Natural Language Processing for Actuarial Applications

Dr. Andreas Troxler

Actuarial Data Science Après-midi

15 Sep 22

AT Analytics

Natural Language Processing for Actuarial Applications

- Introduction and background
- Theory (just a bit)
- Supervised applications
- Unsupervised applications
- Conclusions

Introduction and background

Speaker – purpose – further information

Dr. Andreas Troxler – AT Analytics

- PhD in computational science
- Actuary (Swiss Association of Actuaries), FCAS, CERA, PRM
- Udacity – Machine Learning Engineer

- Pricing, reserving, solvency and capital modeling
- Non-life and health insurance, reinsurance, consultancy

- Founder of **AT Analytics**
- **Actuarial and data analytics consulting**
- https://atanalytics.ch/en/

- By the way, keen to learn about InsurTech and FinTech opportunities

Why?

An abundant amount of information is available in the form of text!

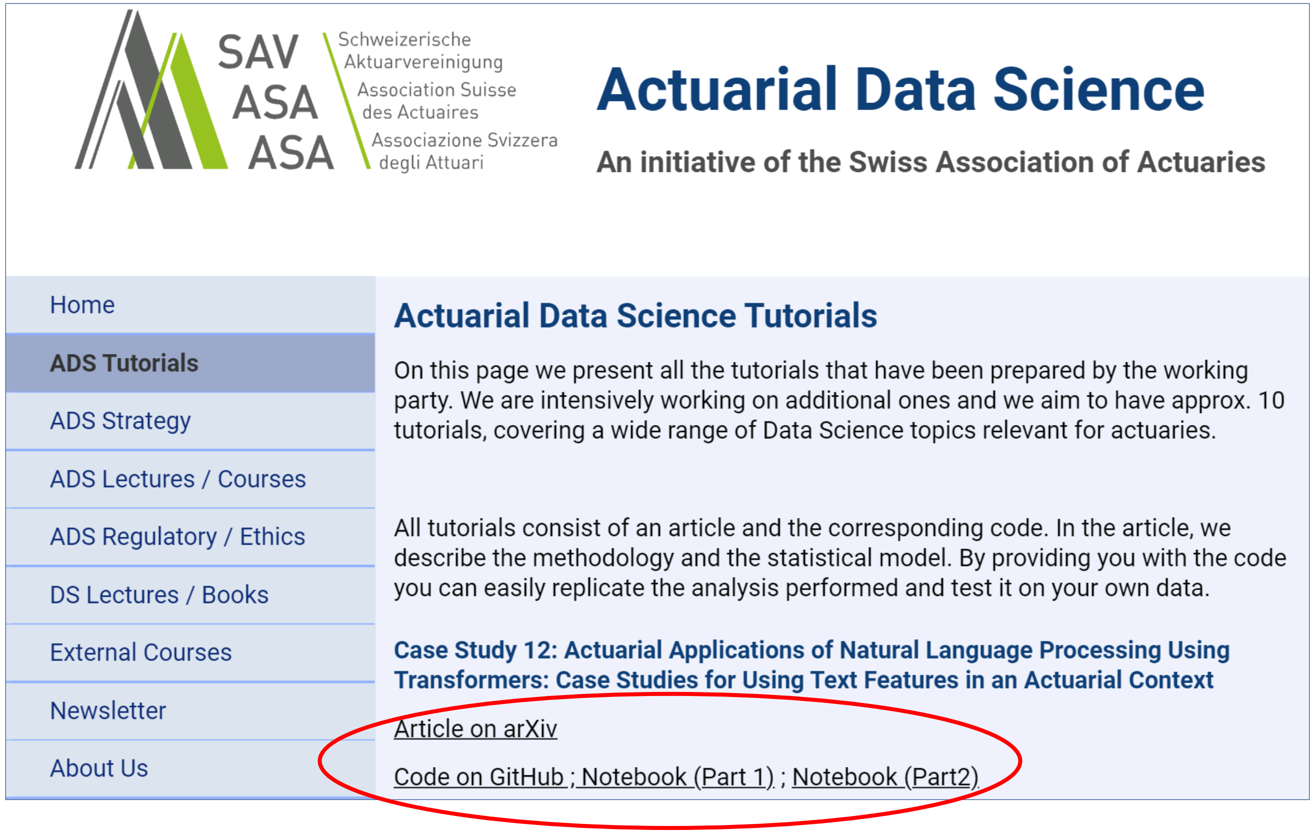
We want to use this text data as input features for predictive models.

Challenges:

- text data is unstructured …
- multiple languages
- specialized terminology
- very short or rather long text samples
- small amount of (labeled) training data
- need to understand why the model arrives at a particular prediction

For more detail, data, and code, please refer to…

www.actuarialdatascience.org



Actuarial Data Science

An initiative of the Swiss Association of Actuaries

Actuarial Data Science Tutorials

On this page we present all the tutorials that have been prepared by the working party. We are intensively working on additional ones and we aim to have approx. 10 tutorials, covering a wide range of Data Science topics relevant for actuaries.

All tutorials consist of an article and the corresponding code. In the article, we describe the methodology and the statistical model. By providing you with the code you can easily replicate the analysis performed and test it on your own data.

Case Study 12: Actuarial Applications of Natural Language Processing Using Transformers: Case Studies for Using Text Features in an Actuarial Context

Article on arXiv

Code on GitHub ; Notebook (Part 1) ; Notebook (Part2)

https://arxiv.org/abs/2206.02014

Actuarial Applications of Natural Language Processing Using Transformers

Case Studies for Using Text Features in an Actuarial Context

Andreas Troxler * Jürg Schelldorfer **

v1, 3 June 2022

Abstract

This tutorial demonstrates workflows to incorporate text data into actuarial classification and regression tasks. The main focus is on methods employing transformer-based models. A dataset of car accident descriptions with an average length of 400 words, available in English and German, and a dataset with short property insurance claims descriptions are used to demonstrate these techniques. The case studies tackle challenges related to a multi-lingual setting and long input sequences. They also show ways to interpret model output, to assess and improve model performance, by fine-tuning the models to the domain of application or to a specific prediction task. Finally, the tutorial provides practical approaches to handle classification tasks in situations with no or only few labeled data. The results achieved by using the language-understanding skills of off-the-shelf natural language processing (NLP) models with only minimal pre-processing and fine-tuning clearly demonstrate the power of transfer learning for practical applications.

