Introduction to Text Mining for Auditors

The Cool Stuff of Natural Language Processing and Machine Learning

Jan Roar Beckstrom
deputy director general/chief data scientist
Office of the Auditor General of Norway

Parts to be covered

Part I: About the OAGN Innovation Lab

Part II: Text Mining & Natural Language Processing - Some Basic Topics

Part III: Text Mining in the Form of Search

10 minute break

Part IV: Classifying Criminal Cases by Using Machine Learning on Text

Part I

About the OAGN Innovation Lab

The Innovation Lab – What it is

innovation lab

(ˈɪnəʊˈveɪʃən læb ə)

A semi-autonomous organization that engages diverse participants—on a long-term basis—in open collaboration for the purpose of creating, elaborating, and prototyping radical solutions to pre-identified systemic challenges.

Source: Gryszkiewicz, Toivonen, Lykourentzou (2018) Innovation Lab: 10 Defining Features. Stanford Social Innovation Review

http://www.socialinnovationacademy.eu/project/innovation-lab-definition/

Why was the Innovation Lab established?

- More innovation
- Free up time for the auditors to do more analysis
- Automation of audit – both possible and desirable
- Other SAIs were pulling ahead on data science (notably UK and Netherlands) – we wanted to follow
- New opportunities in data analytics with the developments in machine learning and computing power

The Innovation Lab - Tasks

1. Curating/wrangling data for financial and performance audit
2. On-demand data analytics
3. Develop small bespoke webapps for use in financial audit analytics
4. Develop custom search apps
5. Promoting data science and the use of machine learning at the OAGN
6. Experiment with new (cloud) technologies and methods

Where we come from

- 2 political scientists
- 1 economist
- 1 sociologist
- 1 physicist

(started out with just 3 people)

… we're not an IT outfit, we're a data science outfit

Our slogan:

We automate the boring stuff,
so you can audit the exiting stuff!

Success factors for Innovation Labs

- Freedom to experiment
- Recruit people from audit, not IT
- No detailed planning — only the “what”, not the “how”
- Full support from top management
- Short development cycles #agile development
- Prioritise solutions to long standing issues

Some more success factors

- Knowledge on both audit and technology is important
- An entrepreneurial mindset is probably even more important
- A lot of useful technology is open source and (almost) free – use it
- Fix concrete problems for the auditors – this builds credibility
- Don't start with big structural problems – it will never fly
- Make *something*, and sell what you have

Part II

Text Mining & Natural Language Processing

Some Basic Topics

A. M. Turing (1950) Computing Machinery and Intelligence. *Mind 49:* 433-460.

COMPUTING MACHINERY AND INTELLIGENCE

By A. M. Turing

1. The Imitation Game

I propose to consider the question, $\boxed{\text{"Can machines think?"}}$ This should begin with

NLP – What is it?

- The Wikipedia definition:

Natural language processing (NLP) is (...) concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data.

The result is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them.

NLP techniques

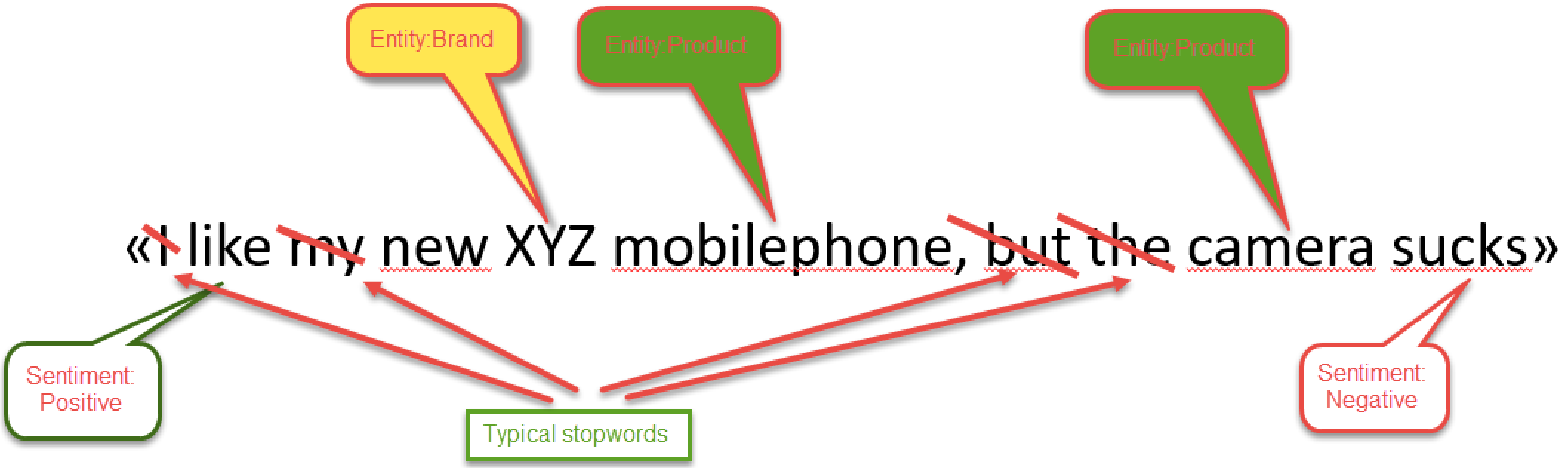
Some examples:

- Sentiment analysis (positive or negative?)
- Topic modelling (Topic A, B or C?)
- Text classification (E-mails: Spam or not-spam?)
- Named Entity Recognition (NER) (organisation or person?)
- Search/Retrieval

Pervasive use in consumer tech (chatbots, cell phones, search engines, Siri/Google Home, spam filters, plagiarism checks, market/customer analysis etc. etc.)

