information

Article

Zero-Shot Topic Labeling for Hazard Classification

Andrea Rondinelli [1,*], Lorenzo Bongiovanni [2] and Valerio Basile [1]

[1] Department of Computer Science, University of Turin, 10149 Turin, Italy
[2] LINKS Foundation, 10138 Turin, Italy
* Correspondence: andrea.rondinelli967@gmail.com

Abstract: Topic classification is the task of mapping text onto a set of meaningful labels known beforehand. This scenario is very common both in academia and industry whenever there is the need of categorizing a big corpus of documents according to set custom labels. The standard supervised approach, however, requires thousands of documents to be manually labelled, and additional effort every time the label taxonomy changes. To obviate these downsides, we investigated the application of a zero-shot approach to topic classification. In this setting, a subset of these topics, or even all of them, is not seen at training time, challenging the model to classify corresponding examples using additional information. We first show how zero-shot classification can perform the topic-classification task without any supervision. Secondly, we build a novel hazard-detection dataset by manually selecting tweets gathered by LINKS Foundation for this task, where we demonstrate the effectivenes of our cost-free method on a real-world problem. The idea is to leverage a pre-trained text-embedder (MPNet) to map both text and topics into the same semantic vector space where they can be compared. We demonstrate that these semantic spaces are better aligned when their dimension is reduced, keeping only the most useful information. We investigated three different dimensionality reduction techniques, namely, linear projection, autoencoding and PCA. Using the macro F1-score as the standard metric, it was found that PCA is the best performing technique, recording improvements for each dataset in comparison with the performance on the baseline.

Keywords: zero-shot; topic labeling; hazard classification

Citation: Rondinelli, A.; Bongiovanni, L.; Basile, V. Zero-Shot Topic Labeling for Hazard Classification. *Information* **2022**, *13*, 444. https://doi.org/10.3390/info13100444

Academic Editor: David Martins de Matos

Received: 28 July 2022
Accepted: 16 September 2022
Published: 21 September 2022

1. Introduction

Nowadays, due to the massive amount of available data in various form (textual, visual, audio), one of the major challenge in machine learning is to use that data efficiently. While it is true that a lot of data can be useful, it is also true that often that data is unlabeled, unstructured and noisy. Moreover, using strictly supervised models in some domains could lead to performance drops. For instance, in the social-media domain, a continuous flow of new topics, trends, neologisms and linguistic adaptations are extremely common, making, thus, models trained over a certain dataset gathered in a specific timespan rapidly obsolete Florio et al. [1]. Zero-shot learning is a technique developed to cope with this broad range of problems: building a model capable of labeling data of unseen classes is a general approach for different classification tasks, leading to flexible, cheap and performing methods that can be adapted to various tasks and stay consistent across time and domains.

Zero-shot learning can be extremely helpful for different tasks, but needs to be implemented intelligently, whether using additional information about the classes to learn or to enrich data with more accurate descriptions or by inserting handcrafted useful connections between classes already learnt and classes yet to be seen. Each domain and task can provide different knowledge that can be leveraged to improve the model performance.

With the recent developments in the field of language models (such as BERT [2], XLNet [3], MPNet [4]), it is possible to represent textual data in the form of numerical vectors, allowing the use of similarity metrics based on geometrical distance, overcoming the sparseness problem encountered while working with raw textual data. In this context, zero-shot single- and multi-label classification for NLP tasks can be realized by converting text and labels to n-dimensional vectors with these pre-trained language models, and employing computationally cheap metrics such as cosine similarity. Since the task can be carried out by comparing each text vector with every label vector, the classification is performed by choosing the association that maximizes the similarity.

The strength of zero-shot learning, and in particular the implementation adopted for the task described in this paper, lies in its simplicity: with a few lines of code, no training and a straightforward (and computationally inexpensive) distance metric as a classification tool, promising results over a variety of different datasets can be achieved.

The rest of this article is structured as follows: after a brief review of the related works in Section 2, we present our methodology in Section 3, and in Section 4 we describe the benchmark datasets as well as the novel Twitter Hazard dataset we created. Finally, in Section 5, the various dimensionality reduction techniques tested are listed and described with the results obtained for the task.

2. Related Works

Ganzha [5] provides an extensive review of different state-of-the-art models compared in identical contexts, explaining in which aspects they differ and showing results on common benchmarks. The number of different approaches proposed in the literature is huge, and with many of them reaching very good performances; in Yin et al. [6] an extensive description of the problem with different solutions and benchmark results is given, while Xian et al. [7] presents a review of the current status of zero-Shot literature in computer vision, defining a proper protocol that can be used as a common baseline for each model developed with different techniques.

