NLP and Insurance – Workshop Results at SwissText 2022

Claudio Giorgio Giancaterino[1]

[1]Intesa SanPaolo Vita, Milano, Italy

Abstract

Natural Language Processing, briefly NLP, will lead in the next years the revolution of the Artificial Intelligence for the Insurance industry. There are several opportunities to employ NLP in the insurance activities, from claims processing to fraud detection and chatbots. In the marketing field, NLP can be used to monitor the sentiment analysis of feedback that people publish on different social networks to better consider insured needs or extract insight risks. Textual analysis of claims and classification can simplify the claims processing to reduce time treatment, operational errors or provide help in fraud detection. Underwriting process can be improved by a better textual assessment. The workshop had the goal to show NLP techniques on fraud detection by the disaster Tweets data set from Kaggle classification competition.

1. Introduction

The workshop started with an introduction of the Natural Language Processing (NLP) explaining use cases in the Insurance world.

NLP can find a slot in broad Insurance fields, from Marketing to Underwriting, from Claims processing to Risk assessment, and also it can be applied in the traditional actuarial Reserving area.

The workshop was organized in the manner to show an application of NLP techniques in the Insurance Fraud Detection by the disaster Tweets data set retrieved from Kaggle classification competition. [1]

The first approach was to apply an Exploratory Data Analysis by the use of the word cloud, statistics of tweets and the language used in the data set.

After the pre-processing activity the work went ahead deeply into the text discovering Named Entity Recognition, Part of Speech Tagging, N-grams analysis, Topic Modelling and Word Embedding.

The workshop ended with the classification task predicting which tweets are real disasters and which one are not by the use of Transfer Learning applied in easy way with the help of "ktrain" library, and exploring the model inference on the test set.

The job was focused on the use of BERT both in the supervised and unsupervised learning tasks. [2]

2. Natural Language Processing in the Insurance world

Natural Language Processing is a branch of Artificial Intelligence with the aim to design some models allowing computers to understand natural language in order to perform some tasks. Some NLP applications are machine translation, question answering, and text summarization.

In the following, some NLP opportunities in the Insurance industry. [1]

-Marketing: NLP can be used to monitor the sentiment analysis of feedback to better consider insured needs, to monitor risks insight on what people are thinking about a particular product, to extract information about expected trends to improve marketing strategy.

-Underwriting: using Optical Character Recognition (OCR) and NLP is possible to extract information from medical reports and help underwriters in a better quote of the insurance coverage. NLP can categorize patients' diseases and retrieves correlation between some symptoms and the likely cost of treatment for the Insurance Company.

-Reserving: the analysis of claim reports during the first notification of loss can improve the reserving process for severe claims.

-Claims processing: textual analysis can simplify the trial reducing the time treatment of the process and reducing operational mistakes.

-Risk management: text classification is useful in the risk assessment giving help in fraud detection.

3. Exploratory Data Analysis

3.1. Activity

Before to start with the application of any Machine Learning model, is better to understand data involved in the project, and Exploratory Data Analysis is a block between data cleaning and data modeling with the goal to understand patterns, detect mistakes, check assumptions and check relationships between variables of a data set with the help of graphical charts and summary statistics.

SwissText 2022: Swiss Text Analytics Conference, June 08–10, 2022, Lugano, Switzerland

claudio.giancaterino@intesasanpaolovita.it (C. G. Giancaterino)

CEUR Workshop Proceedings (CEUR-WS.org)

[1]https://www.kaggle.com/competitions/nlp-getting-started

[2]https://github.com/claudio1975/SWISSTEXT_2022

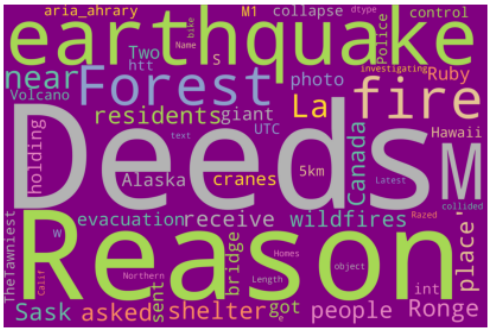

Figure 1: wordcloud tweets.

