Available online at www.sciencedirect.com

ScienceDirect

Procedia Structural Integrity 44 (2023) 1720–1727

Structural Integrity Procedia

www.elsevier.com/locate/procedia

XIX ANIDIS Conference, Seismic Engineering in Italy

Enhancing Natural-Hazard Exposure Modeling Using Natural Language Processing: a Case-Study for Maltese Planning Applications

Justin Schembri*[a], Roberto Gentile[a], Carmine Galasso[b,c]

[a]*Institute for Risk and Disaster Reduction, University College London, London, UK*
[b]*Department of Civil, Environmental, and Geomatic Engineering, University College London, London, UK*
[c]*Scuola Universitaria Superiore IUSS Pavia, Pavia, Italy*

Abstract

The algorithmic processing of written language for tools such as predictive text, sentiment analysis, and translation services has become commonplace. The segment of computer science concerned with the interpretation of human language, NLP (Natural Language Processing), is a versatile and fast-developing field. In this paper, NLP is deployed unconventionally to gather insights into a building's multi-hazard exposure characteristics consistent with the GED4ALL attributes. NLP is used in this study to "read" the contents of digitally-submitted planning applications made on the Maltese archipelago. Maltese architects/engineers submit a concise but detailed description of the proposed works on any given site as part of a planning process. It is suggested that valuable insights exist within this description that can assist in classifying buildings within the bounds of the GED4ALL taxonomy. NLP can be used to layer additional, building-by-building information onto existing exposure models based on more conventional data. Although the results of this study are preliminary, NLP may prove a valuable tool for enhancing exposure modeling for multi-hazard risk quantification and management.

Keywords: Natural-hazard modeling; natural language processing; text mining, exposure modeling.

* Corresponding author. *E-mail address: justin.schembri.20@alumni.ucl.ac.uk*

10.1016/j.prostr.2023.01.220

1. Introduction and Motivations

Natural Language Processing (NLP; e.g. Bird et al., 2009) is the field of computer science concerned with the interaction between computers and "natural" human language. NLP practitioners design a series of algorithmic steps that prepare text data for digital processing. The combined steps are commonly referred to as a *pipeline* (e.g., Wachsmuth, 2015) involving text normalization, vectorization, and numerical fitting by a machine learning (ML) model. The goal of NLP is to derive valuable insights from language.

In disaster risk reduction (DRR) research and practice, NLP deployments often seek to understand the social response to an emergency (e.g., Karimiziarani et al., 2022; Verma et al., 2011). Language has also been abstracted for early-warning purposes, converting communication between individuals into a form of real-time sensing (e.g., Cecilia et al., 2021). Recent work (e.g., Ma et al., 2021; Rodrigues et al., 2021) has manipulated unstructured data, *prima facie* unrelatable to hazard modeling, to understand susceptibility to geological hazards. In this study, we propose using NLP to enhance natural-hazard exposure modeling by abstracting building permits as data points. Building permits are a type of authorization that a government or other regulatory body must grant before the construction (sometimes even modification) of a building can occur. While the structure of the text contents of building permits varies geographically (see Table 1 for an example), it is fair to assume that the majority will include an address and a brief description of the proposal.

Specifically, this work attempts to "mine" multi-hazard exposure information from building permit project descriptions. We propose a three-phase methodology consisting of 1) a supervised machine learning (ML) classifier; 2) an unsupervised ML clusterer; and finally, 3) the design of a series of regular expressions (regex) to derive details of multi-hazard exposure from clustered, semantically similar building permits. Regex patterns are abstract search patterns used for searching for text and are efficiently deployed on structured data, but require additional effort when the data is unstructured and noisy (e.g., Babbar and Singh, 2010), as is likely the case with our building permit dataset.

The classifier is used as a noise removal tool, identifying planning permits that may contain multi-hazard exposure information. Subsequently, the clusterer will cluster the classified data into linguistically similar groups. These groups will be the framework within which regex patterns are designed.

Our work analyzes a *corpus* (i.e., a collection of text documents) of around 100,000 publicly available ("Planning Authority - Advanced Search Facility," 2022) planning applications submitted to the Malta Planning Authority between 2005 and 2021.

Table 1: Basic data entries for Maltese and Londoner building permits. Both contain an address and a proposal description.