One of the pivotal points in NLP-related zero-shot classification tasks is data augmentation: usually, labels are not enough to describe meaningful semantic spaces, especially if their cardinality is high and their meaning can overlap or be noisy; in addition to this, the text to be classified can be noisy too, adding complexity to the task. In this scenario, augmenting the labels can be useful: building a context (manually or automatically) that can be used to refine the quality of the final embedding helps the process of the semantic alignment of labels and texts [8,9]. This augmentation can be carried out in various fashions, e.g., scraping Wikipedia descriptions, exploring the WordNet taxonomy, or navigating a CommonSense graph [10] to gather related content, words and meaningful relations that could help maximise the similarity between a context and a given example.

With the newborn urge of a "global" classifier that could adapt to the vast array of different topics that every day floods social media and a constant fresh supply of news in different forms and fashions, many authors theorized and implemented systems that could work with a power-law distribution of topics. Some trending tags are extremely common and easy to classify, even in a zero-shot context, while others are extremely rare and, therefore, a more challenging task to deal with [11]. These systems showed that neural models can be trained on a handful of (hugely populated) classes; meanwhile the vast majority of them could be inferred by a zero-shot model. The results are promising, since they guarantee high accuracy for previously seen classes and sufficient performances on less popular topics. The main goal of this paper falls under the definition of *semantic utterance classification*, previously discussed in Dauphin et al. [12], which described a novel method used to work in domains where none of the classes is known and no example is labelled. After gathering a huge amount of query click logs by linking the queries with the URLs clicked, a model is trained to learn a semantic space able to capture the sentence meaning: after this step, having a representation of texts and labels in an Euclidean space, the task is solved by means of a similarity metric. In Dauphin et al. [12], the semantic space is admittedly not always a set of well-separated clusters useful for the task, since the model does not learn how to discriminate between labels but only the relation between a label and a website. A refinement of the semantic space is proposed, by reducing the label overlap via conditional entropy minimization.

Ko and Seo [13] describe an unsupervised technique to train a model for unlabeled text classification: each label has a list of keywords attached, and every unlabeled document is split into sentences that are classified with the aid of those keywords. After processing the false positives, the model is trained with this partial information, showing good results.

Finally, on the topic of enhancing text classification via labels augmentation, Haj-Yahia et al. [14] explore human-based enrichment for a system that relies on cosine similarity between document embeddings and labels: again, the dimensionality-reduction techniques tested in this thesis could perfectly fit in this processing pipeline, since embedding lists of keywords and then reducing their dimensionality improves the classification performance, as shown by the results in this paper. In general, these models can be embedded into hybrid frameworks that use both supervised and unsupervised zero-shot techniques: by building good foundations on the data at the developer disposal, zero shot can help refine the results on the new classes and topics, keeping the cost of calibrating the supervised model for the topic distribution drifting low.

3. Methodology

Our plan is to leverage the language understanding of powerful transformer-based deep-language models (DLM) to perform text classification of a set of documents $\mathcal{D}$ into a set of labels $\mathcal{Y}$ purely based on the semantics of both, without any additional supervision.

This can be seen as a semantic text similarity (STS) task, with the additional complication that, in our case, the labels $\mathcal{Y}$ are usually single words or very short keywords, when standard STS tasks compare two context-full pieces of text (e.g., full sentences, paragraphs, documents). A DLM encoder θ, trained on a standard STS datset, learns a semantic linear space $V_{\mathcal{D}} \in \mathcal{R}^N$ where the semantics of long text d is well-represented by a vector $\theta : d \to v_d \in V_{\mathcal{D}}$. However, this semantic space $V_{\mathcal{D}}$ is not guaranteed to represent the semantics of labels $\theta : y \to v_y \in V_{\mathcal{D}}$ as well, since short keywords have not been seen explicitly during training. For this reason, we claim that direct comparison between v_d and v_y in $V_{\mathcal{D}}$ is sub-optimal, and our goal is to find a new linear space $V_{\mathcal{D}\mathcal{Y}}$ where the semantics of the vector representation of $\mathcal{D}$ and $\mathcal{Y}$ are better aligned.

For our experiments, we choose MPNet sentence encoder [4], θ_{MPNet}, pre-trained as a language model and fine-tuned by Hugging face for semantic text similarity (STS) on over a billion sentence pairs (https://huggingface.co/sentence-transformers/all-mpnet-base-v2 last accessed on 15 September 2022). Differently than BERT, MPNet has an autoregression architecture which, together with the masked and permuted language modeling paradigm it has been trained with, makes it ideal for learning sequences and places it as one of the top-performing pre-trained models on STS tasks (https://www.sbert.net/docs/pretrained_models.html last accessed on 15 September 2022).