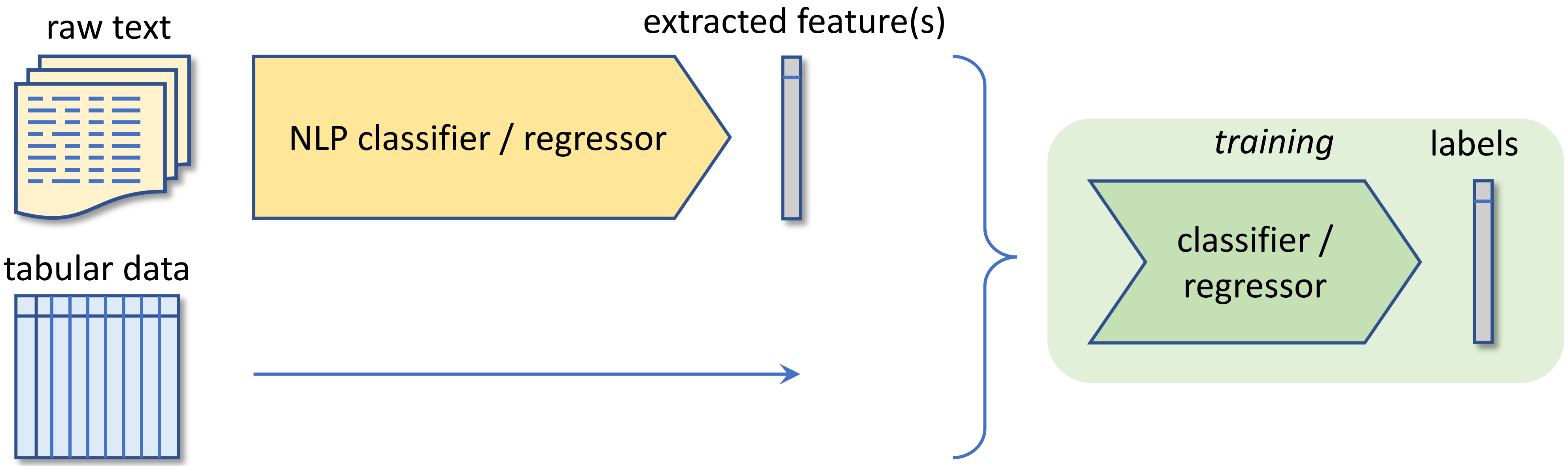
Keywords. Natural language processing, NLP, transformer, multi-lingual models, domain-specific fine-tuning, integrated gradients, extractive question answering, zero-shot classification, topic modeling.

A dash of theory

Key concepts: tokenization – word embedding – self-attention

How?

- Text features can be used to augment existing tabular data by additional columns



- Here, we look at possible ways to implement the “NLP classifier / regressor”

Transformers

- Neural network architecture developed by Google researchers in 2017.
- Uses word embeddings and self-attention layers to understand words in their context.
- Quickly became dominant for achieving state-of-the art results on many NLP tasks.

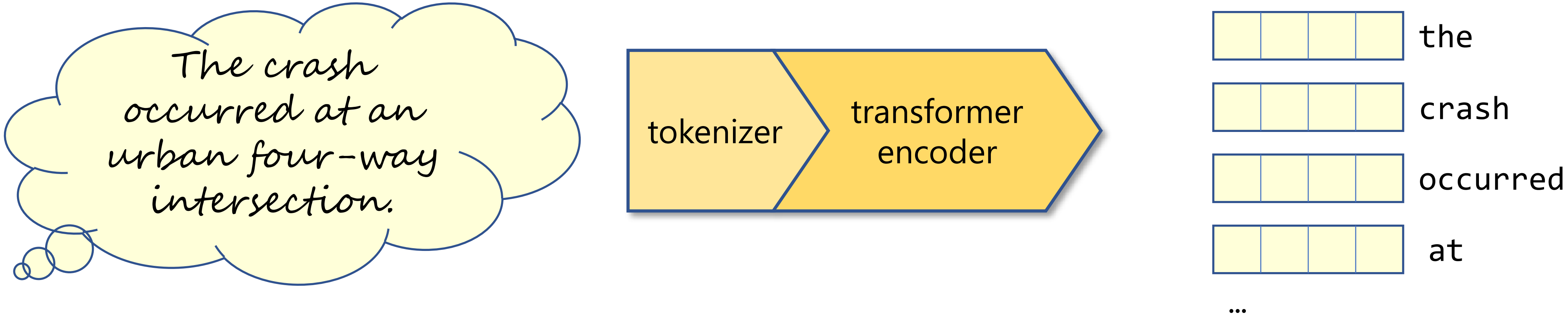
- BERT (Bidirectional Encoder Representations from Transformers) is a Transformer encoder architecture, introduced in 2019

- Multilingual DistilBERT, derived from BERT: 134 million parameters, pre-trained on Wikipedia in 104 different languages
- Multilingual alternatives: XLM, XLM-RoBERTa, …

- Easy-to use Python library and model hub provided by 🤗 Huggingface (https://huggingface.co/)

How does it work? Key concepts

- Encode unstructured raw text into sequence of embeddings (vectors in $\mathbb{R}^E$)



- The embeddings serve as input features to supervised or unsupervised machine learning tasks

Tokenization

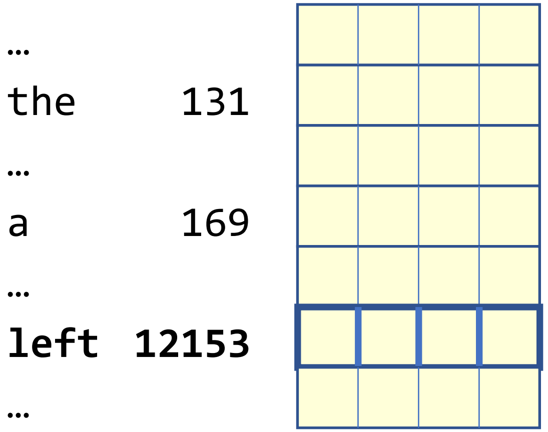
- Split the raw text into tokens (items of a pre-defined vocabulary of size V)
- Example (using the distilbert-base-multilingual-cased model):

raw text `V1, a 2000 Pontiac minivan, made a left turn from a private driveway`

preprocessed text	`[CLS]`	`V`	`##1`	`,`	`a`	`2000`	`Pont`	`##iac`
Token IDs	101	159	10759	117	169	10180	23968	46917