Data	Malta	London
File Ref Format::	PA/0001/20	2022/0212/S2
Address:	Yes	Yes
Sample Description:	*Proposed internal and external alterations including replacement of apertures, excavation at ground floor, demolishing of an existing washroom, construction of washroom, and proposed PV panels*	*Demolition of the existing buildings and erection of five blocks ranging from one to eight stories to provide 209 residential units (Use Class C3) together with 1,190sq.m of floor space at ground floor level, comprising; up to 1,190sq.m (Use Class E); at least 186sq.m (Convenience Store - Use Class E(a)); up to 176sq.m (Hot Food Takeaway - Sui Generis) …*

2. Methodology

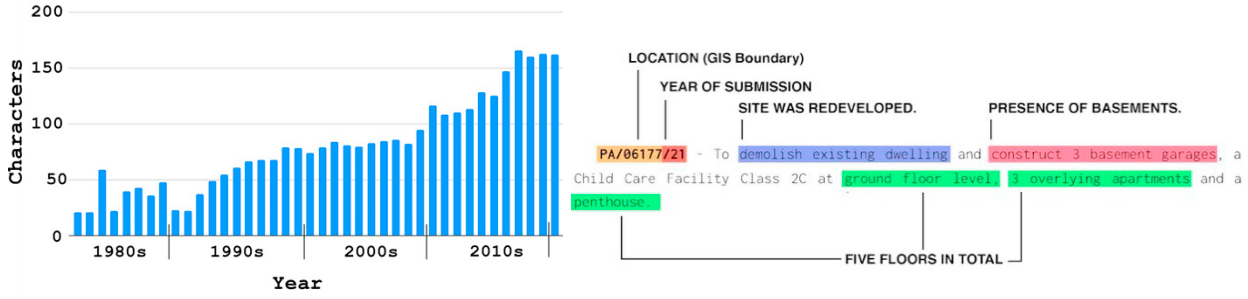
2.1. Understanding the Text Domain: the case-study text corpus

NLP models are specific to the kinds of text (i.e., text *domain*) they have been trained on (e.g., movie reviews). Basic corpus statistics such as those shown in Table 2 are helpful for domain familiarization. We expect the semantic quality of building permits to be similar amongst themselves, as architects/engineers write these applications and professionals tend to share language. The dataset covers a 17-year range of planning applications. We note a trend toward descriptions becoming more verbose with time (see Fig. 1a); since we intend to derive natural-hazard exposure

insights, we must understand what the text might offer (see Fig. 1b). From a sample readthrough of the dataset, we suggest that permits describing *floor additions to existing buildings* and the *construction of new buildings* are relevant to natural-hazard risk modeling. The data is unstructured, but there is a strong linguistic similarity, thus encouraging an NLP application. Finally, data exploration suggests useful phrases may be bundled together with substantial noise.

Table 2. Sample statistics of the corpus of Maltese building permits

Characteristic	Value			
Corpus Length	100,989			
Data Time Range	2007 to 2021			
Mean Document Word Count	18.6 words			
Most Common Words (excl. stop-words)	*Floor*	*Existing*	*Alterations*	*Level*



(a) (b)

Fig. 1. (a) Average character count for different year subgroups. (b) An example of the natural-hazard exposure characteristics the text may offer.

2.2. Methodology Overview

The potential insights identified in Section 2.1 may be correlated to natural-hazard-exposure attributes, such as those in the GED4ALL's taxonomy (Silva et al., 2018), as shown in Table 3. The list of potential insights offered is not comprehensive even to this dataset, and other datasets may suggest other text mining possibilities. Nonetheless, the proposed tentative methodology is flexible for applications to other datasets and/or other exposure attributes.

Guided by the nature of the dataset (and the preliminary tagging of a small corpus sample), we propose a methodology in three phases (see Fig. 2). The tagged dataset is first used to create a supervised ML classifier (Section 2.3). Next, the classified text is clustered (i.e., unsupervised ML) into semantically similar clusters (Section 2.4). In subsequent sections, we demonstrate how several multi-hazard attributes may be embedded into a single planning application. For the purpose of this research, an example regex is designed to capture one class of usable insights: a building's year of construction (Section 2.5).

Table 3. Potential insights offered by text classes correlated with the GED4ALL Taxonomy.

