# Towards Anomaly Detection in EIOS: Natural Language Processing and Supervised Learning Can Help Detect Signals

2019 EIOS Global Technical Meeting
Seoul, 13 November 2019

Stéphane Ghozzi
Signale, Robert Koch Institute, Berlin
DVA, World Health Organization, Geneva

ghozzis@rki.de

# Outline

# A labeled dataset

learn from the experts in the DVA team of WHO

a binary classification: 1 article is "signal" or "not signal"

signals = URLs in signals list + Ebola alerts compiled by DVA team $\Longrightarrow$ **labels**

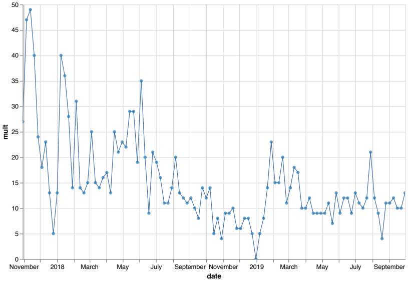
articles = EIOS, 2 boards followed by DVA, in English $\Longrightarrow$ **data**

time ranges:
signals: 1 Nov 2017 - 29 Sep 2019
EIOS: 1 Nov 2017 - 31 Aug 2019

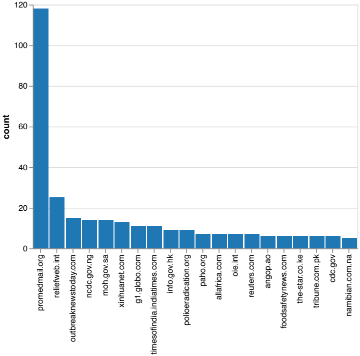
# Signals

- w/o Ebola alerts: 3,499 signals, of which 861 have 1 or more “media” URLs

weekly count




web sites (top 20 of 520)




- 1,315 Ebola alerts, of which 22 have 1 or more “media” URLs

## EIOS articles

Sequentially:

- remove duplicate URLs, keeping the oldest ones
- keep only texts with at least 30 Latin letters
- keep only articles in one of the two boards followed (if not signal)
- keep only texts in English (using `langdetect()`)

$\Longrightarrow$ 492,036 - 9,617 + 1 = **482,420 articles**

that's an average of 722 articles/day

# Matching signals / EIOS

Of 932 unique signal URLs, 274 could be matched to EIOS, of which 20 were removed

$\Longrightarrow$ **254 articles labeled "signal"**

Looking at signals with 7 days delay: 896 signals
- of those: 245 have **web site** not in the EIOS dataset, most not English
- of the 375 w/ web site in EIOS but not matched, **manual inspection** of 100 (in the top 10 domains): no error in matching, rather language is not English or were presumably not categorised in the boards

Memory + balancing: **random sample: 10%** of EIOS that are not signals

$\Longrightarrow$ **48,217 articles labeled "not signal"**

# Data processing

## Vectorisations

= ways of translating texts into numbers

1. **Bag-of-words,** with tf-idf:
   1 text $\sim$ frequencies of its words, with overall frequencies in corpus discounted

2. **Word embeddings,** with Word2vec (Google News corpus, 3m words):
   1 word $\sim$ vector in "semantic space" 300-dimensional representation
   1 text $\sim$ mean of the embeddings of its words

Example of **word embeddings:**

Coordinates of "Ebola":

```
> [0.065, -0.0048, 0.030, 0.11, -0.065, 0.0081, -0.11, -0.059, 0.045,
-0.043 ... ]
```

Words most similar to "Ebola":

```
> [('Ebola_virus', 0.78), ('Marburg_virus', 0.75), ('Ebola_outbreak',
0.70), ('haemorrhagic_fever', 0.69), ('Ebola_fever', 0.69), ('ebola',
0.68), ('Marburg_hemorrhagic_fever', 0.67), ('Ebola_hemorrhagic_fever',
0.67), ('Marburg_fever', 0.67), ('Ebola_haemorrhagic_fever', 0.67)]
```

## Text preprocessing

sentence and then word **tokenisation**

keep only **Latin letters** (accents included), **digits**, and **dots**

remove **stop words**

token processing:

- **tfidf:** remove dots, numbers, accents; lower case; lemmatisation; stemming
- **w2v:** replace digits with “#”

keep tokens with **2 or more characters**

train **bi- and trigrams**

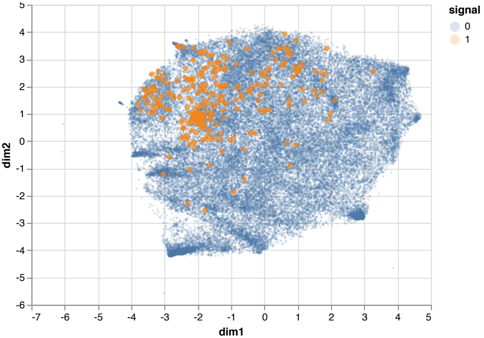
```
> trigram_simple_pp[bigram_simple_pp[['human','immunodeficiency','virus']]]
> ['human_immunodeficiency_virus']

> trigram_simple_pp[bigram_simple_pp[['human','immunodeficiency','apple']]]
> ['human_immunodeficiency', 'apple']
```

# Data exploration

## Sentiment and topics

**quick and dirty...** Nothing much

## 2d visualisations of embeddings (t-SNE)



# Different approaches

## Training and test datasets

**1 partition** training / test sets (80% / 20%)

add **reduced tfidf** (~PCA, 300 components) to the 2 vectorisations

**upsampling** of training data:
- none
- duplicate
- ADASYN (linear interpolation)

**standardisation:**
- none
- standardise (tfidf: not centred because sparse)

all transformations trained on training set, then applied to training and test sets

# Classification algorithms

- complement naive Bayes
- logistic regression
- multilayer perceptron
- random forest
- support vector machine (non-linear)

overall
$(5 \text{ algorithms}) \times (3 \text{ vectorisations}) \times (3 \text{ upsamplings}) \times (2 \text{ standardisations}) - 1 \times 2 \times 3 \times 2$
approaches

$\Longrightarrow$ **78 approaches** to test

CNB needs positive features: no w2v and no reduced tfidf

# Classification performance

**Output** of the algorithms: for each article, **probability of being "signal"**

**Threshold** $t$:
- if $p(\text{signal}) \geqslant t$, then prediction = "signal",
- else prediction = "not signal"

For each $t$:
**confusion matrix = (# true negatives, # false positives, # false negatives, # true positives)**

**Scores** (computed from the confusion matrix):
accuracy / recall (sensitivity) / specificity / precision / F1 / Matthews correlation coefficient / balanced accuracy / geometric mean / index balanced accuracy of the geometric mean

**Scores** (threshold independent):
- AUC / Relative probability gap

ba = average of recall obtained on each class
geom_mean = root of the product of sensitivity and specificity
rel_p_gap $= 2(\mu(p_{\text{signal}}) - \mu(p_{\text{not signal}}))/(\sigma(p_{\text{signal}}) - \sigma(p_{\text{not signal}}))$

# Best scores with $t$ / recall $\approx 0.9$

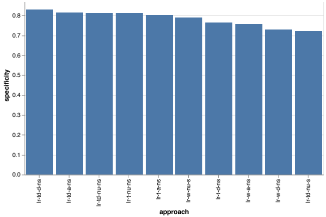
**Logistic regression / reduced tfidf / duplicate / no standardisation**

is best along all scores...

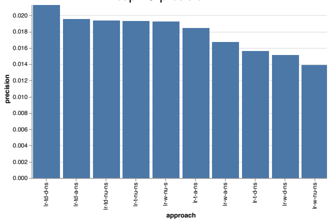
| | |
|---|---|
| **accuracy** | 0.83 |
| **precision** | 0.021 |
| **specificity** | 0.83 |
| **f1** | 0.042 |
| **mcc** | 0.13 |
| **ba** | 0.88 |
| **geom_mean** | 0.87 |
| **iba_gm** | 0.76 |

... but it's a tight race...

top 10 specificity




top 10 precision




**confusion matrix = (TN 7999, FP 1657, FN 3, TP 36)**

# Conclusion and outlook

1 approach stands out at high recall (sensitivity):
**TN 7999, FP 1657, FN 3, TP 36**
i.e. to find (more than) 36 of the 39 signals, just read ~1,700 articles out of ~9,700