Reducing dimensionality: Word embedding

- Map token IDs into dense vector space $\mathbb{R}^E$: $\{0,1\}^V \to \mathbb{R}^E$
- Embedding dimension $E \ll$ vocabulary size V
- distilbert-base-multilingual-cased: $V = 120k$, $E = 768$

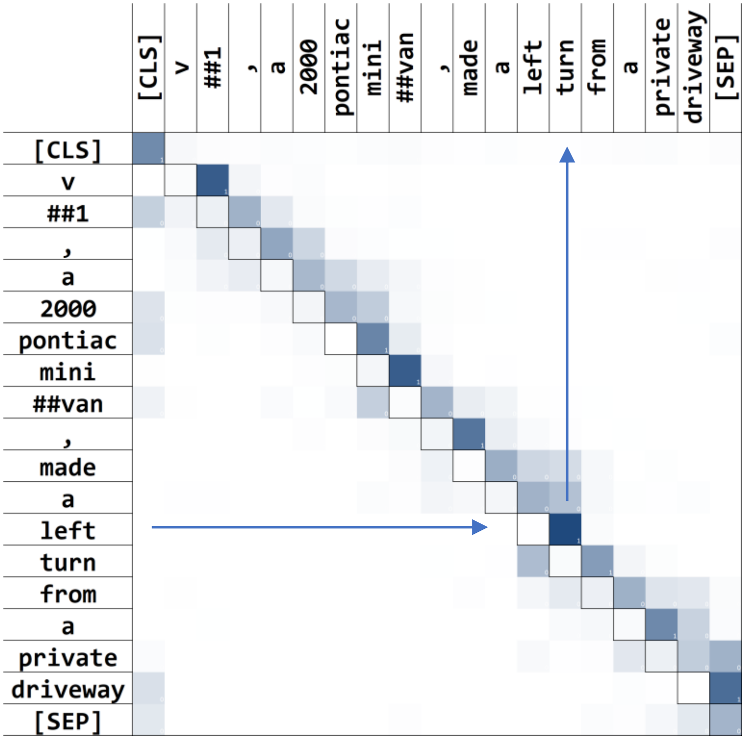


…	
the	131
…	
a	169
…	
left	**12153**
…	

- Ambiguous words: “the vehicle turned **left**” vs “the vehicle **left** the highway”

Understanding each word in its context: self-attention

- Self-attention mechanism
- For each token, calculate the weighted average of the embeddings of all words of the sequence
- This helps understanding the word in its context, e.g. “left” vs “left”
- In the example, “left” strongly attends to “turn” ➔ “left turn”

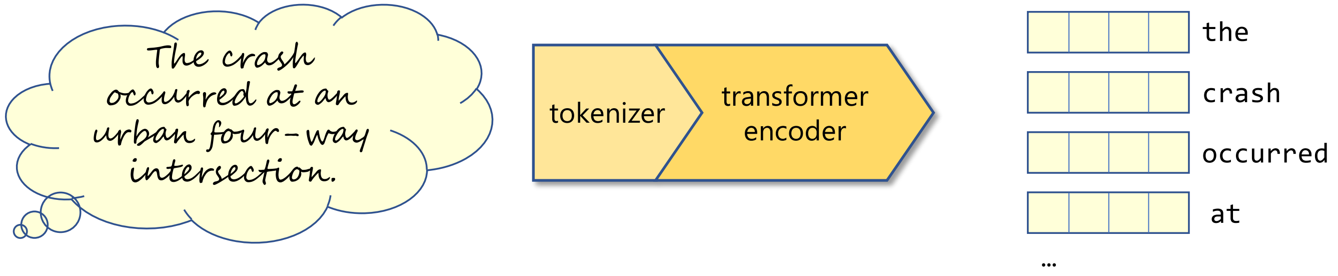


Theory – Summary

- Multi-lingual transformer models, **pre-trained** on large corpora of data

- Raw texts are first **tokenized**

- Then **encoded** it into a sequence real-valued vectors, using word embedding and several self-attention-layers to understand word in contexts



- This is **structured** data!

Supervised applications

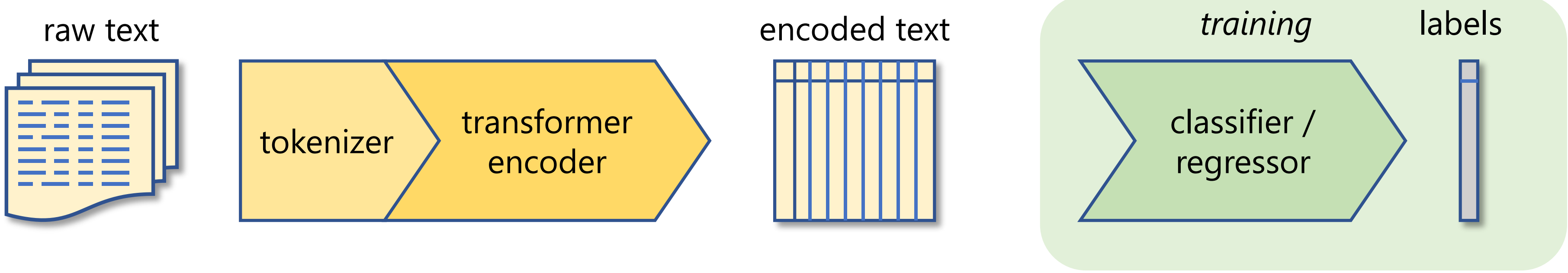
Case studies on a multilingual road accident reports (English, German)

Case study: Road accident reports

- Ca. 7'000 reports describing the accident situation, road and weather conditions, vehicles, drivers, …
- 80% English (source: https://www.nhtsa.gov/), 20% German (DeepL)
- Average length: about 400 words
- Tasks: predict number of vehicles involved (1, 2, 3+) and presence of bodily injury (0, 1)