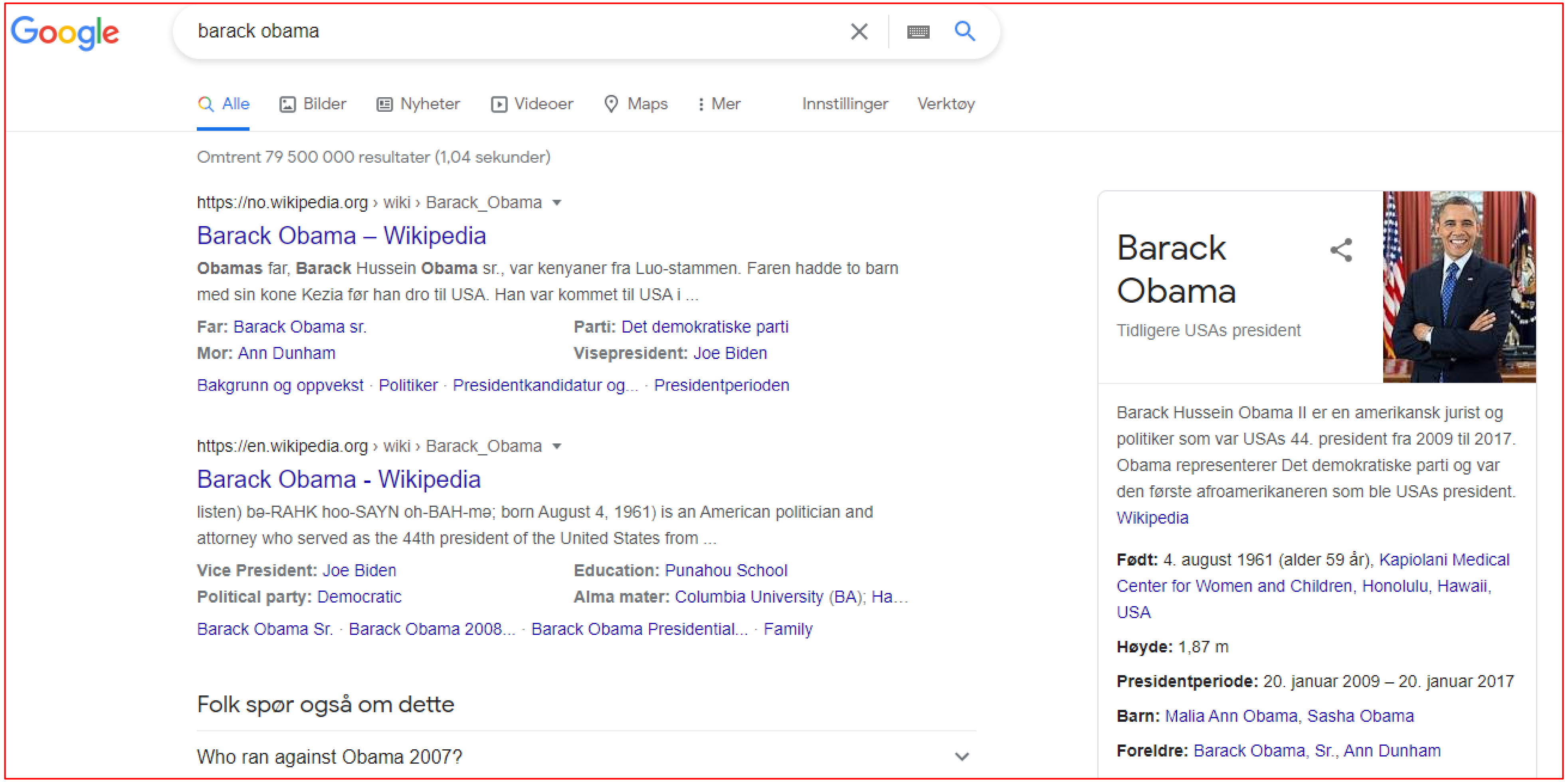
Example: NER & Sentiment Analysis

- Often used in analysis of customer reviews



NER at work

Google «understands» that the keywords «Barack» «Obama» is an «Entity» in the form of a «Person» - gives better search results



NLP - how text is handled

- Basically and highly simplified:
  - Corpus - a collection of text documents
  - «Word» vs. «Term»
  - Each word in a corpus is a data point, each term a variable
  - Prevalence → Importance
  - Lexica of positive/negative terms, synonyms
  - Stopwords

Text pre-processing (almost always)

- Converting Text (all letters) into lower case
- Removing HTML tags
- Expanding contractions
- Converting numbers into words or removing numbers
- Removing special character (punctuations, accent marks and other diacritics)
- Removing white spaces
- Word Tokenization
- Stemming and Lemmatization
- Removing stop words, sparse terms, and particular words

The concept of N-grams

- Tokenization → single words
- Context?

- N-grams → sequence of *n* textual elements

- (“happy” (unigram) vs. “not happy” (bi-gram)
- The sentence “I am not happy” has the following bi-grams:
- I am – am not – not happy

- What if you do sentiment analysis and use only single terms?
- And «not» is a stopword and thus removed?

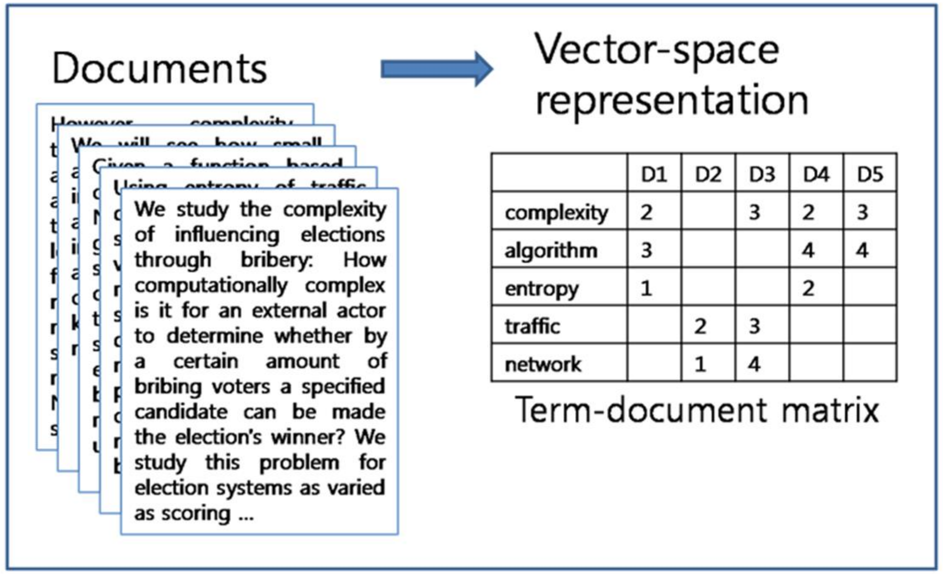
…and then you count…

- Bag-of-words

Document	the	cat	sat	in	hat	with
the cat sat	1	1	1	0	0	0
the cat sat in the hat	2	1	1	1	1	0
the cat with the hat	2	1	0	0	1	1

https://towardsdatascience.com/a-simple-explanation-of-the-bag-of-words-model-b88fc4f4971

- Document Term Matrix



	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

TF-IDF (a measure of importance)

- Term Frequency
  - how often do the term occur in a document
- Inverse Document Frequency
  - log(total number of documents/number of documents containing the term)
- TF-IDF → tf x log(N/df)

Example 1: Sentiment Analysis by the ECA

- Simple question:

- Do the ECA press releases tend to be more negative than their reports, perhaps as a way of making the headlines?