3.2. Results

The data set has 7.613 rows and 5 columns with ”id”, ”text”, ”target” and other 2 columns: ”keyword” and ”location”.

Text of the tweets are written in English for the almost whole data frame.

The data set has a length of 7.613 tweets with an average of 15 words per tweet, an average of 6 characters word length, and roughly 5 stop words per tweet.

Tweets are largely coming from the USA followed by UK and Canada. From the word cloud tool appears words “earthquake”, “deeds”, “reason”, “forest” and “fire” as the most common terms.

4. Text Analysis

4.1. Activity

4.1.1. Named Entity Recognition and Part of Speech Tagging

Text Analysis tools are used by Companies to extract valuable insights from text data.

The first approach followed was to explore Named Entity Recognition (NER) and Part of Speech (POS) Tagging by the use of “Spacy” library.

A named entity is a proper noun that refer to a specific entity like location, person, date, time, organization, etc. Named Entity Recognition systems are used to identify and segmentize named entities in text, and you can find them in application such as question answering, information retrieval, and machine translation. In Insurance, NER applications can be found in customer care to classify customer complaints.

A POS Tagging is the process on marking tokens in a part of speech like nouns, adverbs, verbs, punctuation, etc. The purpose of a POS tagger is to assign grammatical information to each token and in this way is possible to generate tree representation of the sentence. POS tagging works as a prerequisite for further NLP analysis such as chunking, syntax parsing, information extraction, machine translation, sentiment analysis, grammar analysis and word-sense disambiguation.

4.1.2. Text cleaning and N-grams

At this point is required a pre-processing step, known as text cleaning, to generate features, extracts patterns in text and further analysis. The aim of this process is preparing raw text for Natural Language Processing. Text cleaning is based on several steps starting with the text normalization, proceeding with removing Unicode characters and stop words, and ending with performing stemming/lemmatization.

The analysis went ahead with N-grams, a contiguous sentence of n items generated from a given sample of text where the items can be words. N-grams can be used as features from text corpus for Machine Learning models, moreover they are useful in autocompletion of sentences, or speech recognition helping to predict the next word that should be occurring in a sequence.

In the workshop N-grams were performed by Bag-of-Words (BoW) approach retrieving top occurrence of unigram, bigrams and trigrams in the data set.

Bag-of-Words [2] belong to a first level of techniques used to transform raw text into numerical features by vectorization. BoW describes the occurrence of words taken from a vocabulary obtained from a corpus text by labelling it in a binary vector.

Each word is represented by a one-hot vector, a sparse vector in the size of the vocabulary. The Bag-of-Words feature vector is the sum of all one-hot vectors of the words.

4.1.3. Word Embedding

The second level to transform raw text into numerical features is given by Word Embedding with the goal to generate vectors encoding semantic meanings: individual words are represented by vectors in a predefined vector space [3]. It allows words with similar meaning to have a similar representation. In the workshop Word Embedding was performed by Word2vec using “gensim” library. Word2vec is a neural network that try to maximize the probability to see a word in a context window, the cosine similarity between two vectors. This task was explored by two architectures: continuous bag-of-words (CBOW) model that try to predict a word from its context and by continuous skip-gram model that try to predict the context from a word.

```
similar(model_cbow, 'earthquake')
```

```
[('house', 0.3591579794883728),
 ('panic', 0.2768034338951111),
 ('death', 0.2711658477783203),
 ('nuclear', 0.2659183442592621),
 ('think', 0.25583887100219727),
 ('damage', 0.23612067103385925),
 ('electrocute', 0.21050655841827393),
 ('japan', 0.1991567760705948),
 ('california', 0.199130117893219),
 ('watch', 0.1940617561340332)]
```