To validate our hypothesis, we first consider the obvious baseline of simply using MPNet to encode $\mathcal{D}$ and $\mathcal{Y}$ directly in $V_{\mathcal{D}} \in \mathcal{R}^N$, where cosine similarity is used to assign the correct label y to each document d:

$$f : \arg\max_{y \in Y} cos(\theta_{\text{MPNet}}(d), \theta_{\text{MPNet}}(y)) \tag{1}$$

We, then, explore several methods to find a new linear semantic space $V_{\mathcal{D}\mathcal{Y}} \in \mathcal{R}^M$, typically with $M < N$, where the vector representation of long texts and labels are better aligned. This new space is found starting from $V_{\mathcal{D}}$ and learning a projection function $r : V_{\mathcal{D}} \to V_{\mathcal{D}\mathcal{Y}}$ where the classification function is:

$$f : \arg\max_{y \in Y} cos(r(\theta_{\text{MPNet}}(d)), r(\theta_{\text{MPNet}}(y))) \tag{2}$$

Graphically, the architecture of our model can be visualized in Figure 1.

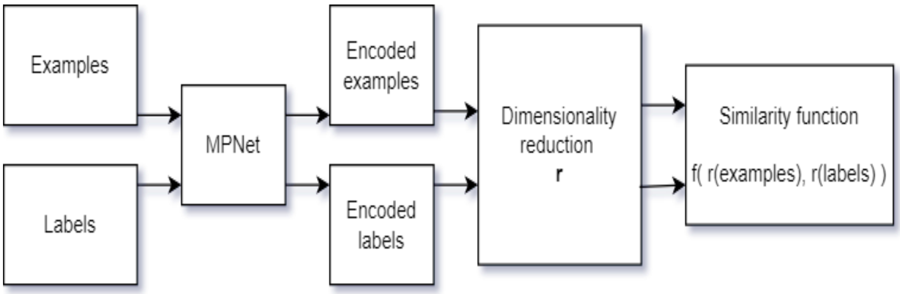

Figure 1. The architecture of the model is composed of a first step of encoding via the MPNet language model, then a dimensionality reduction technique is applied. Finally, the similarity function is used to obtain the predictions.

The principle behind our work is similar to Dauphin et al. [12], but with one major difference: the semantic space is not learnt by correlating queries and clicked URLs; in our proposed method, there is no training at all and the vector representation is obtained by applying a pre-trained language model with no further fine-tuning on it. For the refinement of the semantic space, a similar strategy to the one seen in Dauphin et al. [12] is implemented in our model by searching for the dimensionality reduction that maximizes the F1-score of the classification task on a test set, meaning that the resulting vectors will be more separated and the corresponding examples more cohesive.

In this paper, we implement a technique that uses lists of keywords describing different facets of the same label similar to the one seen in Ko and Seo [13]. Since the lists of keywords are used to gather the textual data from social media, a similar process of false-positive handling is performed and tested in the context of a topic-classification task in the hazard domain.

In this paper we explore three different methods to learn the projection function r, namely: PCA dimensionality reduction, linear projection onto a pre-trained word embeddings space, and projection onto a latent space generated by an autoencoder. Each one of these methods will be tested on the three benchmark datasets, discussed in Section 4, and the results compared against the baseline of the raw encoding without reduction. The best reduction model will then be applied to study the novel Twitter Hazard dataset.

3.1. PCA Dimensionality Reduction

Principal component analysis is one of the dimensionality reduction techniques tested to retain only the most significant dimensions. By most significant dimensions, we mean the values that most contribute to the classification task, minimizing the overall noise in the MPNet embedding vectors. Using the semantic embeddings of the examples obtained with MPNet, we then tried to find the best-fitting number of dimensions that could maximize the performance scores over the classification task proposed. The target semantic space was found by trying to find the best-fitting space on a training set, by iteratively applying the algorithm with an increasing amount of explainable variance to retain. The reduction found was then applied on a test set, which was the benchmark to see if the number of dimensions retained was the one that most improved the classification performance. Once we found the number of dimensions that gave the best results, we applied the transformation to a development set.

3.2. Linear Projection onto Word Embeddings

As we argued before, deep-language models (DLMs), and in the specific context of this paper, MPNet, are not optimal to encode one or two-word phrases such as the labels in our datasets, since they have been trained on context-full text. On the contrary, word embeddings are trained exactly to capture the semantic of atomic words better. A natural idea is, then, to use MPNet to capture the semantics of long text $\theta_{\text{MPNet}} : d \to v_d \in V_{\mathcal{D}}$ and word embeddings to capture the semantics of labels $\theta_{\text{WE}} : y \to v_y \in V_{\mathcal{WE}}$, with $V_{\mathcal{D}} \in \mathcal{R}^N$

and $V_{\mathcal{WE}} \in \mathcal{R}^M$. This is, however, not possible to implement straight away as the two semantic spaces $V_{\mathcal{D}}$ and $V_{\mathcal{WE}}$ are learned independently by the two models θ_{MPNet} and θ_{WE}, respectively, and, threfore, disaligned. To obviate this problem, we implemented a simple linear transformation $W \in \mathcal{R}^{N \times M}$, which has the task of learning an alignment between the two semantic spaces.