- Data material:

129 special reports, 27 reviews, 12 opinions and 30 audit previews + associated press releases

https://medium.com/ecajournal/spinning-negative-messages-a-closer-look-at-the-tonality-of-eca-audit-reports-and-press-releases-42b285c15a97

Tonality – bad news sell?

Spinning negative messages? A closer look at the tonality of ECA audit reports and press releases

European Court of Auditors Following
Feb 10, 2020 · 14 min read

Research has shown that 'negativity' is one of the main criteria that determine newsworthiness: 'bad news sells'. This is reflected in our daily media consumption. To a certain extent, it also influences the way corporate communication operates to attract journalists' attention. This poses a specific challenge for public auditors, whose mandate is to look at risks and identify shortcomings. From the start,

Main results

- The tonality of ECA publications is slightly on the positive side
- Press releases contain more tonality information (‘subjectivity’) than the full publications
- The tonality score of ECA publications correlates well with the tonality score of press releases.

Example 2: Using REGEX

- Regex (regular expressions)
  - a programming tool
  - a host of functions to search for patterns in text
  - available in all programming languages

- (Very simple) example from regex in Linux:

  The regex command ***grep*** *bash /etc/passwd*
  will output all lines containing the word «bash»
  from the file passwd, in the etc folder

A laundromat for text, using REGEX

github.com/navikt/laundromat

nomy News --... Dashboard - Micros... Google Microsoft Demos Overview - Microso... Automating infrastr... Velkomsttavle Trello bookdown Authori...

README.rst

What Is Laundromat?

Laundromat is a python package for scrubbing personal information from text. The current Laundromat is adapted to the Norwegian context, but can easily be modified to fit any language.

Laundromat uses regular expressions coupled with a machine learning model to recognize and remove personal information in norwegian text. The package is built around SpaCy. Current functionality of the package includes:

- Detecting named entities related to personal and sensitive information.
- Censoring texts containing said information.
- Improving the NER model from new data.
- Scoring the model with various metrics.

Supported entities:

- <PER> - Person
- <DTM> - Date-time
- <TLF> - Telephone number
- <FNR> - Norwegian Personal Number
- <AMOUNT> - Amounts like percentages
- <LOC> - Location, i.e. countries, cities, and addresses.
- <CREDIT_CARD> - Credit card number

SpaCy

spacy.io

Astronomy News --... Dashboard - Micros... Google Microsoft Demos Overview - Microso... Automating infrastr... Velkomsttavle Trello bookdown Authori... general Riksrevisjo... general SAI_Data_S...

Out now: spaCy v3.0

USAGE MODELS API UNIVERSE

Industrial-Strength Natural Language Processing

IN PYTHON

Get things done

spaCy is designed to help you do real work — to build real products, or gather real insights. The library respects your time, and tries to avoid wasting it. It's easy to install, and its API is simple and productive.

GET STARTED

Blazing fast

spaCy excels at large-scale information extraction tasks. It's written from the ground up in carefully memory-managed Cython. If your application needs to process entire web dumps, spaCy is the library you want to be using.

FACTS & FIGURES

Awesome ecosystem

In the five years since its release, spaCy has become an industry standard with a huge ecosystem. Choose from a variety of plugins, integrate with your machine learning stack and build custom components and workflows.

READ MORE

What's spaCy?

spaCy is a **free, open-source library** for advanced **Natural Language Processing** (NLP) in Python.

If you're working with a lot of text, you'll eventually want to know more about it. For example, what's it about? What do the words mean in context? Who is doing what to whom? What companies and products are mentioned? Which texts are similar to each other?

spaCy is designed specifically for **production use** and helps you build applications that process and "understand" large volumes of text. It can be used to build **information extraction** or **natural language understanding** systems, or to pre-process text for **deep learning**.

So, what do you need to get started?

- An old laptop (almost free)
- Python or R (free)
- NLP libraries/frameworks (free)
- Knowledge (lots and lots of free NLP tutorials/material online)

Part III – Text Mining in the Form of Search

PDF – Where Information Goes to Die

- PDF
  - A brilliant format for archives
  - The whole point is that a document should be unalterable
  - Terrible if you want to do analysis
  - What if you want to do fast full text search in 10 000 PDF documents?

Example 3: 35 000 PDFs – what to do

- We have 35 000 PDF documents from Norwegian hospital boards

- 25 hospital boards, 12-14 board meetings a year, last 6 years

- All documents published on the web (25 different websites)

- How to make this searchable in full text

We built a pipeline where…

- Hospital documents are now
  - automatically collected from the web (webscraped once a week)
  - ingested into a specific database type, inc. PDF → JSON (Elasticsearch)
  - made searchable in a webapp interface (built with R & Shiny)

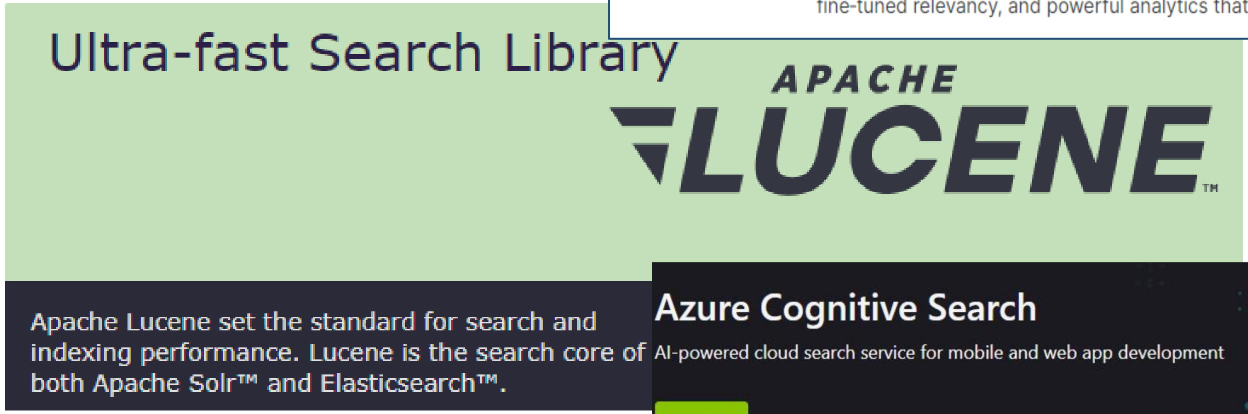
- It takes us now approx a week to develop a new search app for a specific use case

