

What Kind of Emergency?

Providing Complaint Labels for Dutch Short Clinical Texts with Supervised & Explainable Machine Learning

Myrthe Reuver

Thesis project at Topicus

MSc “Cognitive Science & AI”

Introduction

Emergency Triage

- Huisartsenpost (emergency GP): phone call outside of regular doctor's hours, triage specialist does emergency triage.
- Steps in emergency triage (De Nederlandse Triage Standaard, 2020)
 - First: life-threatening (ABCD analysis)
 - Second: determining urgency with 5 labels (immediately needed or not?)
 - Together with: recording details in text, and **determining complaint label**
 - Last: determining follow-up actions (seeing doctor, or not)

Currently, Topicus uses **keyword-scanning** to give suggestions to triage specialists, who then pick 1 to 2 relevant complaints out of the **48 labels**.

Investigation: could NLP and Machine Learning improve this system?

S DA

Klacht/beloop: bij steek
Hulpvraag:
Voorgeschiedenis:
Medicatie:
Algemeen:

 Reanimatie

0

 Reanimatie:

Nee

Ja

 Disability

1

 AVPU:

A/alert

V/verbale reactie

P/pijnreactie

U/bewusteloos

 Airway

1


 Stridor:

Niet

Niet bedreigende stridor

Acute stridor

Geobstrueerd

 Breathing

0

 Kortademig, ernst:

Niet

Gering

Matig

Hevig

Ademt niet/nauwelijks

 Circulation

1

 Bloeding, uitwendig:

Nee/gering

Matig

Massaal

1

 Kleur:

Normaal

Rood

Bleek

Grauw

Blauw

ABCD veilig

Inklappen

Overlijden

Ingangsklachten

1



☐ Allergische reactie of insectensteek

[Meer ingangsklachten](#)

Research Question

“How can we best (in terms of **accuracy** and **flexibility**) automatically classify complaints from textual features in free text of triage assessments?”

- **accuracy**: more useful suggestions than keyword system;
- **flexibility**: use more features than merely keywords.

→ a multi-class, multi-label classification task

The Dataset

- **191.900 labelled text reports** of Dutch triage conversations from one emergency GP in the Netherlands, from 2016 to 2019.
 - out of 284.010 in the original dataset, due to missing labels (25% missingness not systematic)
- Table of invented information:

Complaint Label	Triage text	ABCD-safe	Time
‘Hoesten’ (coughing)	mevrouw hoest al de hele nacht, heeft zere borst (mrs. is coughing all night, has sore chest)	yes	11:10:53 05-03-2018
‘Allergische reactie of insectensteek’ (allergic reaction or insect bite)	moeder belt, jongen is gestoken door wesp (mother calls, boy is stung by bee)	yes	23:11:20 17-01-2017

Complaint labels: 11 needed preprocessing

	label before preprocessing	label after preprocessing
1	'Borstontsteking'	'Huidklachten/borstontsteking'
2	'Huidklachten'	'Huidklachten/borstontsteking'
3	'Trauma algemeen'	'Trauma algemeen/extremiteit'
4	'Trauma extremiteit'	'Trauma algemeen/extremiteit'
5	'Partus'	'Zwangerschap'
6	'Zwangerschap & vochtverlies'	'Zwangerschap'
7	'Zwangerschap & vaginaal bloedverlies'	'Zwangerschap'
8	'Zwangerschap & buikpijn'	'Zwangerschap'
9	'Zwangerschap & bezorgdheid'	'Zwangerschap'
10	'Zwangerschap & overige klachten'	'Zwangerschap'
11	'Zwangerschap & partus'	'Zwangerschap'

Related Work

Related work

- Another master student at Topicus (Kleverwal 2015), used **K-NN** with tf-idf transformed on same task, did not improve baseline on recall and F4.
- Earlier work (Beeksma 2017, Fivez, Šuster, and Daelemans 2017) → word embeddings for Dutch clinical text → capturing semantic information for classification or prediction
- Also: boosted decision tree (Swaminathan et. al. 2017) as explainable model for triaging English-language COPD patients in four health status classes. Find **several key variables** indicating health risk.
- Pre-trained clinical language models (Alsentzer et al. 2019). While most of these resources are in English, Fivez, Šuster, and Daelemans (2017) mention a model trained on 425 million words from Dutch clinical text, but they have not made it publicly available.