Training was performed by considering the top N vocabulary terms (only nouns), where N ranged from 50 to 200 thousand $\mathcal{X} = \{x_1, x_2, .., x_N\}$ of the word-embeddings model, taking the relative word embeddings $\mathcal{H}^{WE} = \{h_1^{we}, ..., h_N^{we}\}$ and the relative embeddings generated by MPNet $\mathcal{H}^{\text{MPNet}} = \{h_1^{mp}, ..., h_N^{mp}\}$ and learning the the linear transformation W such that:

$$W\mathcal{H}^{\text{MPNet}} \simeq \mathcal{H}^{WE} \tag{3}$$

minimizes the mean squared error (MSE).

We repeat this experiment with two different pre-trained and freely available word embeddings: FastText `crawl-300d-2M-subword` (https://fasttext.cc/docs/en/english-vectors.html last accessed on 15 September 2022), which contains two million words with the corresponding subwords making it robust against out-of-vocabulary terms [15], and Word2Vec `word2vec-google-news-300` (https://huggingface.co/fse/word2vec-google-news-300 last accessed on 15 September 2022) [16].

3.3. Autoencoder Projection on Latent Space

We adopted a shallow autoencoder, f_{ED}, trained on a number of sentences that ranged from tens to hundreds of thousand based on which dataset was in use, to learn the latent space $V_{\mathcal{DY}}$ starting from the semantic embeddings generated by MPNet. The architecture is very simple, formed by a single-layer encoder $f_E = (g \cdot W_E) \in \mathcal{R}^{N \times M} : V_{\mathcal{D}} \rightarrow V_{\mathcal{DY}}$ and a single-layer decoder $f_D = (g \cdot W_D) \in \mathcal{R}^{M \times N} : V_{\mathcal{DY}} \rightarrow V_{\mathcal{D}}$, so that the global autoencoder model can be described as

$$f_{ED} = (g \cdot W_D)(g \cdot W_E) \tag{4}$$

where $g\cdot$ is the element-wise application of a non-linearity function. The model learns to reconstruct the sentences, initially embedded by MPNet into $V_{\mathcal{D}}$, by minimizing the mean squared error between the input of the encoder f_E and the output of the decoder f_D. The number of dimensions of the latent space in between the two layers was one of the hyperparameters that was explored during the training process. After training is finished, the decoder f_D is discarded and the encoder f_E is used as the projection function r in Equation (2). This setup did not lead to better results than the precedent techniques.

Two different corpora were used to learn two different latent spaces: the *generalized* approach used data extracted from Wikipedia and embedded with MPNet. The autoencoder learns a latent embedding space that will be used for all the benchmarks already documented. This is purely to test how a generic architecture could perform when faced with a completely different statistical distribution of data.

An *ad-hoc* approach with autoencoders is also proposed, where the autoencoder was trained on the specific dataset of the task and uses it for dimensionality reduction on a test set from the same dataset. This approach aims at modeling the specific features of each dataset, and leveraging them for dimensionality reduction. This approach, since the architecture is built to learn the statistical distribution of the values of the single dataset and since the benchmarks are datasets widely different to each other, should lead to better results than a generalized approach.

4. Datasets

The approach described in this paper is based on the usage of the MPNet language model, freely available on the Huggingface repository (https://huggingface.co/models last accessed on 15 September 2022), and on the principal component analysis (PCA) algorithm. Before the application to the hazard detection domain, the method was tested on different benchmark datasets reviewed in this section: Yahoo Answers, DBPedia, Lexglue/Ledgar.

4.1. Benchmark Datasets

Yahoo Answers (https://huggingface.co/datasets/yahoo_answers_topics last accessed on 15 September 2022) dataset is a classic benchmark in topic-classification tasks, which consists of a set of question–answer pairs and a set of 10 labels that describe the categorization of the questions on the site. It is a very common dataset, since Yahoo Answers is a never-ending source of useful data, although they are often noisy and non-standardized, which means it can be transferred to a real-world domain such as social media and similar platforms. The fact that the text can be noisy, and rich in grammatical errors and slang words, can be used to test the ability of the language model to represent correctly the data and to see how well non-handcrafted definitions can be classified. The split available only considered 1.4 million examples for the training set and 60 thousand examples for the testing set; we then randomly split the training set into training and development sets with a 70%/30% ratio.

DBPedia (https://huggingface.co/datasets/dbpedia_14 last accessed on 15 September 2022) (Lehmann et al. [17]) dataset is a fairly well-known dataset which takes a huge number of entity definitions (defined as title–description pairs) and a set of 14 non-overlapping labels from DBPedia. One of the differences with Yahoo Answers is the nature of the labels: in Yahoo the labels described different topics of the questions which can be noisy or ambiguous; DBPedia instead uses labels that described what is a certain entity, for example an athlete, a film, a written work, or a natural place. Moreover, DBPedia uses a well-written definition, without mistakes and slang of any kind, minimizing the noise. With these premises, this is the dataset that was expected to give the best results, since MPNet should give more precise embedding representations. The split available for this dataset only considered 560 thousand examples for the training set and 70 thousand for the test set; similarly to the Yahoo Answer dataset, we created a development set by splitting the training test.