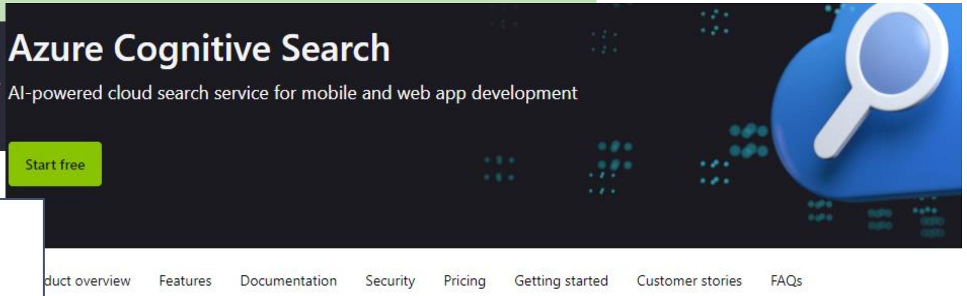
About Elasticsearch (elastic.co)

- Open source search & analytics engine
- Designed for text search
- Fast and relevant full text search – scales extremely well
    - Relevant results based on TF-IDF: how **prevalent** is the search word in the document, and how **unique** is the word across documents
    - 1 000 or 100 000 documents - search takes normally < 0.5 seconds
- Uses stemming and synonym dictionaries

Useful Search Technology

- Elasticsearch
- Azure Cognitive Search
- Apache Lucene
- AWS CloudSearch





Example 4:

80 000 pages of text in the form of .jpg pics

How to build a custom search engine

The case in question

- A performance audit needed to analyse 80 000 pages of material from the 1980s – written on typewriter and scanned to .jpg

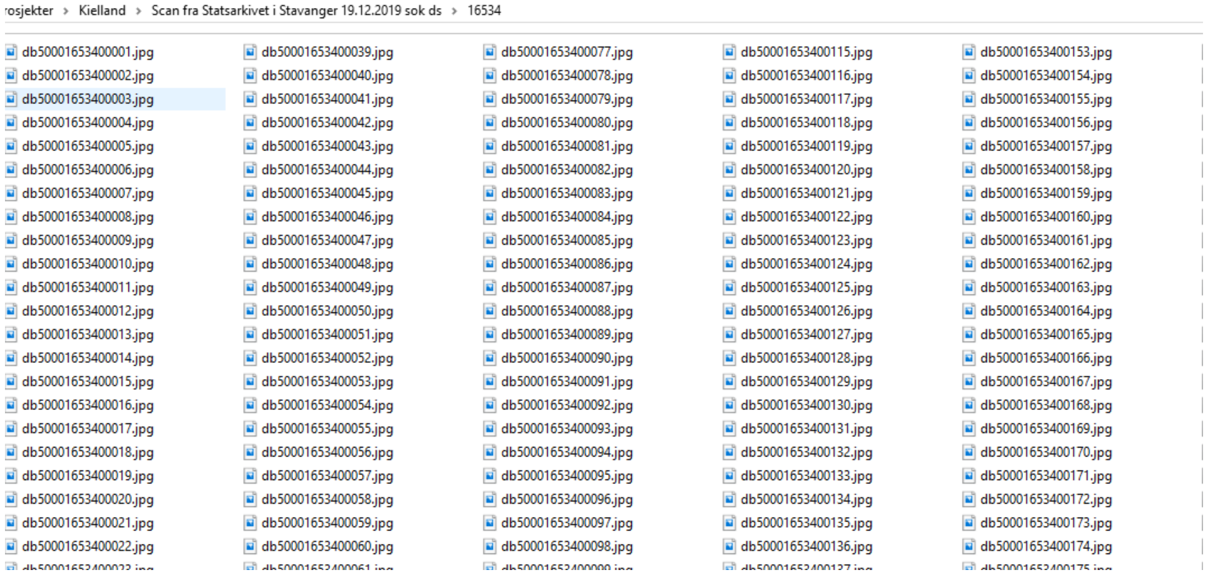
- Audit topic:

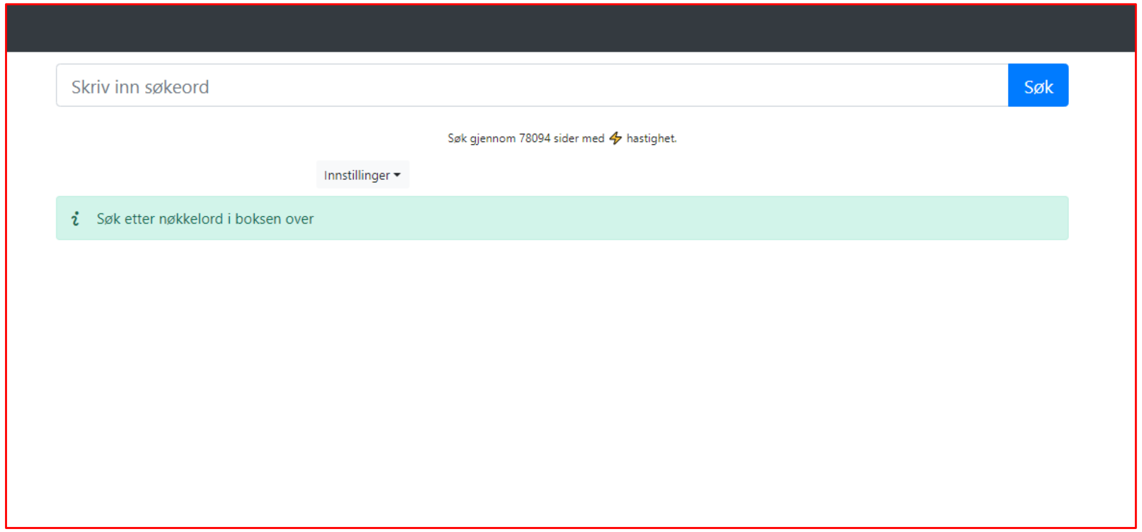
An offshore oil rig disaster in 1980

123 people died

A national trauma ever since

From This TO This ?