Data and Preprocessing

Dataset description

Conversations

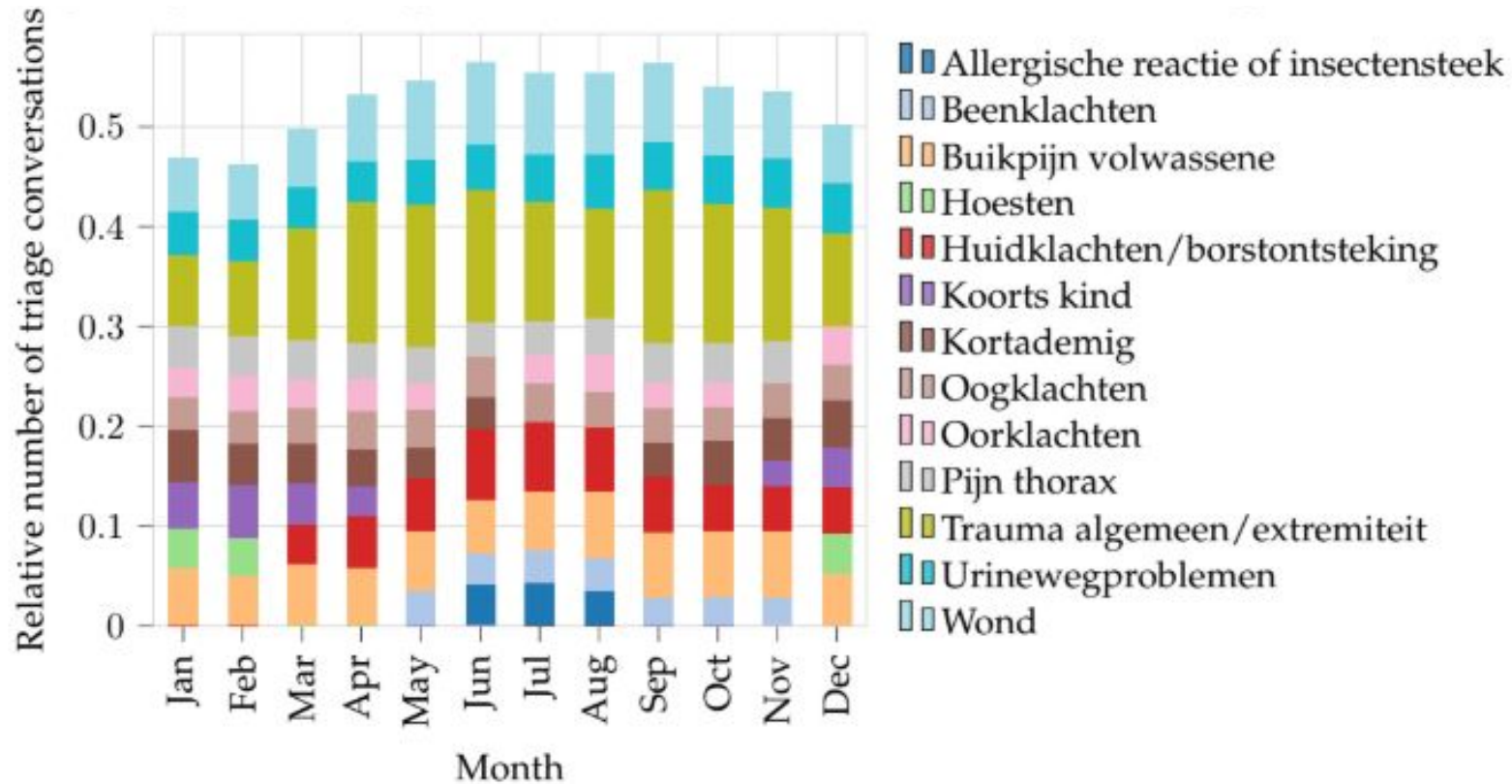
- Average 50.6 words (SD = 22.5), minimum: 2 words, maximum: 375 words.
- Short, time-pressure texts.
- Over entire corpus: 9.4 million words
- On average, a triage conversation uses 44.9 unique words (SD = 17.7)

Complaint labels

- Top 15 complaint labels (out of 48) account for 60% of all conversations. Most common complaint: 'Trauma Algemeen/extremiteit' (general trauma)
- M complaints: 1.13 (SD = 0.38), minimum: 1, maximum: 5.