Textual Insight	Related GED4ALL attributes
Class 1: Addition of Floors	building:levels=*
Class 2: Construction of New Buildings	building:levels=*
	building:age=*
	building:levels:underground=*

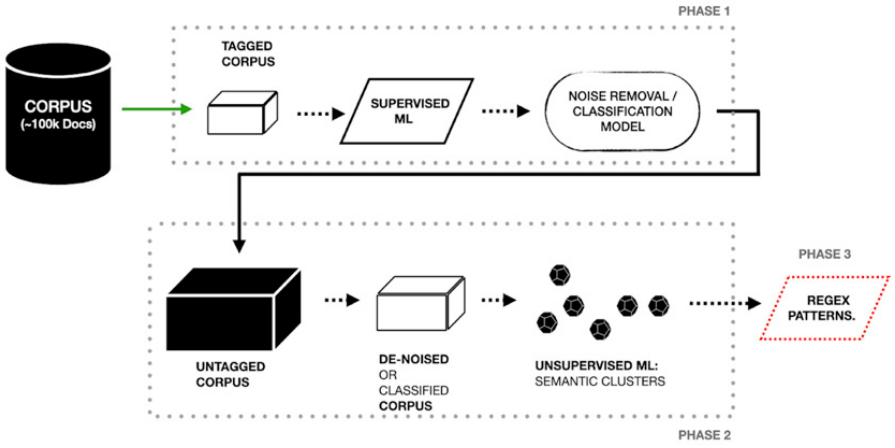


Fig. 2. Proposed NLP methodology, describing 1) classifying data into potentially useful classes; 2) clustering de-noised data into semantically similar groups; 3) deploying Regex to capture multi-hazard exposure attributes

2.3. Phase 1: Classification

Our initial task involved tagging a subgroup (~3,000 permits) into three categories: *Class 0: noise*; *Class 1: additional floors*; *Class 2: new builds*. The tagged subset of building permits is fed into an NLP pipeline (i.e., a sequence of normalization, vectorization, model fitting, and predictive steps). Each process in the pipeline may be run using different models, each with its own subset of parameters (called *hyperparameters)*. By way of example, vectorization may use a term frequency-inverse document frequency (TF-IDF) model (e.g., Aizawa, 2003) or a more context-aware model such as *Doc2Vec* (Le and Mikolov, 2014). Predictive model choices are, for example, Linear Support Vector Classification (LinearSVC) (e.g., Rosipal and Trejo, 2003) or Naive Bayes (e.g., Rish, 2001) estimators. Combinations of different models-parameters are tested iteratively (i.e., a so-called *grid search*). For example, one realization may normalize through lowercasing only, while another may also remove common words. Each realization of the pipeline undergoes a validation regime to test its performance. Validation involves splitting the tagged dataset into training/test sets. The model is then trained on the training data set and tested on the test data. The pipeline code is built using *Sklearn's* (Pedregosa et al., 2011) pipeline module.

While results will be explored fully in later sections, the supervised classifier performed best in noise detection (i.e., *Class 0*). Noise taint is inconvenient when designing regex patterns (i.e., Phase 3), as unintentional captures may lead to faulty conclusions. Therefore, the classifier is used to generate a new subset of de-noised data to be fed in the next phase of the methodology.

2.4. Phase 2: Clustering

This de-noised data is fed into an unsupervised clustering NLP pipeline. The parameters of the pipeline are identical to the supervised model, except that the final step is replaced with a K-Means clustering algorithm (e.g., Likas et al., 2003). The number of clusters (k) is a hyperparameter of the K-Means model, which is trained based on the dataset. One common method to set the hyperparameter is the elbow method (e.g., Nainggolan et al., 2019), which involves measuring a performance-loss metric of K-means for increasing values of k and selecting the value for which the performance loss is minimum (i.e., the elbow). For our dataset, no specific elbow is identified (i.e., the performance-loss function is monotonically decreasing). Therefore, our final selection is 45 clusters, based on manual interpretation of the cluster language, and manual trial-and-error involving the performance of the regex in Phase 3.