> V1, a 2000 Pontiac Montana minivan, made a left turn from a private driveway onto a northbound 5-lane two-way, dry asphalt roadway on a downhill grade. The posted speed limit on this roadway was 80 kmph (50 MPH). V1 entered the roadway by crossing over the two southbound lanes and then entering the third northbound lane, which was a left turn-only lane at a 4-way intersection. The driver of V1 intended to travel straight through the intersection, and so he began to change lanes to the right. He did not see V2, a 1994 Pontiac Grand Am, that was traveling in the second northbound lane. The northbound roadway had curved to the right prior to the private driveway that V1 had exited. As V1 began to change lanes to the right, the front of V1 contacted the left rear of V2 before coming to final rest on the roadway.
>
> The driver of V1 was a 60-year old male who reported that he had been traveling between 2-17 kmph (1-10 mph) prior to the crash. He had no health-related problems, and had taken no medication prior to the crash. He was rested and traveling back home. He was wearing his prescribed lenses that corrected a myopic (nearsighted) condition. He did not sustain any injuries from the crash and refused treatment. The Critical Precrash Event for the driver of V1 was when he began to travel over the lane line on the right side of the travel lane. The Critical Reason for the Critical Precrash Event was inadequate surveillance (failed to look, looked but did not see). Associated factors coded to the driver of V1 include an illegal use of a left turn lane (cited by police) and an unfamiliarity with the roadway. As per the driver of V1, this was the first time he had driven on this roadway.
>
> […]

Approach 1: Apply any classifier on the encoded texts



(1) Train-test split: 80% of the records are used as training set and the remainder as test set.

(2) Tokenize, using the standard tokenizer of the model **distilbert-base-multilingual-cased**.

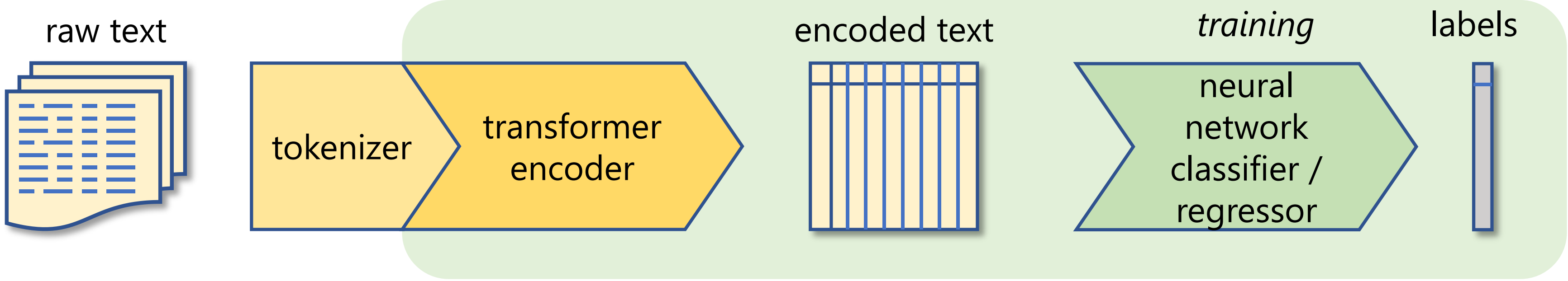
(3) Apply the model to obtain the outputs of the final layer – the encoded sequence (dimension 768).

(4) Condense the encoded sequence into a single vector in $\mathbb{R}^{768}$ by mean pooling (averaging over the sequence)

(5) Use this vector as input feature for a classifier model, and train the classifier to predict the label. Here: multinomial logistic regression classifier with L2-regularization (GLM), from scikit-learn.

Step (5) is the only task-specific step. The NLP model is not tuned to the task at hand!

Approach 2: Task-specific fine-tuning



(1) Train-test split: 80% of the records are used as training set and the remainder as test set.

(2) Tokenize, using the standard tokenizer of the model **distilbert-base-multilingual-cased**.

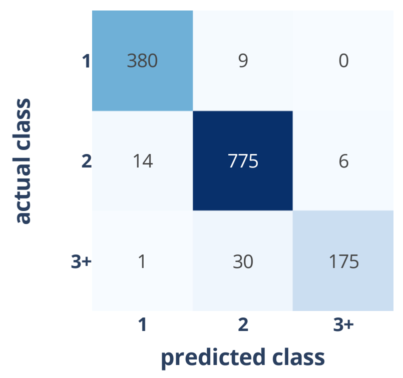
(3) Feed the encoder output into a neural network classifier and train both the transformer encoder and the classification head to the classification task.

- This approach is computationally intensive.
- Certain parameters of the transformer encoder could be held fixed.

Results – predicting the number of vehicles

				accuracy
	test data	**log loss**	**Brier loss**	**score**
Dummy classifier	no matter	0.961	0.574	57.2%
1. Logistic regression classifier	English	0.136	0.068	95.7%
	German	0.160	0.080	95.2%
2. Task-specific fine-tuning	English	0.028	0.009	99.6%
	German	0.072	0.030	98.3%

train EN/GE, test EN

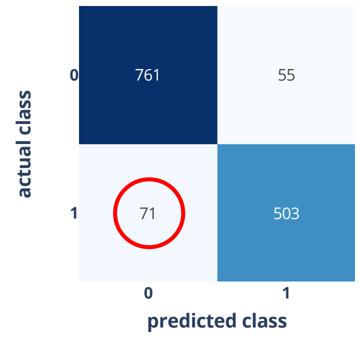


- Results evaluated on the test set, which has not been seen by the model during training.
- Logistic regression works well – the language model is used without any further training!
- Although German is underrepresented in the training data (20%), results for German test data are impressive.
- Very simple to implement – see tutorial.