Lexglue/Ledgar (https://huggingface.co/datasets/lex_glue last accessed on 15 September 2022) (Tuggener et al. [18]) dataset is a lesser known dataset on the domain of contract provisions: documents taken from the publicly available sources of United States Securities and Exchange Commission (SEC). This dataset is relevant for its complexity: its 100 labels are often overlapping or semantically similar. The model was not expected to perform excellently on this domain, since it is so complex and ambiguous; still, it is interesting to see how it behaves when faced with formal writing and high semantic interference between labels, and if the techniques of enhancement can affect even difficult situation like this one. For this dataset, there were available training, validation and test sets, so it was not necessary to perform any further processing.

These three dataset were chosen to test how our zero-shot classification system could handle data that differed in various ways: lexicon, slang usage, grammatical errors, domain and number of labels. Subsequently, the system was applied to the hazard domain in Twitter, using different handcrafted tweet datasets.

4.2. Twitter Handcrafted Datasets

We created three different datasets from Twitter starting from data gathered by LINKS foundation in the period 2020–2021. Since, in this wide time span, different major catastrophes happened (e.g., COVID-19), along with the rise in concerns and worries about global warming and climate crisis-induced hazards, the distribution of examples for each class is highly unbalanced, i.e., the vast majority of tweets gathered talk about COVID-19. This includes not only casualty counts, reports and news, but also opinions about lockdown measures, vaccines and different COVID-19-related issues. The data was originally gathered by retrieving tweets if they contained certain keywords: each label has a set of corresponding keywords that describe different shades of a hazard. This approach was used to gather a huge number of relevant examples and create a dataset that could contain every different way to cite the given disasters. We can consider this dataset labeled in a distant supervised fashion, that is, automatically assigning the labels based on the keywords used for retrieving the items. In Table 1, those keyword sets can be seen, with their corresponding labels. The collection includes an extra category for non-informative tweets, which were not considered for further processing.

Table 1. Summary of the classes with their corresponding keywords in the dataset.

Hazard	Keywords
Extreme weather	heatwave, hot weather, hot summer, cold weather cold winter, extreme weather, extreme cold, extreme hot, hottest summer, hottest weather, coldest winter, coldest weather, drought
COVID-19	covid dead, covid deaths, covid infected, covid hospitalized, covid recovered, covid hospitals, covid cases, covid outbreak, covid-19, pandemic virus, virus dead, virus deaths, virus infected, virus hospitalized, virus recovered, virus hospitals, virus outbreak
Avalanche	avalanche, avalanches, icefall, icefalls, avalanche victims
Fire	forest fire, forest fires, wildfire, wildfires, bushfire, bushfires, conflagration, high flames, burned, explosion fire, firefighter, firefighters, fire fighters, fireman, firemen
Flood	flood, floods, flooding, floodings, flash flood, deluge, inundation, inundated, flood victims, flood affected, flood dead, flood missing, flood warnings, help flood, rescue flood
Earthquake	earthquake, earthquakes, seismic, magnitude, epicentre, epicenter, building collapsed, quake victims, earthquake dead, earthquake injured, help earthquake, missing earthquake
Storm	storm rain, storm rains, storm wind, storm winds, winter storm, summer storm, autumn storm, storm lightning, storm lightnings, severe storm, incoming storm, spring storm, cloud storm, storm clouds, eye storm, storms, heavy rain alert, heavy rains, lightnings, thunderstorm, thunderstorms, thunder storm, thunder storms, windstorm, windstorms, wind storm, wind storms, snowstorm, snow blizzard, blizzards, strong wind, hurricane, tornado, typhoon, rainfall, hurricane category
Terrorism	terrorist attack, terrorists attack, terrorist deaths, terrorist injured, terrorist hostages, terrorists dead, terrorist bomb, terrorism bomb, terrorism attack
Landslide	landslide mud, landslide rain, landslide buried, landslide kills, landslide erosion, mudslide, mudslides, mudflow, mudflows, debris fall
Subsidence	subsidence

As the original tweets were collected by simple keyword matching, they contain a lot of noise. In particular, two kind of mistakes are common, namely, type 1: the keyword is used out of the hazard context (e.g., ’that player is on fire’); and type 2: the Tweet mentions multiple hazards. To show that our method can alleviate these problems, we create three datasets:

1. Gold dataset: the tweet is effectively about the hazard associated to the keyword;
2. Keyword-out-of-context dataset: the tweet is not at all about a hazard and the keyword is just used with another meaning;
3. Multiple-keywords dataset: the tweet mentions multiple hazards but it has been only associated with the one keyword it was retrieved for.