Already works well and could be helpful:
no automatisation, but **ranking**

**Low precision** and F1... are maybe OK:
there might be hidden or discarded signals

Many signals lost, mostly because not in English

## Immediate tasks

Combination with **“noise”** (cf. Émilie Péron and Scott Lee)

Use **all available articles,** not just a sample

Proper **cross-validation,** hyperparameter **optimisation**

**Manual inspection** of predicted positives

Apply similar analysis to **events (in EMS)**

## Perspective

**EIOS meta-data:**
- not seen / title read / text read / article pinned / article flagged
- signals / (risk) assessment

**Beyond English:**
- automatic translation (is being used by experts!)
- language-specific analyses

**Context:**
- as supplementary features for classification

**Fancier approaches:**
- Stacking (combination of approaches)
- Transfer learning of word embeddings, document embeddings, transformer models...
- Deep learning

**Web application:**
- prototypical implementation in an interactive dashboard
- evaluation of usefulness (with new, unfiltered data)

**Computation** infrastructure

# Thank you!

Acknowledgements:

- Sooyoung Kim, Annika Wendland (WHO/DVA)
- Philip Abdelmalik, Émilie Péron, Johannes Schnitzler (WHO/DVA)
- Sandra Beermann, Andreas Jansen (RKI/INIG)
- Auss Abbood (RKI/Signale)

Similar work done at RKI:
Abbood et al (2019) medRxiv, https://doi.org/10.1101/19006395

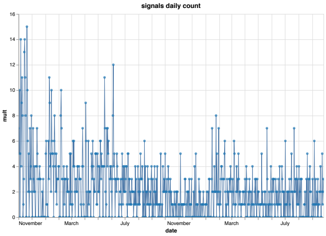
SIGNALE

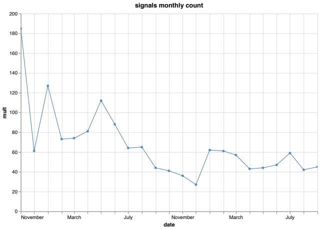
signale@rki.de

rki.de/signale-project

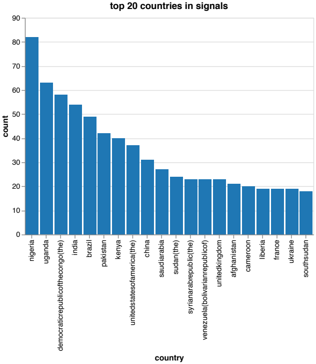
# Supplementary Information

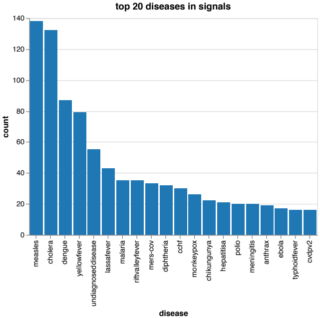
## Signals (w/o Ebola alerts)

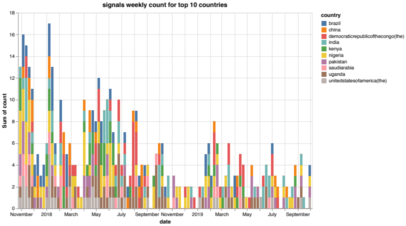


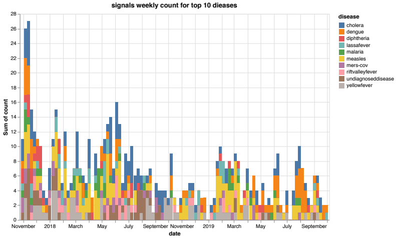



# Signals (w/o Ebola alerts)



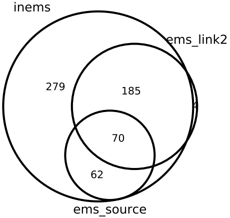



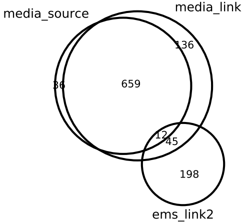



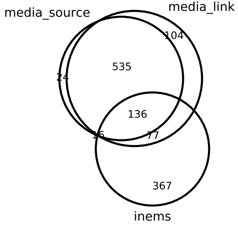



# Signals (w/o Ebola alerts)

media and EMS links



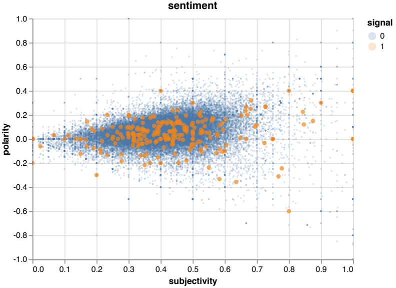
## Word2vec trained on Google News, examples:

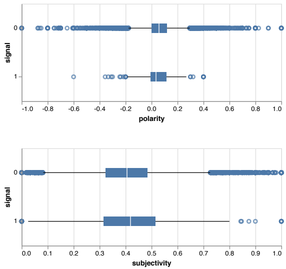
```
> w2v.vectors_norm[w2v.vocab['HIV'].index]
> [-0.027214931, 0.005086286, -0.00077202555, -0.024440594, -0.061563876, -0.0069028167, -0.04993808, 0.028800268,
-0.024704818, -0.03778384 ... ]

> w2v.most_similar('HIV')
> [('HIV_AIDS', 0.8241558074951172), ('HIV_infection', 0.8100206851959229), ('HIV_infected', 0.782840371131897),
('AIDS', 0.763182520866394), ('HIV_Aids', 0.7069978713989258), ('HIV_AIDs', 0.7062243223190308), ('Hiv',
0.6802983283996582), ('human_immunodeficiency_virus', 0.6724722981452942), ('Aids', 0.6655842065811157), ('H.I.V.',
0.6647853255271912)]

> w2v.vectors_norm[w2v.vocab['influenza'].index]
> [0.015480349, 0.00036750827, 0.023640532, 0.04224095, 0.008460191, -0.015480349, -0.08640195, -0.03648082,
0.058801327, -0.027600622 ... ]

> w2v.most_similar('influenza')
> [('flu', 0.8435951471328735), ('H#N#', 0.8313145041465759), ('H#N#_influenza', 0.8289912939071655),
('H#N#_virus', 0.8022348880767822), ('seasonal_influenza', 0.8018087148666382), ('H#N#_flu', 0.7963185906410217),
('Influenza', 0.7937184572219849), ('H#N#_influenza_virus', 0.7823264598846436), ('flu_virus', 0.7783315181732178),
('influenza_virus', 0.7776930332183838)]

> w2v.vectors_norm[w2v.vocab['H#N#'].index]
> [0.040303856, -0.08500449, 0.014717014, 0.027357768, -0.03615134, 0.020884724, -0.085981555, -0.023327382,
0.043479312, 0.0054959804 ... ]

> w2v.most_similar('H#N#')
> [('H#N#_virus', 0.9167306423187256), ('H#N#_flu', 0.8859533071517944), ('swine_flu', 0.8520038723945618),
('H#N#_influenza', 0.850509524345398), ('influenza', 0.8313145041465759), ('H#N#_swine_flu', 0.8082534074783325),
('bird_flu', 0.7901098728179932), ('H#N#_influenza_virus', 0.7855583429336548), ('avian_influenza',
0.7841204404830933), ('H#N#_strain', 0.7841016054153442)]
```

**Quick and dirty:**

# Sentiment

"polarity" = negative to positive sentiment



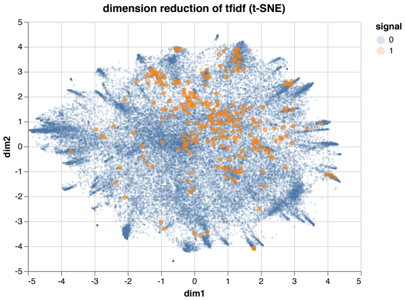



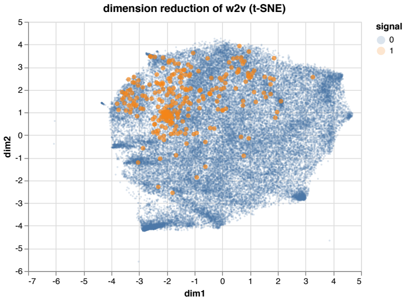

# Topics

"topic modelling" ~ clustering of bag-of-words

Nothing meaningful

# 2d visualisations (t-SNE)

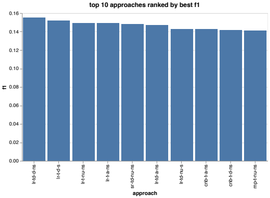


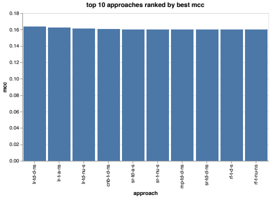



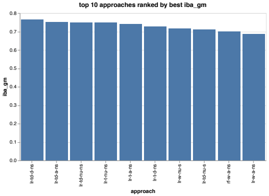

tfidf first reduced to 300 components (~PCA)