Results – identifying bodily injury cases

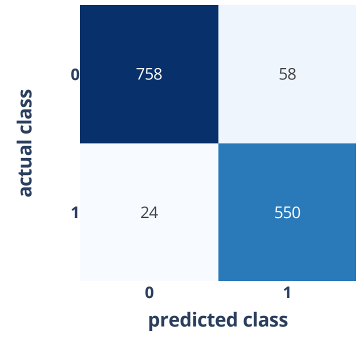
				accuracy	accuracy
	test data	log loss	Brier loss	score	score*
Dummy classifier	no matter	0.679	0.486	58.7%	
1. Logistic regression classifier	English	0.400	0.259	80.1%	
	German	0.517	0.345	74.5%	
2. Task-specific fine-tuning	English	0.244	0.139	90.9%	94.1%
	German	0.408	0.243	84.0%	89.4%

DistilBERT classifier - 2 epochs task-specific



actual class \ predicted class	0	1
0	761	55
1	71	503

DistilBERT classifier - split inputs

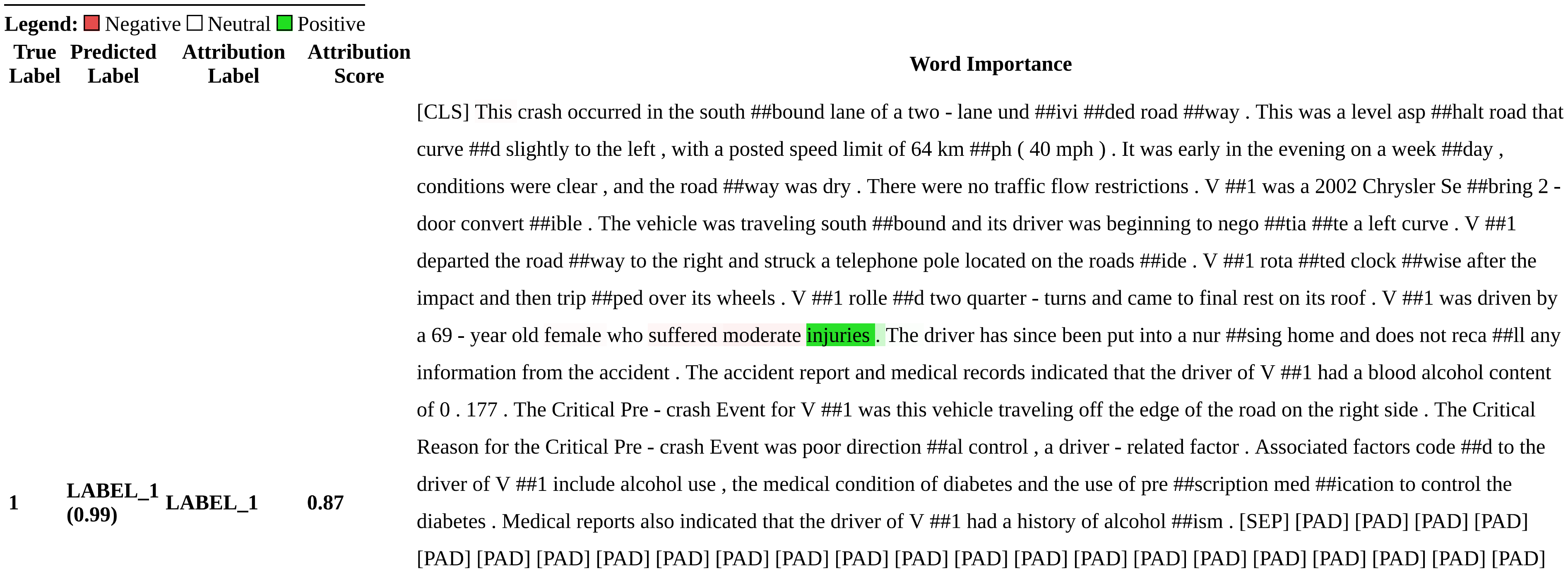


actual class \ predicted class	0	1
0	758	58
1	24	550

- This task is harder than predicting the number of vehicles, in particular for the German samples which are underrepresented in the training data (20%).
- Otherwise, similar conclusions as before.
- * Issue: Some accident reports are longer than the maximum sequence length of the model (512 tokens). This leads to false negatives.
  Possible solution: split the input sequences into slightly overlapping chunks.

Interpretability

- Which part of the text leads to a particular prediction? transformers-interpret!
- Example: Word importance attribution for identification of bodily injury cases



Legend: 🟥 Negative □ Neutral 🟩 Positive

True Label	**Predicted Label**	**Attribution Label**	**Attribution Score**	**Word Importance**
1	LABEL_1 (0.99)	LABEL_1	0.87	[CLS] This crash occurred in the south ##bound lane of a two - lane und ##ivi ##ded road ##way . This was a level asp ##halt road that curve ##d slightly to the left , with a posted speed limit of 64 km ##ph (40 mph) . It was early in the evening on a week ##day , conditions were clear , and the road ##way was dry . There were no traffic flow restrictions . V ##1 was a 2002 Chrysler Se ##bring 2 - door convert ##ible . The vehicle was traveling south ##bound and its driver was beginning to nego ##tia ##te a left curve . V ##1 departed the road ##way to the right and struck a telephone pole located on the roads ##ide . V ##1 rota ##ted clock ##wise after the impact and then trip ##ped over its wheels . V ##1 rolle ##d two quarter - turns and came to final rest on its roof . V ##1 was driven by a 69 - year old female who suffered moderate injuries . The driver has since been put into a nur ##sing home and does not reca ##ll any information from the accident . The accident report and medical records indicated that the driver of V ##1 had a blood alcohol content of 0 . 177 . The Critical Pre - crash Event for V ##1 was this vehicle traveling off the edge of the road on the right side . The Critical Reason for the Critical Pre - crash Event was poor direction ##al control , a driver - related factor . Associated factors code ##d to the driver of V ##1 include alcohol use , the medical condition of diabetes and the use of pre ##scription med ##ication to control the diabetes . Medical reports also indicated that the driver of V ##1 had a history of alcohol ##ism . [SEP] [PAD]

Supervised Learning – Summary

- Multi-lingual case study
- Handling of long input texts
- Explainability
- Transfer learning: Using pre-trained models ➔ little pre-processing or fine-tuning required
- Simple to implement

Unsupervised applications

No labels available? Welcome to the real world!

