identify the contextual relationships between the tokens. It merely identifies topics based on word distributions, which may fail to capture coherent topics generated from small and technical datasets. That is why, BERT topic was used.

BERT topic is a model based on transformer architecture that utilizes embeddings. Within the scope of this study, the BERT topic with "all-MiniLM-L6-v2" sentence transformer was used to create embeddings that help generate meaningful topics.

The BERT topic model was applied to the preprocessed text generated after applying the preprocessing pipeline and iteration function. After loading the sentence transformer model and encoding the preprocessed texts into embeddings, the parameters of the model were defined and adjusted. The parameters involve:

* min\_topic\_size[[9]](#footnote-9)
* top\_n\_words[[10]](#footnote-10)
* n\_gram\_range[[11]](#footnote-11)
* calculate\_probabilities[[12]](#footnote-12)

After adjusting the parameters, topics were displayed. The adjustment step is an iterative process depending on the comparison of domain knowledge with coherence scores.

## Coherence scores are calculated

The coherence score is a measure used to evaluate the degree of similarity between words within the same topic. A higher coherence score indicates that the words in the same topic are more closely related, which indicates that the topic is more meaningful and interpretable. The coherence score is important for evaluating to what degree topics and the words they involve are interpretable by a domain expert and identifying words that are likely to be relevant to the themes present in the corpus.

The “gensim.models.coherencemodel” was used to calculate topic coherence scores.

1. Involves documents related to identifying public health risks and ways to prevent them. [↑](#footnote-ref-1)
2. Involves documents related to the socioeconomic (e.g. economy, education, etc.) impacts, vulnerable groups, and ways to increase resilience. [↑](#footnote-ref-2)
3. Involves documents related to mainly infrastructural impacts in urban areas and ways to alleviate them. [↑](#footnote-ref-3)
4. Involves documents that identify & measure multiple risks. [↑](#footnote-ref-4)
5. Involves documents that identify risks and propose solutions to mitigate and adapt risks. May overlap with Risk identification and assessment category. [↑](#footnote-ref-5)
6. Some of the documents contain foreign words and characters mainly because of the location & author names as well as special terminology used. [↑](#footnote-ref-6)
7. [↑](#footnote-ref-7)
8. [↑](#footnote-ref-8)
9. Defines the minimum number of documents required for a topic to be considered meaningful. [↑](#footnote-ref-9)
10. Defines the number of top words to be extracted for each topic. [↑](#footnote-ref-10)
11. Defines the range of n-grams to consider when generating features from the text data. This is important for including words with multiple components. [↑](#footnote-ref-11)
12. Defines the probabilities for each topic, which is always set to “True”. [↑](#footnote-ref-12)