Figure 2: ”earthquake” similar words from CBOW model.

```
similar(model_sg, 'earthquake')
```

```
[('house', 0.40713250637054443),
 ('panic', 0.3077165484428406),
 ('nuclear', 0.29869383573532104),
 ('death', 0.2960660457611084),
 ('think', 0.2847999334335327),
 ('california', 0.28187745809555054),
 ('japan', 0.26151421666145325),
 ('damage', 0.25667309761047363),
 ('watch', 0.2399403303861618),
 ('wreck', 0.23055985569953918)]
```

Figure 3: ”earthquake” similar words from Skip-Gram model.

4.1.4. Topic Modelling

The activity of this chapter ended with Topic Modelling, a form of unsupervised learning that works discovering hidden relationships in the text, more precisely the purpose is to identify topics in a document.

The purpose of the workshop was to discover new available and best performing tools for NLP, and for this activity was explored the use of BERTopic developed by Maarten Grootendorst. [3]

There are four key components in BERTopic [4], and can be considered as an ensemble of models.

It starts generating several document embeddings to represent the meaning of the sentences using a pre-trained language model: BERT (Bidirectional Encoder Representations from Transformers).

Given the huge dimension of vectors generated, is applied a general non-linear dimensionality reduction technique: UMAP (Uniform Manifold Approximation and Projection).

At this point the reduced embeddings are clustered with HDBSCAN (Hierarchical Density-Based Spatial Clustering). It finds clusters of variable densities converting DBSCAN into hierarchical clustering.

To retrieve topics from clustered document is applied a modified version of TF-IDF (Term Frequency-Inverse Document Frequency): the class-based TF-IDF procedure, where the class represents the collection of documents merged into a single document per each cluster.

Input features for BERTopic has been generated by TF-IDF that is an extension of Bag of Words, where terms are weighted and in this way are highlighted words with useful information [2].

```
model.get_topic(9)
```

```
[('wildfire', 0.284084992758143),
 ('california', 0.27536872346507246),
 ('northern', 0.26136464655634073),
 ('ablaze', 0.13055265826022472),
 ('arson', 0.12168477785910824),
 ('arsonist', 0.11723194770474094),
 ('rocky', 0.09048708492821848),
 ('catch', 0.06390157645701867),
 ('suspect', 0.06187450923089676),
 ('advance', 0.049968548519998025)]
```

Figure 4: Top words from a topic by BERTopic model.

4.2. Results

From the N-grams analysis the top occurrence words are: “people”, ”video”, ”crash”, ”emergency”, and ”disaster”. Looking at the bigrams they are: “suicide bomber”, “youtube video”, “northern california”, “california wildfire”, “bombe detonate”, and “natural disaster”.

Interesting results are coming from Word Embedding, where the vocabulary is similar between the two architectures models with these relevant words: “earthquake”, “forest”, “evacuation”, “people”, ’wildfire’, “california”, “flood”, “disaster”, “emergency”, “damage”.

What is changing is the similarity between words, the CBOW model usually provides a similarity between words lower than the one provided by the Skip-Gram model.

The Word Embedding exploration ended reducing the vectors dimension with the application of the Principal Component Analysis giving the opportunity for words visualization into two dimensions.

After that, was the turn of the Topic Modelling with BERTopic. The tool was used in easy way, without fine tuning parameters, using TF-IDF features as input and with the arbitrary choice of ten topics.

Participants asked how to trust in results, so the job has been completed with the coherence score evaluation of BERTopic and the comparison with the Latent Dirichlet Allocation (LDA) model [5], the common model

[3] https://github.com/MaartenGr/BERTopic

employed in the Topic Modelling.

The same relevant words from BERTopic appears also in LDA, but with different probabilities. The evaluation has been done with the UMass coherence score that calculates how often two words appear together in the corpus. The perfect coherence is in 0, and it usually decrease with the rising number of topics. The issue with the LDA model was that the number of trained documents in each chunk has been reduced to make it converge. From results BERTopic shows a number closer to 0 (roughly -14) than LDA model (roughly -18), though the result can be improved tuning the model.