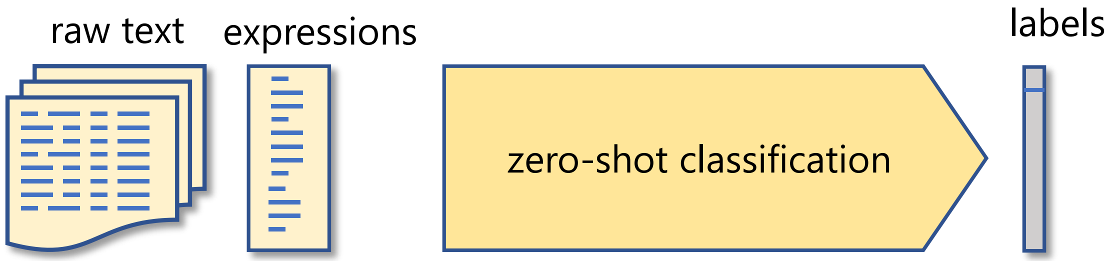
Case study: Property insurance claims descriptions

- Ca. 7’000 records
- Short English claim description (5 words on average)
- Hazard type with 9 different levels: Fire, Lightning, Hail, Wind, WaterW (weather related water claims), WaterNW (other water claims), Vehicle, Vandalism and Misc (any other)
- Task: Predict hazard type from claim description – **without using the labels!**

row	Description	Vandalism	Fire	Lightning	Wind	Hail	Vehicle	WaterNW	WaterW	Misc
1	lightning damage	0	0	1	0	0	0	0	0	0
2	lightning damage at Comm. Center	0	0	1	0	0	0	0	0	0
6	surveillance equipment stolen	1	0	0	0	0	0	0	0	0
7	wind blew stack off and damaged roof	0	0	0	1	0	0	0	0	0
9	forklift hit building damaging wall and door frame	0	0	0	0	0	1	0	0	0
11	water damage at courthouse	0	0	0	0	0	0	0	1	0
30	light pole damaged	0	0	0	0	0	1	0	0	0

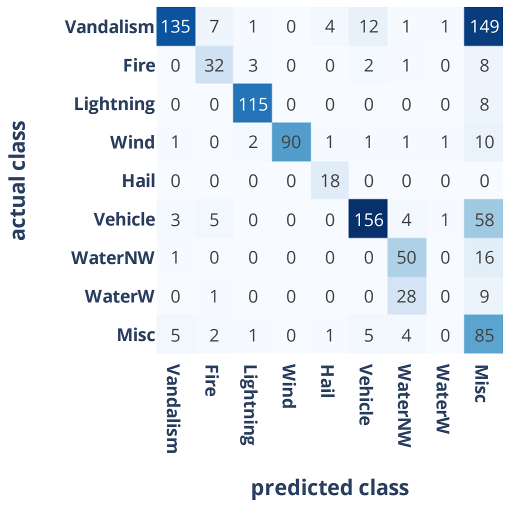
Approach 1: Zero-shot classification

- Zero-shot classification: Classification of text sequences in an unsupervised way (without having training data in advance and building a model).
- The model is presented with a text sequence and a list of expressions, and assigns a probability to each expression.
- This is very simple!



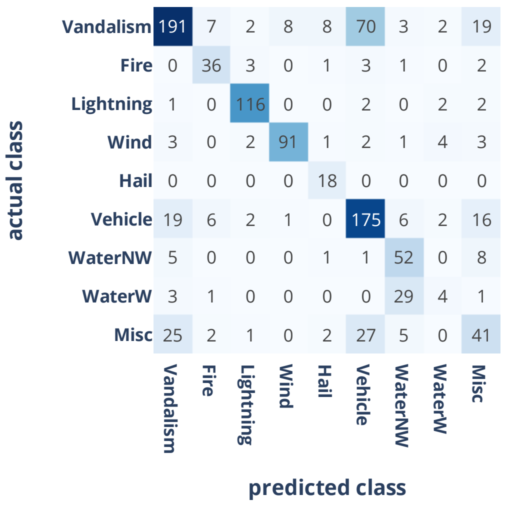
expression	hazard type	
"Vandalism"	0	Vandalism
"Theft"	0	Vandalism
"Fire"	1	Fire
"Lightning"	2	Lightning
"Wind"	3	Wind
"Hail"	4	Hail
"Vehicle"	5	Vehicle
"Water"	6	WaterNW
"Weather"	7	WaterW
"Misc"	8	Misc

Results – Zero-shot classification



Zero-shot-classification

actual class \ predicted class	Vandalism	Fire	Lightning	Wind	Hail	Vehicle	WaterNW	WaterW	Misc
Vandalism	135	7	1	0	4	12	1	1	149
Fire	0	32	3	0	0	2	1	0	8
Lightning	0	0	115	0	0	0	0	0	8
Wind	1	0	2	90	1	1	1	1	10
Hail	0	0	0	0	18	0	0	0	0
Vehicle	3	5	0	0	0	156	4	1	58
WaterNW	1	0	0	0	0	0	50	0	16
WaterW	0	1	0	0	0	0	28	0	9
Misc	5	2	1	0	1	5	4	0	85



Zero-shot classification, refined

actual class \ predicted class	Vandalism	Fire	Lightning	Wind	Hail	Vehicle	WaterNW	WaterW	Misc
Vandalism	191	7	2	8	8	70	3	2	19
Fire	0	36	3	0	1	3	1	0	2
Lightning	1	0	116	0	0	2	0	2	2
Wind	3	0	2	91	1	2	1	4	3
Hail	0	0	0	0	18	0	0	0	0
Vehicle	19	6	2	1	0	175	6	2	16
WaterNW	5	0	0	0	1	1	52	0	8
WaterW	3	1	0	0	0	0	29	4	1
Misc	25	2	1	0	2	27	5	0	41