Distribution of Complaints

Only different distributions per month, and time of day



Preprocessing I: labels

- NTS complaint labels change over the years, for uniform processing we changed these 11 complaint labels in the baseline system and in dataset:

	label before preprocessing	label after preprocessing
1	'Borstontsteking'	'Huidklachten/borstontsteking'
2	'Huidklachten'	'Huidklachten/borstontsteking'
3	'Trauma algemeen'	'Trauma algemeen/extremiteit'
4	'Trauma extremiteit'	'Trauma algemeen/extremiteit'
5	'Partus'	'Zwangerschap'
6	'Zwangerschap & vochtverlies'	'Zwangerschap'
7	'Zwangerschap & vaginaal bloedverlies'	'Zwangerschap'
8	'Zwangerschap & buikpijn'	'Zwangerschap'
9	'Zwangerschap & bezorgdheid'	'Zwangerschap'
10	'Zwangerschap & overige klachten'	'Zwangerschap'
11	'Zwangerschap & partus'	'Zwangerschap'

Train/test split + evaluation

- Predicting future cases:
 - 2019 test/dev set (50% stratified split on month and time of day)
 - 2016-2018 training set
- Baseline: the keyword-scanning system:
- comparing to several ML- based solutions

- Evaluation metric: recall and **F4 score**
$$F4 = \frac{(1 + 4^2) \cdot precision \cdot recall}{(4^2 \cdot precision) + recall}$$
- **Evaluation problem: labels based on current baseline (keyword scanning system).**

Preprocessing: Text Normalization

- Clinical text, especially triage text, has many:
 - misspellings ('reringvinger' → 'ringvinger')
 - abbreviations ('cva' → 'Cerebrovasculair Accident' or stroke)
 - synonyms ('growth' / 'tumor')
 - .. which can impede complaint classification.
- Normalization approaches:
 - > **Stemming** (Kleveral 2015): kanker, kankers > kank (but also: vallen -> gevall, uses no syntactic information).
 - > **Spell checking** (Beeksma 2017): fixes misspellings but not synonyms;
 - > **Lemmatization** uses syntactic information (PoS tags)
 - > **(pre-trained) word embeddings** also help in 'normalizing' semantic information

Results normalization, on base models & subset dev

- Lemmatization ($F1 = .64$, $prec = .68$, $rec = .60$) rivals, but does not beat stemming ($F1 = .64$, $prec = .69$, $rec = .60$) on the complaint classification task with K-NN (cf. Kleverwal 2015).
- Spell checking algorithms can be effective, but true reproduction of effective clinical spell checkers (cf. Beeksma 2017) exceeded our hardware capabilities.
- Most successful normalization method and text transformation by far: **word embeddings** (word2vec, Mikolov et al. 2013)

Text Representation: Word Embeddings

Word embeddings: assumption “similar words occur in similar contexts”, making numeric vectors more similar.

Also a manner of implicit, unsupervised text normalization.