5. Text Classification

5.1. Activity

The approach followed in this chapter is the classification task: given a target variable the aim is to predict if a tweet can be considered a "disaster" or otherwise "not disaster". Twitter has become an important emergency communication channel, because people by smartphones are able to announce an emergency they're observing in real-time. For this reason, more agencies and Insurance Companies are interested in monitoring Twitter. Moreover, it's not always clear whether a person's words are actually announcing a disaster, so this task can be linked to a Fraud Detection task.

The approach followed was to use Transfer Learning [6] that is a Machine Learning method where a model developed for a task is reused as the starting point for a model on a second task. In the traditional Supervised Learning approach, Machine Learning models are trained on labelled data sets expecting to perform well on unseen data of the same task and domain. The traditional approach falling down when there is not enough labelled data to perform training for the task or domain of interest.

The idea behind Transfer Learning is to try to store the knowledge gained in solving the source task in the source domain and apply it to another similar problem of interest, it is the same concept of learning process by experience, so the aim is to exploit pre-trained models that can be fine-tuned on smaller task or specific data sets.

Bidirectional Encoder Representation from Transformers (BERT) is one of the most popular state-of-art NLP approaches for Transfer Learning, published by Google in 2018 [7].

BERT is a bidirectional multi-layer Transformer model that exploits the attention mechanism [8]. A basic Transformer uses an encoder-decoder architecture. The encoder learns the representation from the input sentence and the decoder receives the representation and produces a prediction for the task.

The attention mechanism was introduced to improve the performance of the encoder-decoder model for machine translation. The idea behind the attention mechanism was to permit the decoder to use the most relevant parts of the input sequence in a flexible manner, by a weighted combination of all of the encoded input vectors.

With Transformers we had reached the third level of vectorization technique in NLP, the Contextual Embedding [9]. Both traditional Word Embedding (word2vec, Glove) and Contextual Embedding (ELMo, BERT), aim to learn a continuous (vector) representation for each word in the documents.

The Word Embedding method builds a global vocabulary using unique words in the documents by ignoring the meaning of words in different context. Hence, given a word, its embedding is always the same in whichever sentence it occurs, and for this reason, the pre-trained Word Embeddings are static.

Contextual Embedding methods are used to learn sequence-level semantics by considering the sequence of all words in the documents. The embeddings are obtained from a model by passing the entire sentence to the pre-trained model. The embeddings generated for each word depends on the other words in a given sentence. The Transformer based models work on attention mechanism, and attention is a way to look at the relation between a word with its neighbours, and for this reason, pre-trained Contextual Embeddings are dynamic.

BERT's goal is to generate a language representation model and uses a rich input embedding representation, derived from a sequence of tokens, which is converted into vectors and then three embedding layers are combined to obtain a fixed-length vector processed in the Neural Network. BERT is pre-trained using two Unsupervised Learning tasks: Masked LM (MLM) and Next Sentence Prediction (NSP).

The usual workflow of BERT consists of two stages: pre-training and fine-tuning.

The attention mechanism in the Transformer allows BERT to model many downstream tasks, such as sentiment analysis, question answering, paraphrase detection and more. In this workshop has been used the "ktrain" [10] library, a low-code library developed by Arun S. Maiya [4] that provides a lightweight wrapper for "Keras", making it easier to build, train, and deploy Deep Learning models.

5.2. Results

Thanks to the "ktrain" low coding library was easy the implementation of BERT, just to split the data set into a

[4] https://github.com/amaiya/ktrain

Metrics on disaster classification

	model	metrics	train	test
0	BERT_by_ktrain	Accuracy	0.853530	0.821405
1	BERT_by_ktrain	F1_score	0.853102	0.820450

Figure 5: BERT model results.