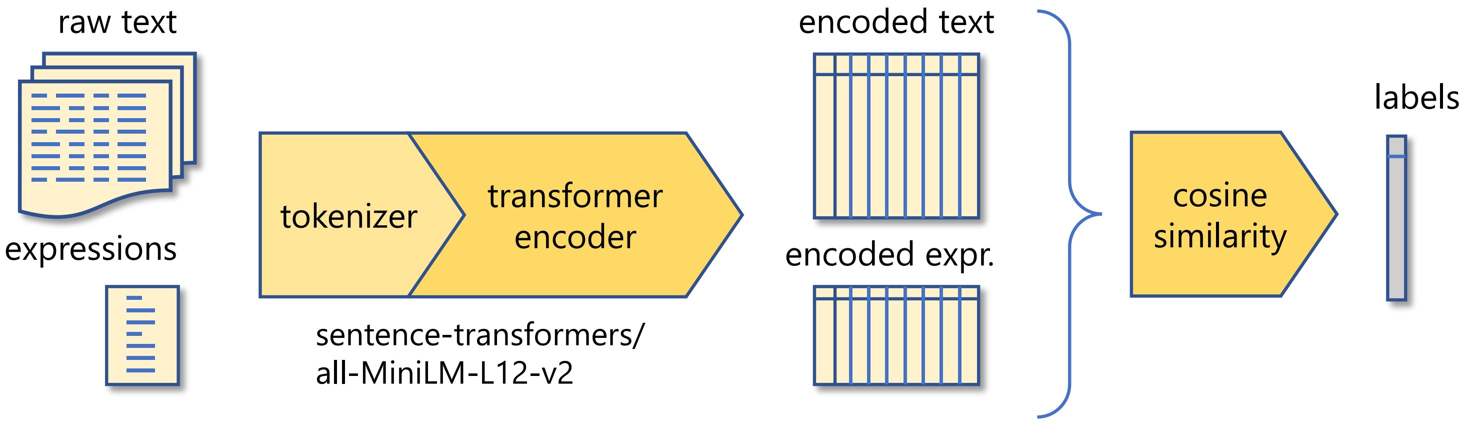
	log loss	Brier loss	accuracy score
Dummy classifier	1.977	0.835	29.8%
Logistic regression classifier (supervised)	0.554	0.233	84.7%
1. Zero-shot classification	1.043	0.463	65.5%
… refined: mapping "Misc"	n/a	n/a	69.7%

- Vehicle vs vandalism: “light pole damaged”
- Water: Weather vs non-weather unclear (e.g. “Water damage at courthouse”)
- Refinement: “Misc” produces many false positives. To mitigate, we select the second most likely hazard type if the probability margin for “Misc” is less than 50%.

Approach 2: Sentence similarity

(1) Encode each input sentence and each candidate expression into an embedding vector.

(2) Calculate pairwise cosine similarity scores. Select the expression with the highest score.

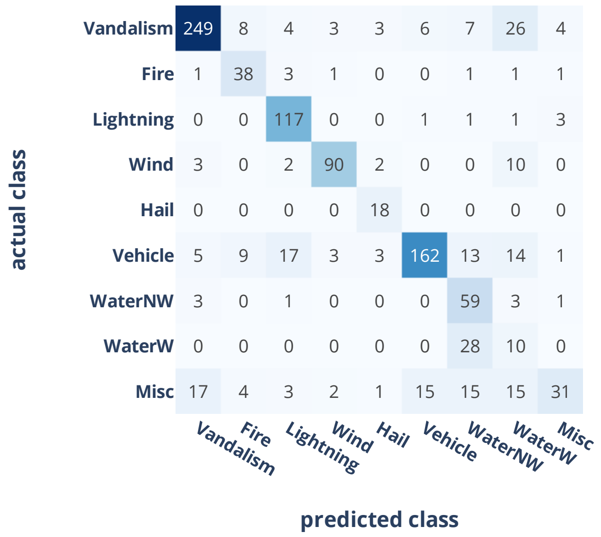


sentence-transformers/
all-MiniLM-L12-v2

expression	hazard type	
"Vandalism"	0	Vandalism
"Glass"	0	Vandalism
"Theft"	0	Vandalism
"Fire damage"	1	Fire
"Lightning damage"	2	Lightning
"Wind damage"	3	Wind
"Hail damage"	4	Hail
"Damage caused by a vehicle"	5	Vehicle
"Water damage"	6	WaterNW
"Weather damage"	7	WaterW
"Ice"	7	WaterW
"Electricity"	8	Misc
"Power surge"	8	Misc

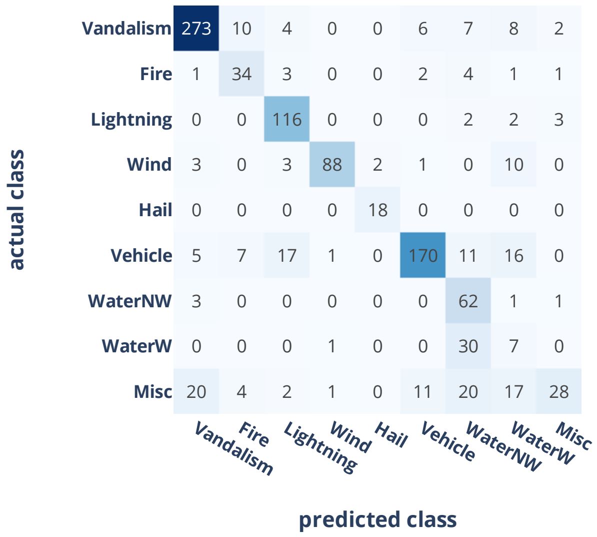
Results: Sentence similarity

Similarity



actual class \ predicted class	Vandalism	Fire	Lightning	Wind	Hail	Vehicle	WaterNW	WaterW	Misc
Vandalism	249	8	4	3	3	6	7	26	4
Fire	1	38	3	1	0	0	1	1	1
Lightning	0	0	117	0	0	1	1	1	3
Wind	3	0	2	90	2	0	0	10	0
Hail	0	0	0	0	18	0	0	0	0
Vehicle	5	9	17	3	3	162	13	14	1
WaterNW	3	0	1	0	0	0	59	3	1
WaterW	0	0	0	0	0	0	28	10	0
Misc	17	4	3	2	1	15	15	15	31

Simularity, refined



actual class \ predicted class	Vandalism	Fire	Lightning	Wind	Hail	Vehicle	WaterNW	WaterW	Misc
Vandalism	273	10	4	0	0	6	7	8	2
Fire	1	34	3	0	0	2	4	1	1
Lightning	0	0	116	0	0	0	2	2	3
Wind	3	0	3	88	2	1	0	10	0
Hail	0	0	0	0	18	0	0	0	0
Vehicle	5	7	17	1	0	170	11	16	0
WaterNW	3	0	0	0	0	0	62	1	1
WaterW	0	0	0	1	0	0	30	7	0
Misc	20	4	2	1	0	11	20	17	28

	log loss	Brier loss	accuracy score
Dummy classifier	1.977	0.835	29.8%
Logistic regression classifier (supervised)	0.554	0.233	84.7%
1. Zero-shot classification	1.043	0.463	65.5%
... refined: mapping "Misc"	n/a	n/a	69.7%
2. Sentence similarity	n/a	n/a	74.5%
... refined	1.172	0.403	76.6%

- Performance fine for most hazard types
- Exceptions: WaterW vs WaterNW, Misc
- Refinement: Train a supervised sequence classifier using the labels from the unsupervised approach.