Optical Character Recognition (OCR) using

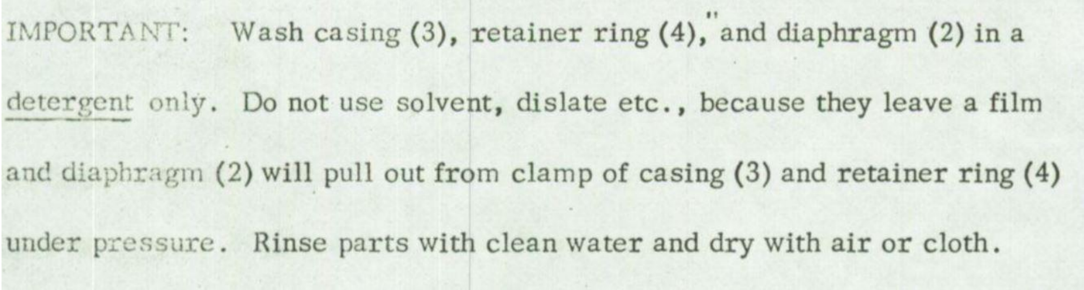
R packages:

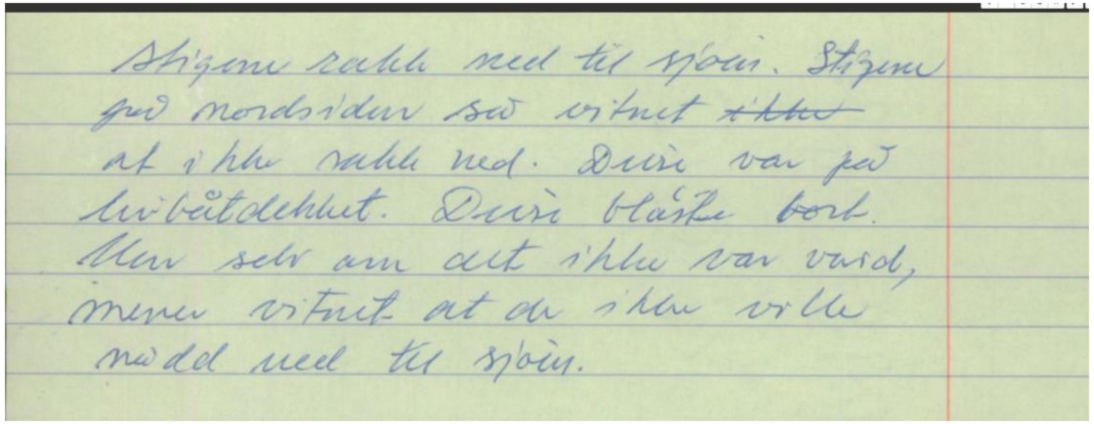
- magick
  - image pre-processing: converted to black/white and scaled to 2000 px.
- tesseract
  - OCR-scanning, using both the Norwegian and English libraries
  - Ignoring certain special characters (blacklist)
- future and future.apply
  - parallel processing

Result: a folder with the same files in .txt-format

Quality of result

Original


IMPORTANT: Wash casing (3), retainer ring (4), and diaphragm (2) in a detergent only. Do not use solvent, dislate etc., because they leave a film and diaphragm (2) will pull out from clamp of casing (3) and retainer ring (4) under pressure. Rinse parts with clean water and dry with air or cloth.


Stigene rakk ned til sjøen. Stigene på nordsiden så vitnet ~~ikke~~ at ikke rakk ned. Denne var på livbåtdekket. Denne blåste bort. Men selv om det ikke var [illegible], mener vitnet at de ikke ville nådd ned til sjøen.

After OCR

IMPORTANT: Wash casing (3), retainer ring (4), and diaphragm (2) in å detergent only. Do not use solvent, dislate etc., because they leave a film and diapbragm (2) will pull out from clamp of casing (3) and retainer ring (4) under pressure. Rinse parts with clean water and dry with air or cloth.

j d / f i s € c b , brt er , JE FN tt / , LA T PE Å F K da å , p L / E CA (U FR erd ø" 9 " på 7 t d , H) / J EØN » Li) » å , i 4 xå , (- d tt 81 IG åt ng uU , fa) I v "H) å å K G Z f Zee K UTG EX Ø Z t E / p p X / å K C KA REA CE ha k 8 é v € - o 3 TX I ÅD k LÅæt Y t 7 A t 3 f 3 - v d 7 , ke) å / 3 U N p € E . Ha «2 KE E Viten vg i b / L é j / VP Å å U (A c) y / » f 7 yt Z p , 7 Hit ar PA j / SI po z ZZ E É æ (/ rd Lak Å r " ø ET 1 HE LEAN EE I 7 A € (€ (p P- J , så (, le bi , / 19 2A «

Attaching metadata to the files through webscraping

- 1 file = 1 page
- No metadata (title, archive ref, topic etc.) attached
- Helper file could identify each page's archive folder reference, and each folders webpage
- Webscraping using the R packages rvest and RSelenium

Innhold

Visningsvalg: Utvidet Komprimert

Statsarkivet i Stavanger

Pa 1503 - Stavanger Drilling AS

Korrespondanse og saksdokumenter

Fides A/S	1977 - 1981			2
Andresens Bank International	1977 - 1980			205
Avdragsutsettelser	1980			321
Aksjeprotokoll	1978 - 1983			451
Labour Contract - Forhandlinger Forasol	1974			574
Kontraktsforhandlinger, Forex Neptune	1973 - 1974			612
Selskapsavtalen, addendum	1977 - 1982			727

Kildeinformasjon

Statsarkivet i Stavanger

Oppbevaringssted

Statsarkivet i Stavanger Statsarkivet i Stavanger

Arkivreferanse

SAST/A-101906/D/L0004

Lenke til Arkivportalen

Arkiv

Pa 1503 - Stavanger Drilling AS

Serie og underserie(r)