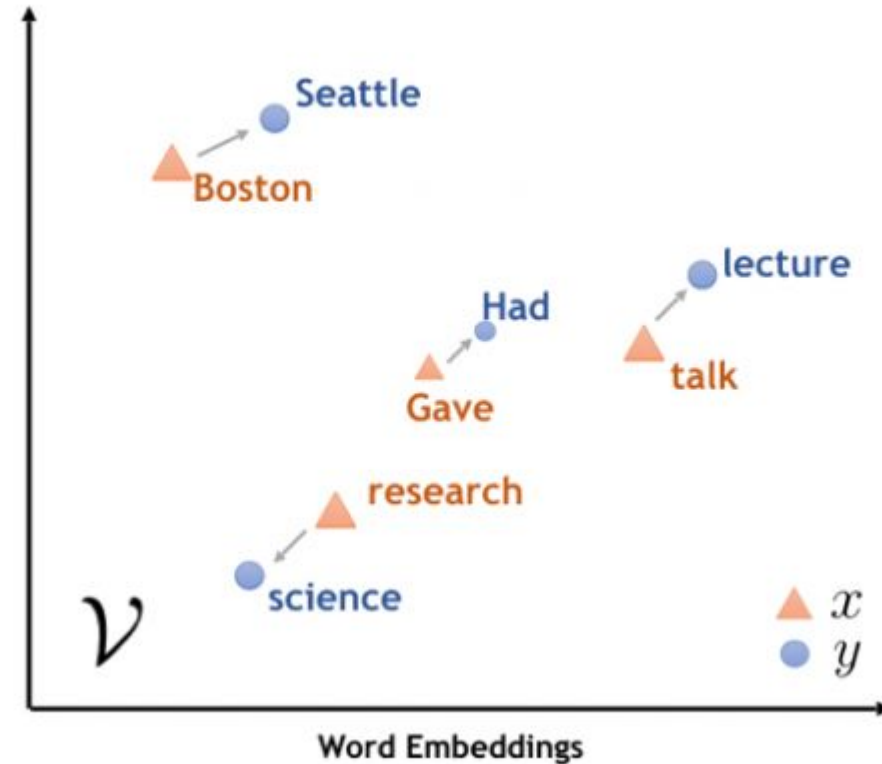


Image:

<https://www.ibm.com/blogs/research/2018/11/word-movers-embeddings/>

Trained Word Embeddings

- Trained on the training data (143,748 conversations, with approximately 6.7 million words), then transform all texts with the trained skip-gram word2vec model, removing OOV words.
- We systematically tested context windows size and minimal frequency for context: window = 20 with minimal frequency = 15 leading to best complaint classification.
- Large-scale pretrained model on 25.1 million Dutch words from CoNNL corpus (Kutozov, 2017) lead to significant performance loss ($F1 = < .50$). Fivez, Šuster, and Daelemans (2017) mention a model trained on 425 million words from Dutch clinical text, but not publicly available.

Semantic understanding of triage domain by trained model

input word	similar term <i>Top 5</i>	translation	cosine similarity
'val' (fall)	'sprong'	jump	0.59
	'struikelen'	to stumble/trip	0.59
	'ongeval'	accident	0.58
	'verstappen'	to misstep	0.56
	'stoten'	to bump	0.56
'pijn' (pain)	'pijnscheuten'	sharp pains	0.80
	'pijnklachten'	pain(complaints)	0.66
	'steken'	stinging pains	0.65
	'kramp'	cramp	0.61
	'drukpijn'	pressure pain	0.58
'hart' (heart)	'ritme'	(heart)rythm	0.65
	'hartje'	little heart (diminutive)	0.62
	'hartritme'	heart rate	0.62
	'hartritmestoornissen'	cardiac arrhythmias	0.55
	'hartkloppingen'	heart palpitations	0.55
'brandwond' (burn)	'brandwonden'	burns	0.81
	'brandblaar'	blister (burn-related)	0.80
	'blaar'	blister	0.78
	'brandplek'	burn(mark)	0.75
	'verbrand'	sunburn	0.74

Complaint Classification

Baseline: Topicus keyword scanner

- 830 keywords predicting 48 labels, no semantic or syntactic information
→ e.g. “val” (fall) is a keyword for trauma-related complaints,, but “gevallen” is not, neither is “smak” (colloquial word for “fall”).
→ Re-built from documentation with preprocessed labels: might thus not be 1-1 with current in-use system
- Two versions (**baseline & within-word keyword scanning**) on three datasets: development set, test set, and subset not predicted by baseline:

	Baseline		Baseline in-word keyword scanning		subset baseline	
	<i>dev</i>	<i>test</i>	<i>dev</i>	<i>test</i>	<i>dev</i>	<i>test</i>
Mean nr of suggestions	10.3	10.3	15.88	15.88	13.38	13.53
<i>SD</i>	(4.80)	(4.79)	(5.33)	(5.39)	(5.15)	(5.22)
Mean nr correctly predicted	0.94	0.94	1	1.00	1.03	1.06
<i>SD</i>	(0.53)	(0.53)	(0.50)	(0.50)	(0.17)	(0.16)

Results baseline

- performance is lower than in Kleverwal (2015) (recall of .87, precision of .23), but: newer system, other dataset, and we stratified and used multiple years;
- especially trauma-related complaints (“Trauma algemeen”, “Trauma Schedel”) improve from within-word keyword scanning (“val”).