Approach 3: Topic clustering

Idea: Identify clusters of “similar” texts and apply labels to the clusters

(1) Encode text sequence into a vector in $\mathbb{R}^E$, using a transformer model and mean pooling

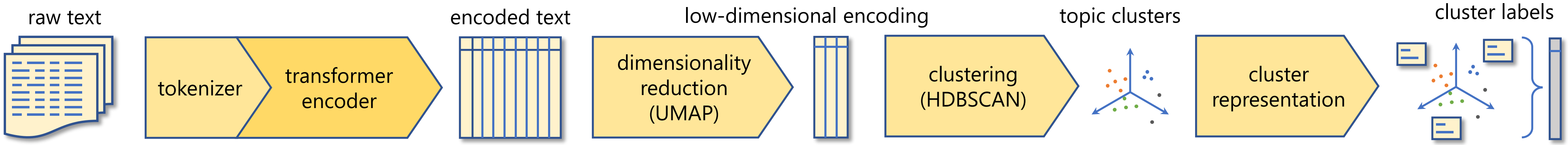
(2) Reduce dimensionality using UMAP ➔ vectors in $\mathbb{R}^d, d \ll E$

(3) Identify clusters using HDBSCAN

(4) Represent each cluster by the most frequent expressions, e.g. “hydrant”, “fire”, “hit”, “damaged”

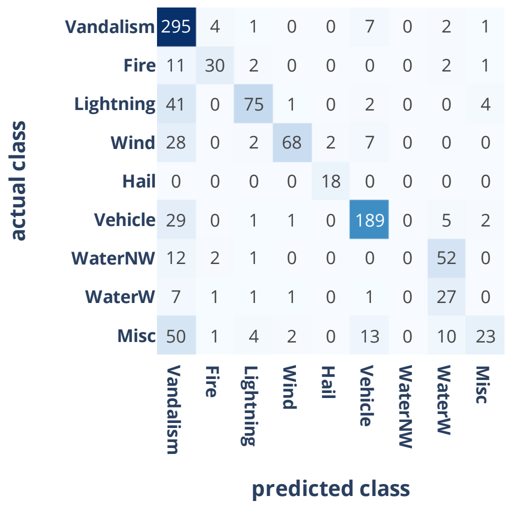
(5) Manually map each cluster to a label ➔ each sample is assigned the label of its cluster

Implementation: BERTopic



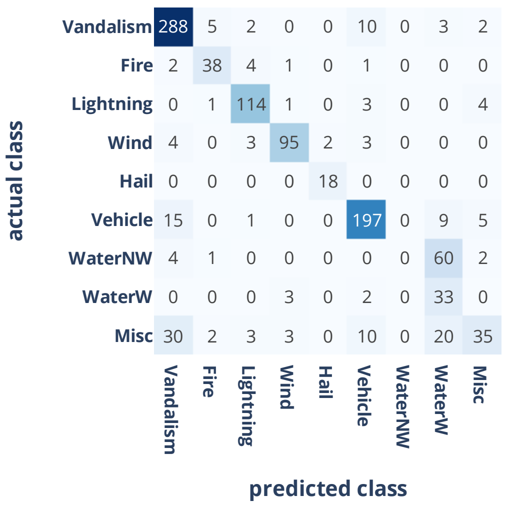
Results – Topic clustering

Topic modeling by clustering



actual class \ predicted class	Vandalism	Fire	Lightning	Wind	Hail	Vehicle	WaterNW	WaterW	Misc
Vandalism	295	4	1	0	0	7	0	2	1
Fire	11	30	2	0	0	0	0	2	1
Lightning	41	0	75	1	0	2	0	0	4
Wind	28	0	2	68	2	7	0	0	0
Hail	0	0	0	0	18	0	0	0	0
Vehicle	29	0	1	1	0	189	0	5	2
WaterNW	12	2	1	0	0	0	0	52	0
WaterW	7	1	1	1	0	1	0	27	0
Misc	50	1	4	2	0	13	0	10	23

Topic modeling by clustering, refined



actual class \ predicted class	Vandalism	Fire	Lightning	Wind	Hail	Vehicle	WaterNW	WaterW	Misc
Vandalism	288	5	2	0	0	10	0	3	2
Fire	2	38	4	1	0	1	0	0	0
Lightning	0	1	114	1	0	3	0	0	4
Wind	4	0	3	95	2	3	0	0	0
Hail	0	0	0	0	18	0	0	0	0
Vehicle	15	0	1	0	0	197	0	9	5
WaterNW	4	1	0	0	0	0	0	60	2
WaterW	0	0	0	3	0	2	0	33	0
Misc	30	2	3	3	0	10	0	20	35

			accuracy
	log loss	**Brier loss**	**score**
Dummy classifier	1.977	0.835	29.8%
Logistic regression classifier (supervised)	0.554	0.233	84.7%
1. Zero-shot classification	1.043	0.463	65.5%
… refined: mapping "Misc"	n/a	n/a	69.7%
2. Sentence similarity	n/a	n/a	74.5%
… refined	1.172	0.403	76.6%
3. Topic clusterig	n/a	n/a	69.9%
… refined	1.486	0.389	79.3%

- Differentiation of weather-related vs non-weather water claims difficult
- Refinement: Train a supervised sequence classifier using the labels from the unsupervised approach
- Accuracy score approaches that obtained by supervised approach!

Unsupervised Learning – Summary

- Able to handle situations with no or little labeled data
- Simple to implement

Conclusions

Conclusions

- Extract features from unstructured text data: sentence encoding and /or sequence classification
- Multilingual setting possible
- Able to handle very short and longer texts
- Unsupervised approaches to handle situations with no or few labels
- Transparency: We know which parts of the texts lead to a particular prediction

- Transfer learning: We have used very powerful NLP models which had been pre-trained on very large (multilingual) text corpora
- Minimal pre-processing and fine-tuning required
- Simple to implement thanks to the 🤗 Huggingface transformer library and model hub

Thank you!

Any questions or remarks?

Please do not hesitate to contact me!

Dr. Andreas Troxler, Actuary SAA, FCAS, CERA, PRM

andreas.troxler@atanalytics.ch

https://atanalytics.ch/en/