D: Korrespondanse og saksdokumenter

Stykke/mappe

L0004: Korrespondanse og saksdokumenter

Kildetype

Annen kilde

Protokollnr./tidsrom

nr. 4 /1973 - 1982

Område

-

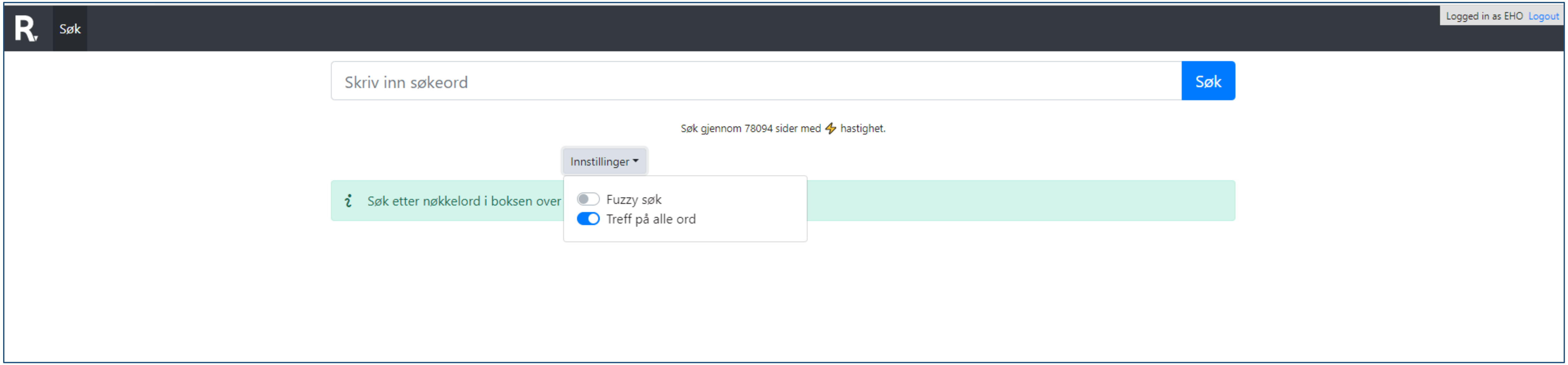
Merknader

Korrespondanse og saksdokumenter: Diverse

Emneknagger

Sakarkiv | Brev og korrespondanse | Industri | Olje- og petroleumsindustri | Privatarkiver | Bedriftsarkiver | Alexander L. Kielland-ulykken

Search app using



- Webapp created using the Shiny package in R (and our own UI package)
- Documents (texts) pushed to Elasticsearch index via R (package elastic)

kielland

Søk

Søk gjennom 78094 sider med ⚡ hastighet.

Innstillinger

Arkiv

Pa 1503 - Stavanger Drilling AS (9328)

Justisdepartementet, Granskningskommisjonen ved Alexander Kielland-ulykken 27.3.1980 (2503)

Dokumentserie

Alexander L. Kielland (3859)

Saksarkiv ordnet etter evt. andre (sideordnede) systemer (3808)

Alexander L.Kielland (3704)

Alexander L. Kielland - Sak og korrespondanse (3666)

Granskningskommisjonen ved Alexander Kielland-ulykken (2503)

Møtebøker, referatprotokoller, forhandlingsprotokoller o.l. (1307)

Styret (1187)

Styrekorrespondanse (774)

Korrespondanse og saksdokumenter (354)

Styredokumenter (263)

Etiketter

Industri (11831)

Olje- og petroleumsindustri (11831)

Bedriftsarkiver (9328)

Privatarkiver (9328)

Brev og korrespondanse (6814)

Sakarkiv (6078)

Departementene (2503)

Statlige arkiver (2503)

Rettergang (1943)

Møteprotokoller (1307)

Hoveddokument

Viser treff 1 til 10 av totalt 11831 treff på kielland. Søket tok 66 millisekunder

Sak og korrespondanse

1976 - 1984 (nr. 9 /1976 - 1984)

KIELLAND". Til orientering oversendes kopi av brev datert 13. april 1977 fra Sjøfartsdirektoratet, Oslo.

Kielland" No. S 195-1354 "Alexander L. Kielland", Albushell Field. No. S 195-071 'Alexander L. Kielland", Eldfisk Plat. 2/7F.T.P. 2/MA to Eldfisk Plat. 2/7B. No. S 195-1345 "Alexander L.

Kielland", anchor pattern Eldfisk Platform 2/7B. No. S 195-1347 "Alexander L. Kielland", anchor pattern vessel warped clear of platform Eldfisk Platform 2/7B. No. 8 195-1351 "Alexander L.

Kielland", anchor pattern Albushell Platform 2/4F. No. 8 195-1353 "Alexander L. Kielland", anchor pattern vessel warped clear of platform Albushell Platform 2/4F. Med hilsen, for A.

Sakarkiv | Brev og korrespondanse | Industri | Olje- og petroleumsindustri | Privatarkiver | Bedriftsarkiver

Gå til bilde | Gå til pdf | ★ Se lignende dokumenter

Styrekorrespondanse Stavanger Drilling II A/S

1982 - 1983 (nr. 9 /1981 - 1983)

ADVOKATENE VÅLAND å STAALESEN side 3 ' så vidt gjelder avgjørelsen om at krav fra Staten, Kielland- fondet og Henry Andreassen i forbindelse med Alexander L.

Kielland-ulykken er unntatt fra begrensning efter reglene i Søkes 234. 2.

-Subsidiært: Stavanger Drilling II A/S frifinnes for påstand om at Statens, Kielland-fondets og Henry Andreassens krav i forbindelse med Alexander L.

Kielland-ulykken skal være unntatt fra begrens ning efter reglen: Syøl, 15423545 . Den ankende part tilkjennes : tninger hos de ankend: Sue len 26. oktober 1983 v

Møteprotokoller | Brev og korrespondanse | Industri | Olje- og petroleumsindustri | Privatarkiver | Bedriftsarkiver

Gå til pdf | ★ Se lignende dokumenter

10 min break

Happy to answer questions
from this first session
after the break

Part IV

Classifying criminal cases by using machine learning on text

Used in a recent performance audit report on «The Police's efforts towards ICT-crime»

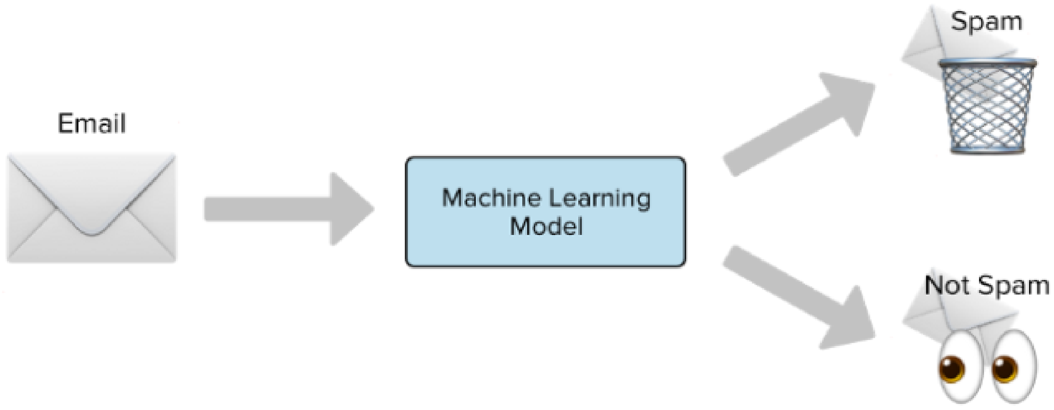
ML used on text– nothing new

For example used in spam filters for a long time

Spam Mail Detection Using Support Vector Machine.