Metrics on disaster classification

	model	metrics	train	test
0	LR_by_ktrain	Accuracy	0.628900	0.586999
1	LR_by_ktrain	F1_score	0.629975	0.586999

Figure 6: Logistic Regression model results.

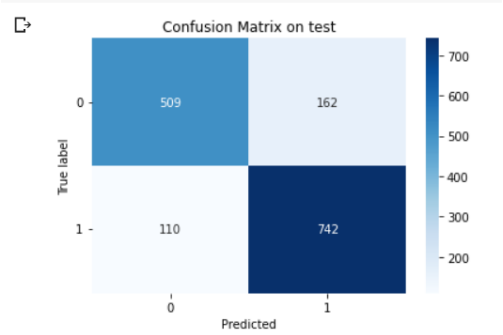

Figure 7: Confusion matrix on test set with BERT.

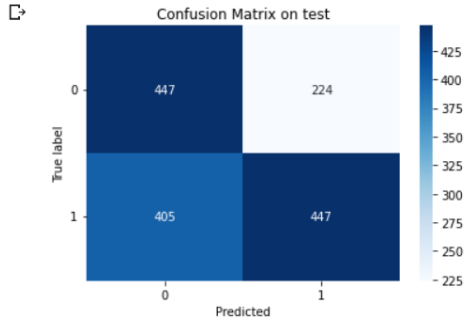

Figure 8: Confusion matrix on test set with Logistic Regression.

train and test set, then sent the train set as input into the "ktrain" pre-processing and Deep Learning model.

The Transformer based model shows, as expected, a good performance both on the train set and test set with an accuracy greater than 82%. Given the imbalanced data set the performance was evaluated also by F1 score with quite the same results.

Participants asked an evaluation with other models, and the job has been completed with the common classification Machine Learning model: the Logistic Regression, always with "ktrain" library.

Implementation of this model is easy, but there is a poorly performance: roughly 63% on train set and roughly 58% on validation set.

Looking at the confusion matrix, BERT model shows a large number of elements across the diagonal and small number of elements off the diagonal, so a better matrix than the Logistic Regression.

Last step was the inference of the model, testing the right prediction of test tweets, and also in this situation BERT outperformed Logistic Regression model with all right predictions.

6. Conclusions

With this workshop there has been the opportunity to have an overview of Natural Language Processing applications in the Insurance world, and with the disaster Tweets data set there has been the opportunity to discover NLP applications in Fraud Detection.

In this work the development of Natural Language Processing has been retracted, from Tokenization to Bag of Words, from Word Embedding to Contextual Embedding.

Transfer Learning has been explored, looking on its potential that outperforms benchmark models both on Topic Modelling and Classification prediction.

References

[1] A. Ly, B. Uthayasooriyar, T. Wang, A survey on natural language processing (nlp) and applications in insurance, arXiv preprint arXiv:2010.00462 (2020).

[2] A. Ferrario, M. Nägelin, The art of natural language processing: classical, modern and contemporary approaches to text document classification, Modern and Contemporary Approaches to Text Document Classification (March 1, 2020) (2020).

[3] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013).

[4] M. Grootendorst, Bertopic: Neural topic modeling with a class-based tf-idf procedure, arXiv preprint arXiv:2203.05794 (2022).

[5] H. Jelodar, Y. Wang, C. Yuan, X. Feng, X. Jiang, Y. Li, L. Zhao, Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey, Multimedia Tools and Applications 78 (2019) 15169–15211.

[6] A. Malte, P. Ratadiya, Evolution of transfer learning in natural language processing, arXiv preprint arXiv:1910.07370 (2019).

[7] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[9] Q. Liu, M. J. Kusner, P. Blunsom, A survey on contextual embeddings, arXiv preprint arXiv:2003.07278 (2020).

[10] A. S. Maiya, ktrain: A low-code library for augmented machine learning, J. Mach. Learn. Res 23 (2020) 1–6.