	Baseline		Within-word scanning			
	<i>dev</i>	<i>test</i>	<i>dev</i>	<i>test</i>	<i>subset</i>	
					<i>dev</i>	<i>test</i>
precision	0.10	0.10	0.07	0.07	0.03	0.03
recall	0.82	0.82	0.88	0.88	0.30	0.31
<i>F1</i> -score	0.18	0.18	0.13	0.12	0.05	0.05
F-4 score	0.57	0.57	0.52	0.52	0.20	0.20

Heterogeneous classes and performance

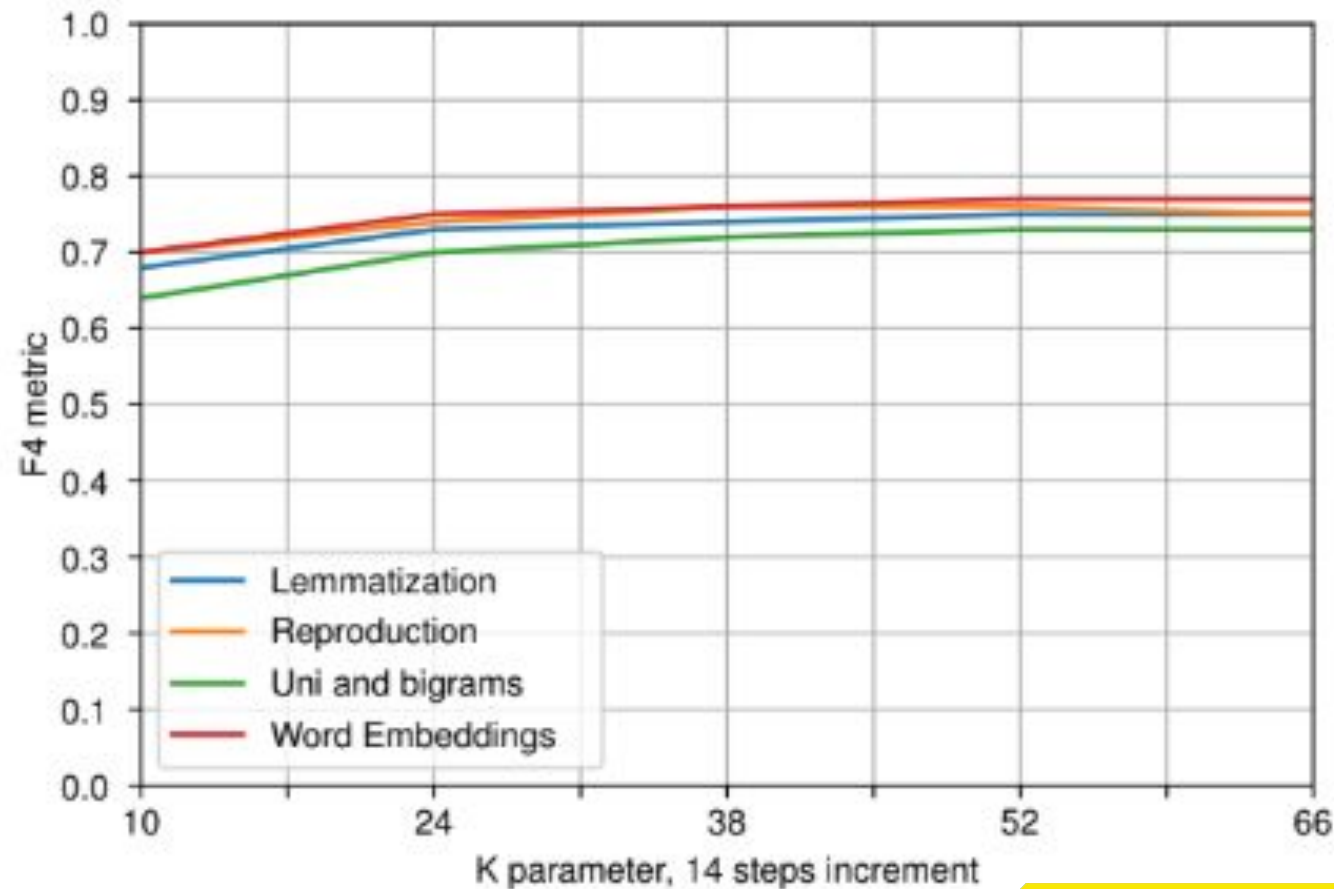
	>.90 in-class recall	< .60 in-class recall
1	'Algehele Malaise Volwassene' 'General Unwellness Adult'	'Corpus Alienum' 'Foreign object'
2	'Buikpijn Volwassene' 'Stomach Pain Adult'	'Diabetes' 'Diabetes'
3	'Armklachten' 'Arm Complaints'	'Intoxicatie' 'Intoxication'
4	'Trauma Algemeen/Extremiteit' 'General Trauma'	'Vreemd Gedrag of Suicidaal' 'Strange Behaviour & Suicide'
5	'Zwangerschap' 'Pregnancy'	'Oogklachten' 'Eye Complaints'

ML model I: K-NN, reproduction Kleverwal

Top parameter: number of probable unique labels on the nearest neighbours → increased recall.