2.5. Phase 3: Capturing the Nuance with Regex

One cluster is selected to demonstrate the final step, i.e., regex writing. Regex design requires syntactical skill and familiarity with the insights one is attempting to gain. Our approach begins by plotting a word cloud for each cluster (Fig. 3d). The major theme of the selected example cluster is identified as "demolition and construction". The cluster contained texts associated with new buildings (in essence, capturing a segment of the Class 2: *new builds* category). Following the identification of keywords, and familiarization with the cluster, we carry out simple preprocessing (e.g., lowercasing, and the conversion of numbers to word equivalents) to simplify the regex writing.

The first item in the cluster is selected. The item is pasted into an online regex tool (Dib, n.d.), and a pattern is defined, which makes the correct capture. Our initial pattern is *"(demolition of).*(building).*(construction)"*. This pattern will only make a capture if "*demolition of*", *"building"* and "*construction"* appear in the phrase in that specific order and irrespective of the words between individual phrases/words (i.e., ".*" is a wildcard).

Once the first pattern is designed, another document is added to the editor. If the initial pattern does not capture the new phrase, a decision on whether to keep or modify the regex pattern is made. If a simple pattern alteration can capture both documents, such a modification is suggested. If the language of the two documents forks, the design of another regex pattern is suggested to capture the second document. After modifications of the pattern, the regex is applied to the entire cluster, counting the number of captures. The process is repeated for the first few documents in the cluster (e.g., the first ten documents) until a reliable set of regex patterns is obtained.

For the specific example cluster, including synonyms of the word "building" is enough to obtain a reliable pattern. This is done by adding an OR operator followed by the synonym. The revised pattern is *"(demolition of).*(building|premises).*(construction")*, with the word "*premises"* being a synonym for the word *"building*". The evolution of this specific regex pattern is presented in Fig. 4. The pattern evolved to allow for several synonyms and an operator (^), which only makes captures at the start of a string. The final pattern (pattern 1) is:

r"(^demolition of).(building|premises|structure|dwelling|property|existing|house).*(construct|proposed)*

(a)

Regex Iteration	1	r"(demolition of).*(building).*(construction)"	Match?
Text Reference	1	demolition of *dilapidated* building, *and* construction ~~of new residence.~~	✓

(b)

Regex Iteration	1	r"(demolition of).*(building).*(construction)"	Match?
Text Reference	1	demolition of *dilapidated* building, *and* construction ~~of new residence.~~	✓
	2	demolition of existing premises and reconstruciton of basement garage and overlying dwelling	×

(c)

	1	r"(demolition of).*(building).*(construction)"	
Regex Iteration	2	r"(demolition of).*(building|premises).*(construction)"	Match?
Text Reference	1	demolition of *dilapidated* building, *and* construction ~~of new residence.~~	✓
	2	demolition *of existing* premises *and re*construciton ~~of basement garage and overlying dwelling~~	✓

(d)

Fig. 3. Sample of the development of the regex pattern for cluster 0; (a) first text from the cluster is inputted, and a capture is made; (b) second text from the cluster is inputted, and a capture is not made due to the word "building" being absent; (c) original regex is modified and the second string is captured, (d) Word cloud for sample cluster, font size represents word frequency.

3. Results and Discussion

The predictive power of the classifier is measured by comparing the predictions made on the test data and the predicted text's class with the actual class (Table 4). The so-called precision and recall scores capture the model's ability to return true positives and true negatives. The classifier's performance is higher for *Class 0* (noise) than for the remaining two classes. This result motivates using this model as a filter, removing texts which offer no insights related to natural-hazard exposure modeling attributes. Deploying the trained classifier on an unseen dataset of 3,139 documents returned 895 documents (i.e., filtering out 71% of the documents regarded as noise). The 45 clusters produced by the clusterers generally show a similar size (a median of 16 documents, see Fig. 5a), except Cluster 12, which was relatively larger, suggesting a supercluster of permits with very similar linguistic and thematic content.

Cluster 0 is most carefully analyzed; however, a visual inspection of the remaining clusters suggested a potentially strong linguistic clustering and repetitive sentence structure.

The recorded model performance (both related to classification and clustering) is only valid for this particular application, and specific checks should be performed when using the same methodology for different datasets (e.g., analyzing an analog of Table 4).