Shreyak Follow
Aug 5, 2020 · 3 min read

In this blog, we are going to classify emails into Spam and Anti Spam. Here I have used SVM Machine Learning Model for that.



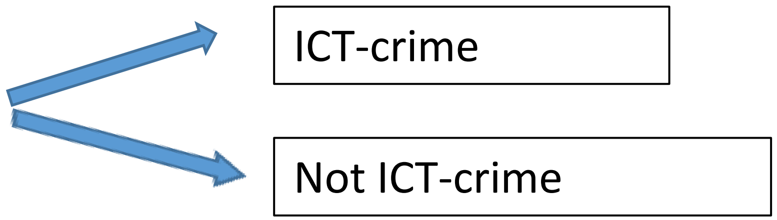
Source: https://becominghuman.ai/spam-mail-detection-using-support-vector-machine-cdb57b0d62a8

What we (well, not me) did

- Quite similar to the spam-filter example

- The question:
  - «Of all reported criminal cases – how many are related to ICT crime?»

  - A classic (binary) classification problem
    - 300 000+ cases → ML-algorithm



Alternative algorithms tested

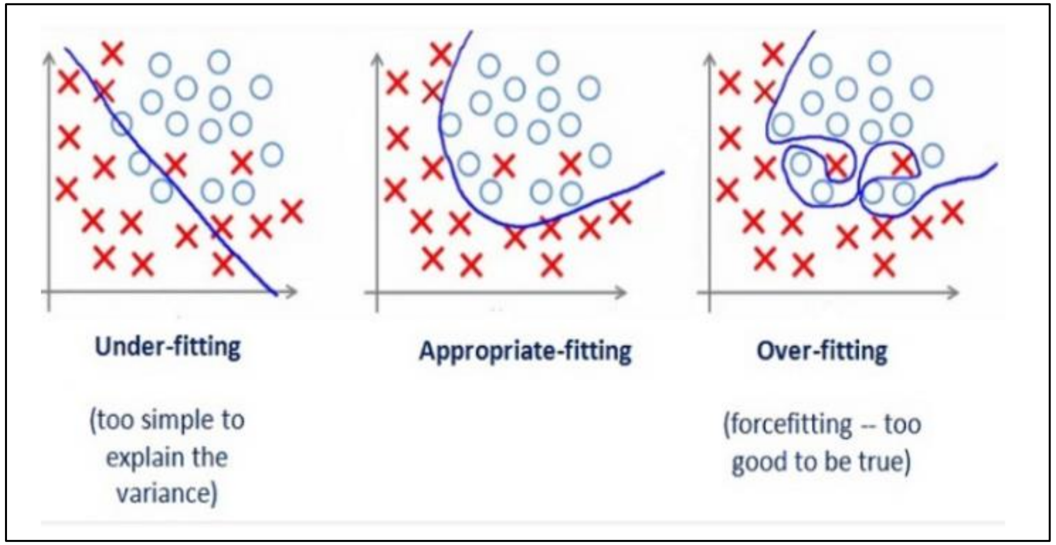
- Naive Bayes (bad result – just a bit better than a coin toss)
- Random Forest (heavily overtrained)
- XGBoost (heavily overtrained)
- Neural Network (bad result – not good enough data)
- **Support Vector Machine (SVM) – chosen**
  - SVM is a supervised, black box ML-model

- Test always several alternatives → seldom obvious what’s best for your case

A note om «overtraining»
(also known as «overfitting»)

- Overtraining is a chronic problem when doing supervised ML

- Simplified:
  - You make (train) a classification model on the basis of some known (training) data
  - To predict class affiliation for new units with unknown data
  - Always a risk for the model to be «too well» fitted to training data → result is meagre prediction power



Source: https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76

Train a model? Fine, but…

- Help, we don’t have training data!
- Well, then we must make training data…

The making of training data

Manual classification of 1072 cases

1. Got all documentation from 334 544 criminal cases

2. Drew a random sample of 1072 cases, for manual assessment and classification

Result: 1072 manually classified cases. This became our training data

ML-classification

- Data → text from crime report, case description etc.
- Standard data preparation: tokenization, stemming, removing stopwords etc.
- Some terms specific/more prevalent for ICT crime cases
- Terms are thus the variables (features) defining the prediction

Choice of terms/variables/features

- Some terms more prevalent for IKT crime, than for not-ICT crime
- Weighting terms using TF-IDF (based on training data)
  - (Term Frequency – Inverse Document Frequency)
- Used the 150 terms with highest weight from each class, removed common words (from the 1072 training cases)
- Chose 70 terms (variables) with the greatest difference in weight from the two classes

On synonyms – a thing…

- Some terms appear rarely, but are very ICT specific (for example «hack»)
- Important terms, but each has limited weight as they are rare
- Made an "index/synonym" variable which represented the collection of these terms

«Synonym» to remove standard text

- OCR scan of forms → you also get unwanted standard form text
- Made a «synonym»-variable which contained the standard form text (sentences)
- «IF form is used, THEN remove text as defined in synonym variable»

Synonyms – General experience

- Important to have a plan for handling synonyms
- Often important for the model's prediction power

Training-test-validation of the model

- Algorithm: Support Vector Machine
- 70+ variables
- 1072 manually classified cases as training data
- Used 5-fold cross validation (944 cases for training/test, 137 cases to validate the model)