Reproduction result ($F4 = .76$) is similar to Kleverwal (2015)'s reported score ($F4 = .75$)

Best model on dev:
WORD2VEC, cosine distance, **K = 52**, **Top = 7**, **F4 = .78**,
***M* (SD) suggestions = 5.1 (1.2).**



Model II: Random Forest

- Test two forms of tuning for optimizing recall over precision:
 - The Top function

data transformations	Top = 7				
	word embeddings				
<i>N trees</i>	200	400	600	800	1000
F-4 score	0.62	0.64	0.66	0.67	0.64

- Threshold function (from softmax output of random forest model over labels, suggest al labels above probability *thresh*)

Threshold	0.2	0.1	0.05	0.025	0.01
Mean (SD) suggestions	1.8 (0.5)	2.7 (0.9)	4.7 (1.7)	9.3 (3.6)	19.5 (7.6)
F-4 score	0.46	0.61	0.65	0.62	0.45

Model II: decisive features on keyword-less conversations

- Performance without the 830 keywords is lower (precision = .12, recall = .68) compared to with keywords (precision = .14, recall = .88), especially on recall.
- We find the most important features by averaging the decrease in impurity over the trees in our Random Forest. We then use a dictionary of tf-idf weighted vectors and terms to find the 100 most predictive unigrams.
- some highly predictive features that are not a keyword: '**moeder**' (**mother**), '**erg**' (**severe**) and '**rood**' (**red**). Some of these appear related to child-related complaints, others to wounds or skin problems.

All tuned model's performance on the test set

	Baseline rule-based	Adjusted Baseline in-word keyword scan	ML solution 1 K-NN	ML solution 2 Decision Tree
training data	raw texts	raw text	WORD2VEC	WORD2VEC
<i>Top</i>	-	-	7	7
specifications	-	-	cosine, $K = 52$	800 trees
precision	0.10	0.07	0.22	0.14
recall	0.82	0.88	0.92	0.88
<i>F4</i> -score	0.57	0.52	0.78	0.67
<i>M</i> suggestions (<i>SD</i>)	10.3 (4.8)	15.9 (5.4)	5.1 (1.2)	5.9 (0.9)

Discussion & future work

- Some classes (trauma-related classes) improved recall in within-word keyword scanning, without drastically increasing suggestions;
- pre-trained language models on specific domain texts (emergency triage): general Dutch (CoNNL model) did not perform well. How narrow should domain be?
- Top-function works better than threshold in optimizing for recall;
- We hoped to perform baseline-independent labelling, but was not possible → interaction with humans in gold standard/texts
- Concept drift: NTS changes over years, so possibly meaning of 'coughing' also chan

Conclusion

Answering our RQ

- “How can we best (in terms of **accuracy** and **flexibility**) automatically classify complaints from textual features in free text of triage assessments?”
 - Text normalization is key: Word2Vec transformed data especially can improve performance, but pre-trained on general Dutch do not;
 - ML with word2vec text transformations can outperform baseline and earlier studies;
 - Adjusted baseline can increase recall for some classes;
 - ML solutions are very dependent on the keywords (and thus baseline) but there are predictive non-keywords.