Table 4. Supervised learning Classification Report showing performance of the best performing model. The selected model parameters are: Normalizer - *Basic*, Vectorizer - *TFIDF*, Max Features: *256*, N-gram Range: *(1,1)* Classifier - *Linear SVC*, Dual SVC: *True*, Date Range: *all*. Precision: true positive to the sum of true positive and false positives; Recall: true positives to sum of true positives and false negatives; F1 score: weighted combined precision/recall score, support: actual number of occurrences in the dataset.

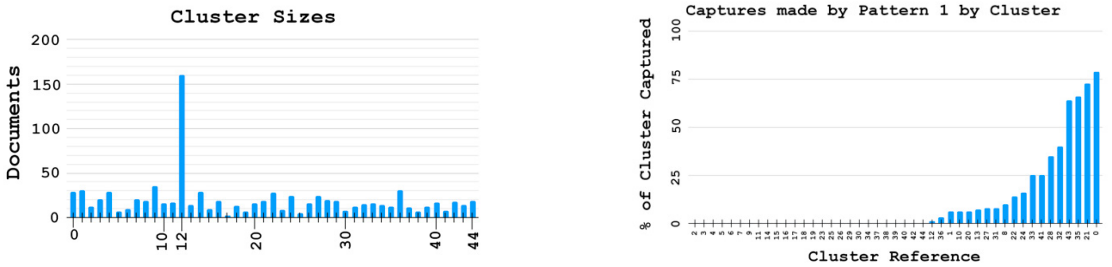
Tag	Precision	Recall	F1 Score	Support
Class 0 - *Noise*	0.900	0.945	0.922	805
Class 1 - *Extensions*	0.765	0.659	0.708	188
Class 2 - *New Builds*	0.899	0.831	0.864	183
Accuracy			0.882	1176

The classification-clustering sequence assists in writing a set of flexible regex patterns capable of deriving exposure attributes from individual pieces of text through three mechanisms. First, as demonstrated in Section 2.5, each cluster groups together documents with similar language styles, acting as a linguistic guide for the writing of regex. The cluster also allowed one to spot common synonyms (e.g., *dwelling, house, premises* are synonyms of *building),* which should be accounted for in the regex patterns (see Section 2.5). They also enable one to leverage language similarity.

Second, filtering noise from the dataset allows using slightly less strict regex patterns, which are: 1) easier to write; and 2) have higher precision. Without accurate noise filtering, one incorrectly-designed regex pattern could potentially capture many noisy documents, returning false positives that are hard to detect unless checked manually. Finally, since the noise removal quantifies how many useful data points exist in a corpus, we approximately know how many captures the regexes should be returning. For example, if 100 de-noised texts are returned, our suite of regexes should attempt to make around 100 unique captures.

While noise elimination is a valuable result, we suggest that pattern design should remain reasonably tight, i.e., try to capture a specific way of describing a building proposal. For example, there are several ways to describe a new building with four floors: e.g., "*to construct a four-story block*" as opposed to "*to construct shops at the ground floor, and three overlying floors above*". Two regex patterns should be drawn up for each sentence structure instead of one that tries to capture both.

Furthermore, patterns should ideally remain as specific to a given cluster as possible (i.e., return a low number of captures if used in other clusters). For example, the regex pattern designed for Cluster 0 made a large number of captures outside of its own cluster (see Fig 5b). This indicates that other clusters (e.g., 21, 35, 43) share linguistic similarities with Cluster 0. Although not pursued in this specific study, this kind of result may be used as a feedback loop for a re-definition of the clusters, complementing the adoption of the elbow method (Section 2.4).



(a) (b)

Fig 5 (a) Distribution of cluster sizes; (b) Performance of pattern 1, showing some cross-cluster capturing.

In the context of this work, regex patterns should be seen as pattern-logic pairs: i.e., a pattern is designed to be as linguistically specific as possible (at the cost of a lower rate of capture) and designed to tag a particular natural-hazard exposure attribute based on a given logical conclusion. For the pattern example in Section 2.5, we deduce that the specific planning applications captured by this pattern involve the demolition and construction of new buildings. From here, we correlate the year of the permit as being approximately equivalent to the year of construction, and we can assign it to the related building (since the permit is geolocated). Applying this pattern to the permit from Cluster 0 "*demolition of existing and proposed basement garage, 1 maisonette, 3 flats and washrooms*" identifies the related year of construction. Still, it does not, for example, identify the number of floors that the building is made up of (although such information is available in the text). We propose designing different patterns to capture such extra information.