Embedding a set of keywords resulted in non-accurate vectors: to overcome this problem, we split the categories and computed the embeddings for each keyword, averaging them to find the vector that could depict more accurately the meaning behind all the shades of the same hazard.

Since almost 50% of the tweets gathered were classified as COVID-19 hazard, a balancing of the class was made. Building a dataset with 1000 instances and 10 classes, we decided to assign almost evenly 100 examples per class, as can be seen in Table 2. An exception was made for the avalanche hazard, since it is also an NHL hockey team name and a crypto currency name and, in the gathered data, almost 99% of the tweets labeled with this keyword were completely unrelated to the natural hazard. Some tweets contains different hazards inside them: often, earthquakes are related to landslides, floods to storms and hurricanes, and an exceeding amount of tweets related to the Myanmar coup of 2021 are related to COVID-19 emergencies. Some tweets that contained these associations were kept inside the gold dataset, allowing study of how the model behaves in these examples.

Table 2. Class distribution in the Gold dataset.

Label	Support
Extreme weather	105
COVID-19	101
Avalanche	10
Fire	110
Flood	114
Earthquake	102
Storm	100
Terrorism	122
Landslide	124
Subsidence	112

To tackle the task of metaphors in tweets, a subset of 100 tweets that use a hazard keyword metaphorically was built. This is one of the main challenges posed by the starting data and one of the main reasons for pollution in the data: studying this problem singularly should help understand better the behaviour of PCA. Ideally, PCA should improve classification by increasing the accuracy in actual hazard tweets; it should decrease similarity between hazard labels and metaphorical references to the hazard; furthermore, in the case of multiple hazards, it should show more fitting similarities for a different number of labels. The goal of this subtask is, then, to explicitly show how our system is able to capture the semantics of the tweet and understand that the keyword is not used in the context of a hazard, therefore assigning a low semantic similarity score to it. This test was performed by taking every similarity value for the gold label for each example and using it to compute the average: by comparing the values pre and post the PCA computation, a decrease should be observed.

To tackle the problem of ambiguous tweets that reference multiple hazards, the tweets that contained references to at least two hazards were kept in a separate dataset, in order to avoid confusing the classifier.

5. Results

5.1. Benchmark Results

Table 3 shows the results of the methods introduced in the previous section tested on the benchmark datasets described in Section 1. PCA, as a method of dimensionality reduction, provided the best performance, pushing further beyond the expectations. Since PCA reorders dimensions based on the amount of explainable variance they carry, we can compute how much variance is needed to capture the most valuable dimensions that describe the semantic space for this task better. This approach is particularly flexible because we know that different data, for their intrinsic nature, could use a different number of dimensions. To find the optimal number of target dimensions, we tested the method with a range of values of explainable variance and kept the one that allowed to save the configurations that returned the highest F1-score. The expectations, confirmed by the experimental results, were that the most difficult dataset would require a higher amount of variance while the other datasets would require less. In terms of absolute performance, it is clear how PCA applied to the MPNet vectors outperforms every other technique tested.

Table 3. The F1-scores results obtained by each method on the three benchmarks tested. Bold face indicates the best performance.

Reduction Technique	Yahoo Answer	DBPedia	Lexglue/Ledgar
Baseline MPNet	0.520	0.648	0.227
LinearProjection (FastText)	0.343	0.286	0.067
LinearProjection (Word2Vec)	0.519	0.688	0.158
PCA	**0.580**	**0.795**	**0.279**
Generalized autoencoder	0.259	0.514	0.029
Ad-hoc autoencoder	0.223	0.298	0.113
Word2Vec autoencoder	0.283	0.299	0.042

5.2. Twitter Dataset Results

Here, we present the results of the PCA-based method on the Twitter dataset on hazard classification, summarized by the labels in Table 4.

Table 4. F1-scores @3 results obtained with PCA on the Twitter Gold standard Dataset.

Hazard	F1@3	Support
Extreme weather	0.934	105
COVID-19	0.966	101
Avalanche	1.0	10
Fire	0.982	110
Flood	0.927	114
Earthquake	0.971	102
Storm	0.960	100
Terrorism	0.976	122
Landslide	0.976	124
Subsidence	0.982	112
Macro-average	0.961	1000
Micro-average	0.957	1000

The evaluation is carried out my measuring the F1-score at 3, meaning that a prediction was considered correct if the gold label was in the top-3 most similar labels. This metric is used rather than a straightforward F1-score in order to tackle the problem of similarities between some of these labels, such as, for example, earthquake/subsidence or floods/storms/landslides, which are all labels that often occur together.

Table 5 shows the aggregated classification results and a comparison with the identical architecture without the PCA dimensionality reduction, in terms of precision recall and F1-score. A noticeable improvement provided by PCA is evident both with the metrics at 1 and at 3.