The End

UNSUPERVISED MACHINE LEARNING

SUPERVISED MACHINE LEARNING

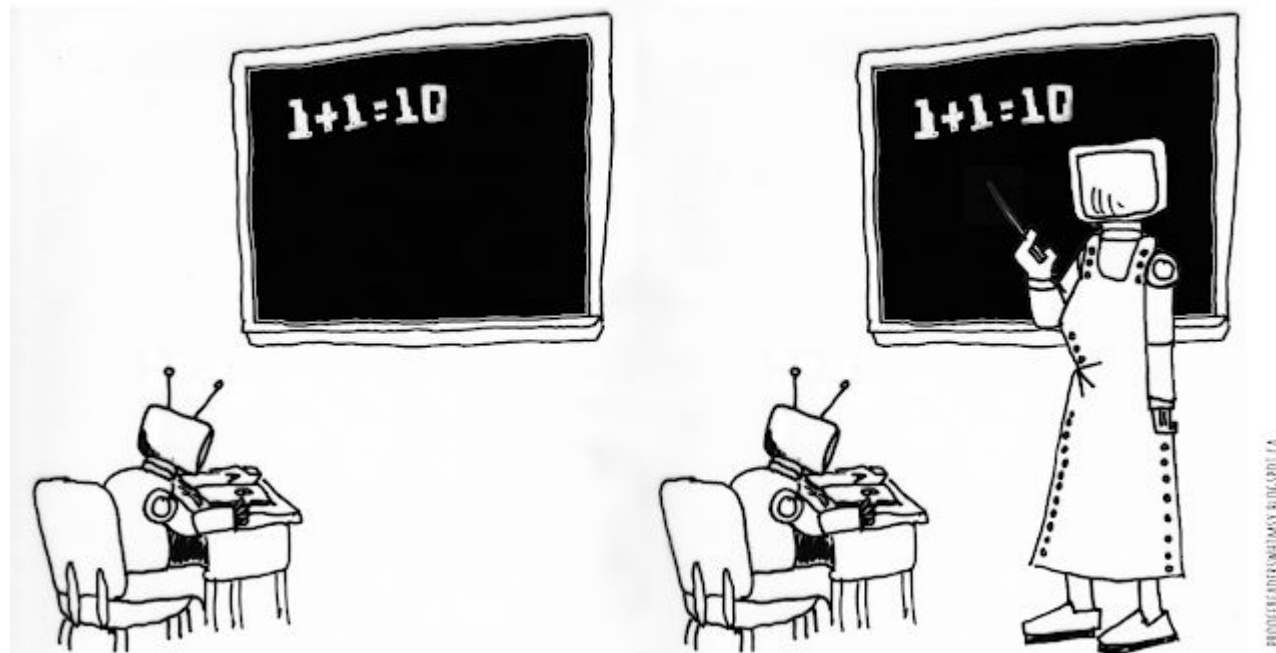


Image: <http://prooffreaderswhimsy.blogspot.com/2014/11/machine-learning.html>

References

- Alsentzer, Emily, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. page 72–78.
- Beeksma, Merijn T. 2017. Computer, how long have i got left? predicting life expectancy with a long short-term memory to aid in early identification of the palliative phase. Master's thesis, CLS, Radboud University.
- De Nederlandse Triage Standaard. 2020. Hoe werkt triage in nts?
- Fivez, Pieter, Simon Šuster, and Walter Daelemans. 2017. Unsupervised context-sensitive spelling correction of clinical free-text with word and character n-gram embedding. In 16th Workshop on Biomedical Natural Language Processing of the Association for Computational Linguistics, pages 143–148.
- Kleverwal, Jurgen. 2015. Supervised text classification of medical triage reports. Master's thesis, University of Twente.
- Swaminathan, Sumanth, Klajdi Qirko, Ted Smith, Ethan Corcoran, Nicholas G Wysham, Gaurav Bazaz, George Kappel, and Anthony N Gerber. 2017. A machine learning approach to triaging patients with chronic obstructive pulmonary disease. PloS one, 12(11).
- Kutuzov, Andrei, Murhaf Fares, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In Proceedings of the 58th Conference on Simulation and Modelling, pages 271–276, Linköping University Electronic Press.
- Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems, pages 3111–3119.

Results

	Base-text		stemmed-text		lemmatized-text	
	+ stop unchanged	-stop	+stop	-stop	+ stop	-stop
Rule-based						
micro- <i>F1</i>	0.18	0.23	0.17	0.23	0.18	0.23
precision	0.10	0.13	0.10	0.13	0.10	0.14
recall	0.85	0.84	0.70	0.69	0.82	0.81
Rule-based in-word						
micro- <i>F1</i>	0.13	0.14	0.12	0.14	0.13	0.14
precision	0.07	0.08	0.07	0.08	0.07	0.08
recall	0.89	0.89	0.81	0.81	0.88	0.87
K-NN + tf-idf, $K=19$						
micro- <i>F1</i>	0.64	0.63	0.64	0.61	0.64	0.63
precision	0.68	0.68	0.69	0.68	0.68	0.68
recall	0.60	0.59	0.60	0.60	0.60	0.60
Random Forest, trees = 600						
micro- <i>F1</i>	0.59	0.60	0.58	0.60	0.58	0.60
precision	0.63	0.64	0.62	0.65	0.62	0.64
recall	0.55	0.56	0.55	0.57	0.55	0.56