# Best scores achieved with varying $t$

| score_type | score_value | approach | confusion_matrix |
|---|---|---|---|
| **f1** | 0.15 | logistic_regression-tfidf_dr-duplicate-no_st | TN 9576 / FP 80 / FN 29 / TP 10 |
| **mcc** | 0.16 | logistic_regression-tfidf_dr-duplicate-no_st | TN 9576 / FP 80 / FN 29 / TP 10 |
| **ba** | 0.88 | logistic_regression-tfidf_dr-duplicate-no_st | TN 7999 / FP 1657 / FN 3 / TP 36 |
| **geom_mean** | 0.87 | logistic_regression-tfidf_dr-duplicate-no_st | TN 7999 / FP 1657 / FN 3 / TP 36 |
| **iba_gm** | 0.76 | logistic_regression-tfidf_dr-duplicate-no_st | TN 7999 / FP 1657 / FN 3 / TP 36 |
| **auc** | 0.92 | logistic_regression-tfidf_dr-adasyn-no_st | None |
| **rel_p_gap** | 1.75 | logistic_regression-w2v-duplicate-no_st | None |



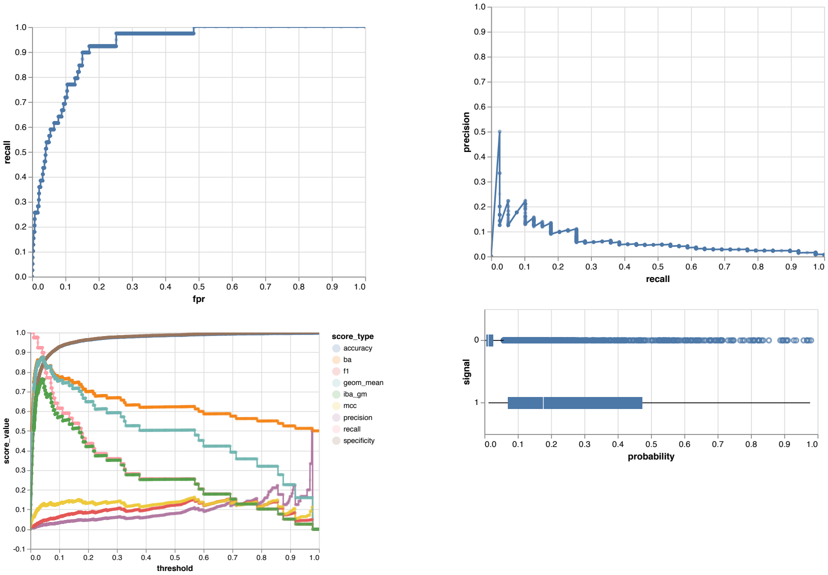




recall of 1 resp. specificity of 1 can always be achieved with $t = 0$ resp. $t = 1$

best accuracy and precision not meaningful (~no positives)

# Logistic regression / reduced tfidf / duplicate / no standardisation




fpr = 1 - specificity

Apply similar analysis to **events (in EMS)** and not just signals:

- "event" defined as disease + country + time range → **collection of articles**
- match with EMS database
- predict **(risk) assessments**

  `IHR Assessment` (0/1), `Serious Public Health Impact (WHO)` (0/1), `Unusual or Unexpected (WHO)` (0/1), `International Disease Spread (WHO)` (0/1), `Interference with international travel or trade (WHO)` (0/1)

  `RRANationalRiskLevel` (0/1/2/3/4), `RRARegionalRiskLevel` (0/1/2/3/4), `RRAGlobalRiskLevel` (0/1/2/3/4)
- events and signals partially **linked**
- labeled datasets **already prepared!**