Table 5. Topic classification metrics at 1 and at 3 on the Gold dataset. Only the macro metrics are shown. Bold face indicates the best performance for each metric.

	Precision	Recall	F1-Score	Accuracy
MPNet@1	**0.822**	0.798	0.794	0.814
MPNet + PCA@1	0.805	**0.839**	**0.808**	**0.830**
MPNet@3	0.955	0.949	0.950	0.943
MPNet + PCA@3	**0.961**	**0.962**	**0.961**	**0.957**

5.3. Keyword-out-of-Context Dataset Results

While exploring the data, we noticed how a consistent source of classification errors is represented by the metaphorical use of some of the terms present in the labels, e.g., “flooding the market” in the financial context.

In Table 6, we present the results of a test conducted on a subset that contains only tweets that contained keywords used metaphorically. The aim of this experiment is to analyze the extent of the geometric transformation of the embedding space caused by the PCA dimensionality reduction. Since MPNet embeds the meaning of the sentence, the similarity between the single keyword and the tweet in which it is used metaphorically is expected to be low. The experiment was conducted by averaging the similarities between the gold labels and the tweets in the Gold dataset and the similarities between the assigned labels and the tweets in the Keyword-out-of-context dataset. What is encouraging is that the similarity is still very low, meaning that the model is still very aware of the meaning of these text–label couples. As can be seen, PCA reduces this similarity in the averaged setting, while it keeps it almost identical in the normal setting: the difference is not that relevant and probably a larger dataset could highlight significant changes in the results; it is, however, an interesting result that could lead to further studies and more in-depth works oriented toward analysis of the hazard labelling task.

Table 6. Average gold label similarities pre and post PCA on the two datasets.

	Keyword-out-of-Context Dataset	**Gold Dataset**
MPNet	0.220	0.416
MPNet + PCA	0.175	0.326

5.4. Multiple-Keywords Dataset Results

This dataset, as already said, was extracted by retaining 100 tweets that contained at least two keywords inside of them. Since it was observed that some hazard categories are strictly related, some of them co-occur often inside of a single tweet; by automatic labeling with a single keyword, the classifier could get wrong predictions based on the presence of another hazard in the text. These tweets are very common, since the nature of the hazards is extremely similar and they often appear together. We then considered interesting the idea to build this dataset and perform a minor experiment that consisted of comparing the similarities of a tweet with each label before and after the dimensionality reduction: if our system works as intended, after the reduction, only the most relevant labels should have good values.

As an example of a tweet that contains multiple hazards, consider the following tweet:

> “heavy rainfall from #grace will result in significant flash and urban flooding as well as mudslides.high surf generated by grace will affect the southern gulf of mexico coastline through the weekend.”

The gold label assigned to this example refers to the word “mudslides” which is not the focal point of the tweet: it is, however, a hazard related to the heavy rainfall and possible flooding depicted earlier in the text. It is not a wrong association, but if the model labeled it with the “Hazard: storm” or “Hazard: flood” labels, it would be considered wrong from a classification point of view. Isolating a subset of tweets that express this phenomenon and manually reviewing the behaviour of the system could further help explain the results.

Keeping in mind the example tweet just shown above, those shown in Table 7 are the similarity values pre and post PCA.

Table 7. Cosine similarities for each label, pre and post PCA application for the Multiple-keywords dataset.

Hazard	MPNet	PCA
Extreme weather	0.22	−0.14
COVID-19	0.16	−0.06
Avalanche	0.14	−0.03
Fire	0.13	−0.25
Flood	0.49	0.38
Earthquake	0.35	0.15
Storm	0.42	0.36
Terrorism	0.12	−0.25
Landslide	0.37	0.26
Subsidence	0.25	0.05

What is interesting about this table is that, after the PCA application, each and every similarity experiences a decrease: the top-three classes are the same both pre and post PCA, and, even if they experience a decrease in similarity, the net difference is substantially lower than the net differences for other classes. In fact, while the decrease for these three classes ranges from 6 to 11, the decrease for every other class ranges from 17 to 38, leading most of the similarities to negative values.

These results reinforce the idea that dimensionality reduction is an effective solution to improve semantic spaces' alignment: this does not necessarily mean that the classification will be corrected after the PCA approach but shows that the results are not obtained by chance; furthermore, it goes to show the multiple possible cases that can occur in the Twitter domain and how it can impact the complexity of the task from the nature of the data, the cost of manually annotating and checking it, and the multiple possible strategies to obtain solid results.