We suggest that it is only technically feasible to design patterns that target a specific kind of phrasing and a specific exposure attribute. This allows pattern results to be layered onto multiple permits: e.g., one pattern identifies that a building permit refers to a new building, another identifies its overall height, and another identifies the presence of basements. The implication here is that cross-cluster regex patterns may need to be permissible, provided the pattern is specific and robust.

The application of the methodology on the full dataset (100,989 documents) returned 31,138 documents (filtering out 69% of noise data), and the designed regex pattern captures 3,437 documents (11% of the data not considered noise). The year of construction was extracted algorithmically from this resultant subset. The permits are also georeferenced and can be mapped to specific buildings. The distribution of new buildings by year (see Fig. 6) follows a construction boom in Malta motivated by a major change to planning policies in 2015, "*which encouraged developers to redevelop existing two-story dwellings into higher apartment blocks"* (Debono, 2018).

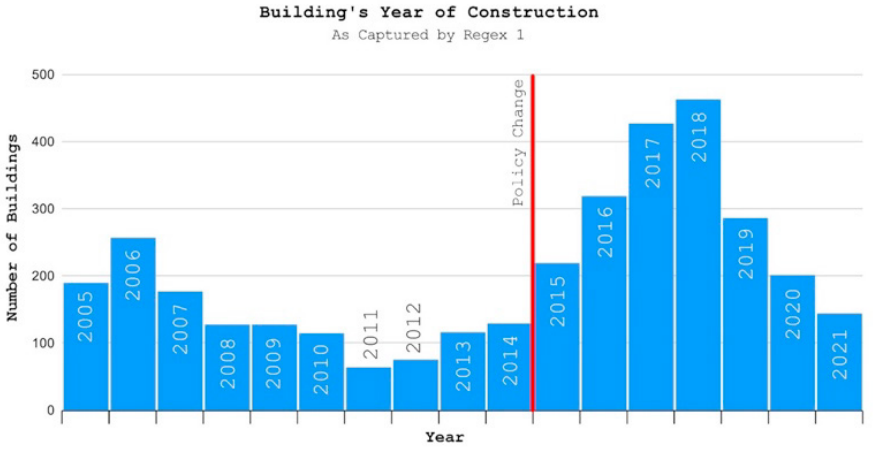


Fig. 6. Results from applying methodology on the full (100,389) corpus of documents. Number of new buildings constructed over the 2005 - 2021 date range in Malta.

4. Conclusions and Limitations

We have introduced a tentative methodology to mine natural-hazard exposure attributes using natural language processing (NLP) of building permits. The procedure involves first using a classifier (based on a small sample of tagged data) to distinguish relevant permit data (i.e., containing exposure information) from noise. Then, clustering (i.e., unsupervised machine learning) is used to group linguistically-similar data. The outcome of this process is used to write regular expression (regex) patterns to derive natural-hazard exposure attributes (according to the GED4ALL multi-hazard exposure taxonomy). The proposed methodology is applied to a corpus of digitally-submitted planning applications made on the Maltese archipelago. We show a regex pattern able to identify a building's year of construction. The proposed procedure appears promising since it allows automatically deriving building-by-building exposure attributes to be used in multi-hazard risk modeling. The results are summarized as follows:

- The classifier effectively filtered out noise from meaningful data. The precision and recall related to this task are equal to 0.900 and 0.945, respectively;
- The clustering is particularly useful because it does not reduce the linguistic complexity of the data. Rather, it organizes it, thus facilitating the subsequent phase of the methodology related to regex design;
- A regex pattern needs to be specific enough not to make false conclusions but flexible enough to make a reasonable amount of captures. To comply with this trade-off, we suggest interpreting regex as pattern-logic pairs: a pattern is designed to be as linguistically specific as possible (at the cost of a lower capture rate) and designed to tag a specific natural-hazard exposure attribute based on a given logic. We propose writing more concise patterns aiming at tagging one exposure parameter at a time;
- Regex writing for unstructured data is still relatively laborious, even with clustering as a guide. Therefore, a pattern writing methodology must be developed and/or more robust regex writing methodologies investigated.
- Realistically, other datasets will vary in their linguistic style. However, owing to the likelihood that the text is being generated within a very specific text domain, similarity in writing style can be leveraged to benefit the wider NLP process and possibly natural-hazard exposure modeling.

References

Aizawa, A., 2003. An information-theoretic perspective of tf–idf measures. Inf. Process. Manag. 39, 45–65. https://doi.org/10.1016/S0306-4573(02)00021-3

Babbar, R., Singh, N., 2010. Clustering based approach to learning regular expressions over large alphabet for noisy unstructured text, in: Proceedings of the Fourth Workshop on Analytics for Noisy Unstructured Text Data - AND '10. ACM Press, Toronto, ON, Canada, p. 43. https://doi.org/10.1145/1871840.1871848

Bird, S., Klein, E., Loper, E., 2009. Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. Beijing ; Cambridge Mass.

Cecilia, J.M., Cano, J.-C., Calafate, C.T., Manzoni, P., Periñán-Pascual, C., Arcas-Túnez, F., Muñoz-Ortega, A., 2021. WATERSensing: A Smart Warning System for Natural Disasters in Spain. IEEE Consum. Electron. Mag. 10, 89–96. https://doi.org/10.1109/MCE.2021.3063703

Debono, J., 2018. [ANALYSIS] Malta's building boom: how planning policies triggered a construction explosion [WWW Document]. MaltaToday.com.mt. URL http://www.maltatoday.com.mt/news/national/93061/analysis_maltas_building_boom_how_planning_policies_triggered_a_construction_explosion (accessed 7.28.22).

Karimiziarani, M., Jafarzadegan, K., Abbaszadeh, P., Shao, W., Moradkhani, H., 2022. Hazard risk awareness and disaster management: Extracting the information content of twitter data. Sustain. Cities Soc. 77, 103577. https://doi.org/10.1016/j.scs.2021.103577

Le, Q., Mikolov, T., 2014. Distributed Representations of Sentences and Documents, in: Proceedings of the 31st International Conference on Machine Learning. PMLR, pp. 1188–1196.

Likas, A., Vlassis, N., Verbeek, J.J., 2003. The global k-means clustering algorithm. Pattern Recognit. 11.

Ma, J., Rao, A., Silva, V., Liu, K., Wang, M., 2021. A township-level exposure model of residential buildings for mainland China. Nat. Hazards 108, 389–423. https://doi.org/10.1007/s11069-021-04689-7

Nainggolan, R., Perangin-angin, R., Simarmata, E., Tarigan, A.F., 2019. Improved the Performance of the K-Means Cluster Using the Sum of Squared Error (SSE) optimized by using the Elbow Method. J. Phys. Conf. Ser. 1361, 012015. https://doi.org/10.1088/1742-6596/1361/1/012015

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É., 2011. Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Planning Authority - Advanced Search Facility [WWW Document], 2022. . Plan. Auth. URL https://www.pa.org.mt/en/advanced-search-facility (accessed 7.26.22).

Rish, I., 2001. An Empirical Study of the Naïve Bayes Classifier. IJCAI 2001 Work Empir Methods Artif Intell 3.

Rodrigues, S.G., Silva, M.M., Alencar, M.H., 2021. A proposal for an approach to mapping susceptibility to landslides using natural language processing and machine learning. Landslides 18, 2515–2529. https://doi.org/10.1007/s10346-021-01643-3

Rosipal, R., Trejo, L.J., 2003. Kernel pls-svc for linear and nonlinear classification, in: In Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003. pp. 640–647.

Silva, V., Yepes-Estrada, C., Dabbeek, J., Martins, Luis, Brzev, S., 2018. GED4ALL - Global Exposure Database for Multi-Hazard Risk Analysis – Multi-Hazard Exposure Taxonomy, GEM Technical Report 2018-01. GEM Foundation, Pavia, Italy.

Verma, S., Vieweg, S., Corvey, W., Palen, L., Martin, J., Palmer, M., Schram, A., Anderson, K., 2011. Natural Language Processing to the Rescue? Extracting "Situational Awareness" Tweets During Mass Emergency. Proc. Int. AAAI Conf. Web Soc. Media 5, 385–392.

Wachsmuth, H., 2015. Text Analysis Pipelines, in: Wachsmuth, H. (Ed.), Text Analysis Pipelines: Towards Ad-Hoc Large-Scale Text Mining, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 19–53. https://doi.org/10.1007/978-3-319-25741-9_2