6. Conclusions and Future Work

The results showed that the initial hypothesis was right: in fact, lower dimensionality spaces carved out from sentence embedding vectors are more representative of the sentence–topic-aligned semantics, even for single words or 2–3 words phrases. The technique that performed the best reduction is PCA, but this does not mean that another dimensionality reduction that performs even better cannot be found. Therefore, this work lays the foundations for possible new approaches to NLP tasks that involve embedding vectors. The results presented in this paper uncover possible new paths in zero-shot classification: if the task is treated as a semantic clustering problem in which the center of each cluster is the label, it is possible to improve the position of the examples in the space to better fit its cluster by just performing the dimensionality reduction. Since the task is not trivial, it can be said that the strength of this system relies on its simplicity: this transformation is easy to perform, completely automatic and unsupervised.

The approach presented in this paper is simple and effective, but comes with the implication that the reason behind a certain classification could be oblivious to the final user. When faced with the problem of human-understandable AI and NLP, a comprehensible explanation of why a document has been classified under a certain tag could be needed: in the related-works section we saw how an approach that used the extraction of ConceptNet relations used to augment the semantic space will also provide, with a natural-language-generation technique, a human-understandable explanation of the correlation between topic and text. This is just one of the possibilities in this direction and, in this paper, none of these techniques were used, since the focus revolved around the dimensionality reduction of the embedding vectors. However, these approaches could coexist, using the dimensionality reduction as the foundation of more complex and robust classification systems.

Author Contributions: Conceptualization, L.B.; Data curation, L.B.; Formal analysis, L.B.; Methodology, A.R. and V.B.; Resources, A.R.; Software, A.R.; Supervision, L.B.; Validation, V.B.; Writing—original draft, A.R.; Writing—review & editing, A.R., L.B. and V.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The three Twitter datasets generated during the study are freely available at the following URL: https://osf.io/8c3nf/ last accessed on 15 September 2022.

Acknowledgments: Thanks to Links Foundation (https://linksfoundation.com/ last accessed on 15 September 2022) for the provided data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Florio, K.; Basile, V.; Polignano, M.; Basile, P.; Patti, V. Time of your hate: The challenge of time in hate speech detection on social media. *Appl. Sci.* **2020**, *10*, 4180. [CrossRef]
2. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
3. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 5753–5763.
4. Song, K.; Tan, X.; Qin, T.; Lu, J.; Liu, T.Y. Mpnet: Masked and permuted pre-training for language understanding. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 16857–16867.
5. Ganzha, M. Practical Aspects of Zero-Shot Learning. *arXiv* **2022**, arXiv:2203.15158.
6. Yin, W.; Hay, J.; Roth, D. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv* **2019**, arXiv:1909.00161.
7. Xian, Y.; Schiele, B.; Akata, Z. Zero-shot learning-the good, the bad and the ugly. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4582–4591.
8. Zhang, J.; Lertvittayakumjorn, P.; Guo, Y. Integrating semantic knowledge to tackle zero-shot text classification. *arXiv* **2019**, arXiv:1903.12626.
9. Halder, K.; Akbik, A.; Krapac, J.; Vollgraf, R. Task-aware representation of sentences for generic text classification. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 3202–3213.
10. Harrando, I.; Troncy, R. Explainable Zero-Shot Topic Extraction Using a Common-Sense Knowledge Graph. In *Open Access Series in Informatics (OASIcs), Proceedings of the 3rd Conference on Language, Data and Knowledge (LDK 2021)*; Gromann, D., Sérasset, G., Declerck, T., McCrae, J.P., Gracia, J., Bosque-Gil, J., Bobillo, F., Heinisch, B., Eds.; Schloss Dagstuhl—Leibniz-Zentrum für Informatik: Dagstuhl, Germany, 2021; Volume 93, pp. 17:1–17:15. [CrossRef]
11. Rios, A.; Kavuluru, R. Few-shot and zero-shot multi-label learning for structured label spaces. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; Volume 2018, p. 3132.
12. Dauphin, Y.N.; Tur, G.; Hakkani-Tur, D.; Heck, L. Zero-shot learning for semantic utterance classification. *arXiv* **2013**, arXiv:1401.0509.
13. Ko, Y.; Seo, J. Automatic text categorization by unsupervised learning. In *Proceedings of the COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2000.
14. Haj-Yahia, Z.; Sieg, A.; Deleris, L.A. Towards unsupervised text classification leveraging experts and word embeddings. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July– 2 August 2019; pp. 371–379.
15. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *arXiv* **2016**, arXiv:1607.04606.
16. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
17. Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P.N.; Hellmann, S.; Morsey, M.; Van Kleef, P.; Auer, S.; et al. Dbpedia—A large-scale, multilingual knowledge base extracted from wikipedia. *Semant. Web* **2015**, *6*, 167–195. [CrossRef]
18. Tuggener, D.; von Däniken, P.; Peetz, T.; Cieliebak, M. LEDGAR: A Large-Scale Multi-label Corpus for Text Classification of Legal Provisions in Contracts. In Proceedings of the 12th Language Resources and Evaluation Conference, Online, 13–15 May 2020; European Language Resources Association: Marseille, France, 2020; pp. 1235–1241.