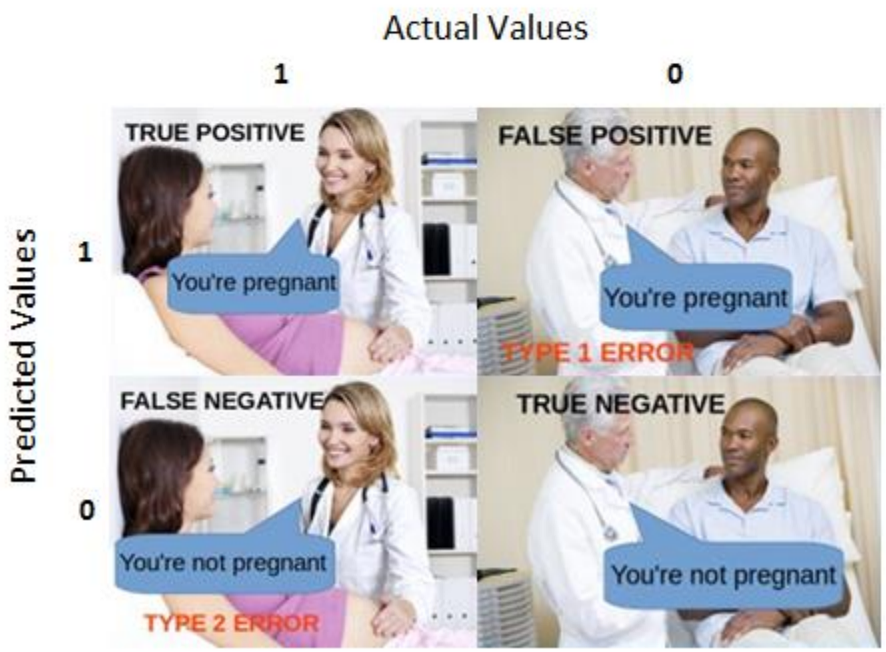
The final, trained model

- Run on the entire population (300 000+ cases)
- Cases most similar textually to ICT crime…
- …are put in the «ICT crime box» by the model

Results…?

- Different metrics on model fit

- No model is perfect (Ref. «model»)



https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

- F.ex. pregnancy – what is worse?
  - To get the message of pregnancy – when you in fact are not? (false positive)
  - To get the message of non-pregnancy – when you in face are pregnant? (false negative)

- So: Which metric is best in a certain case?

Matthews correlation coefficient (MCC)

- MCC = 1
  - Perfect fit between prediction and reality (all cases classified correctly)
- MCC = 0
  - The model is as good (bad) as a coin toss
- MCC = -1
  - 100 % of the cases are misclassified

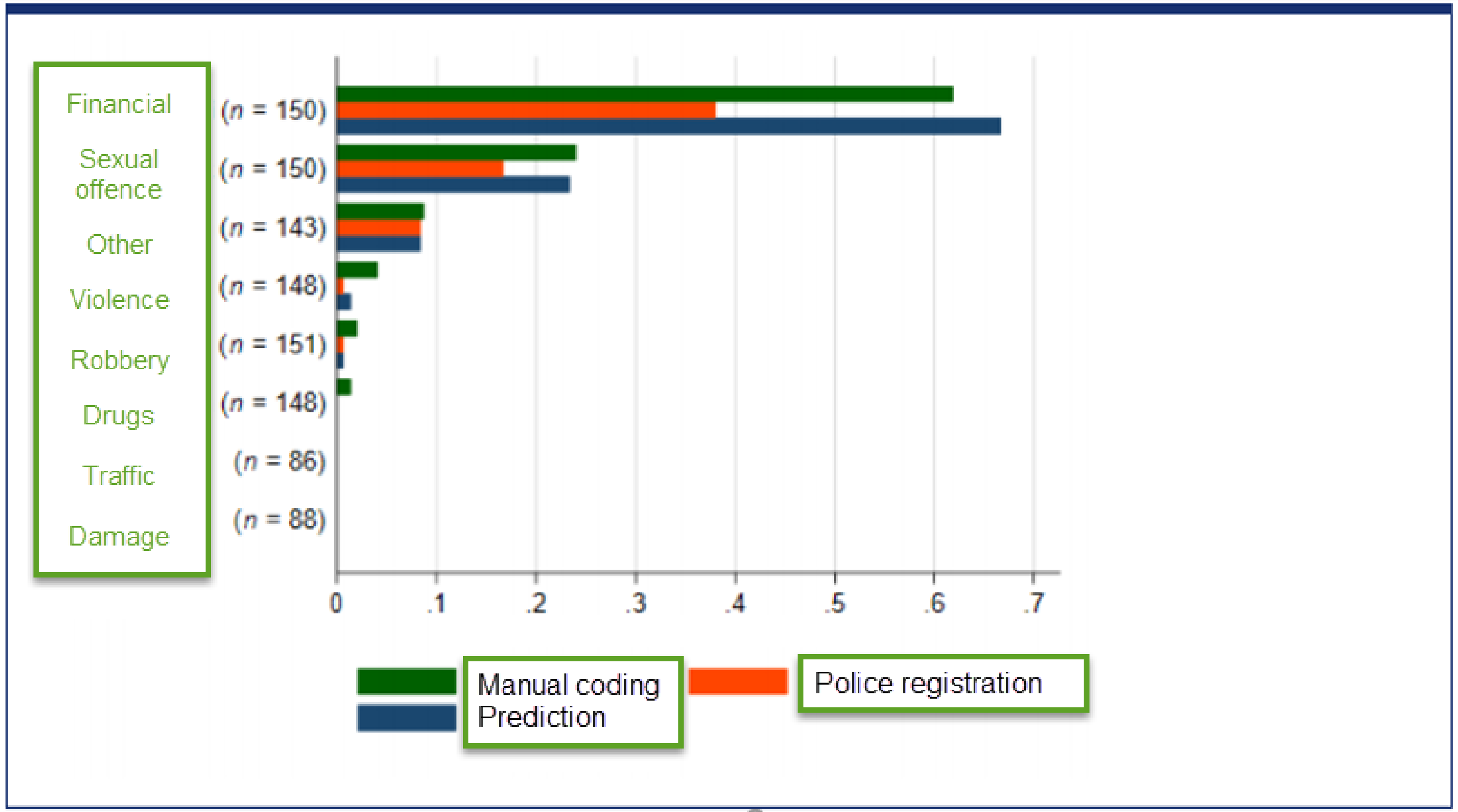
$$MCC = \frac{TN \times TP - FP \times FN}{\sqrt{(TN + FN)(FP + TP)(TN + FP)(FN + TP)}}$$

- Result - MCC for cases on ICT crime: 0,82

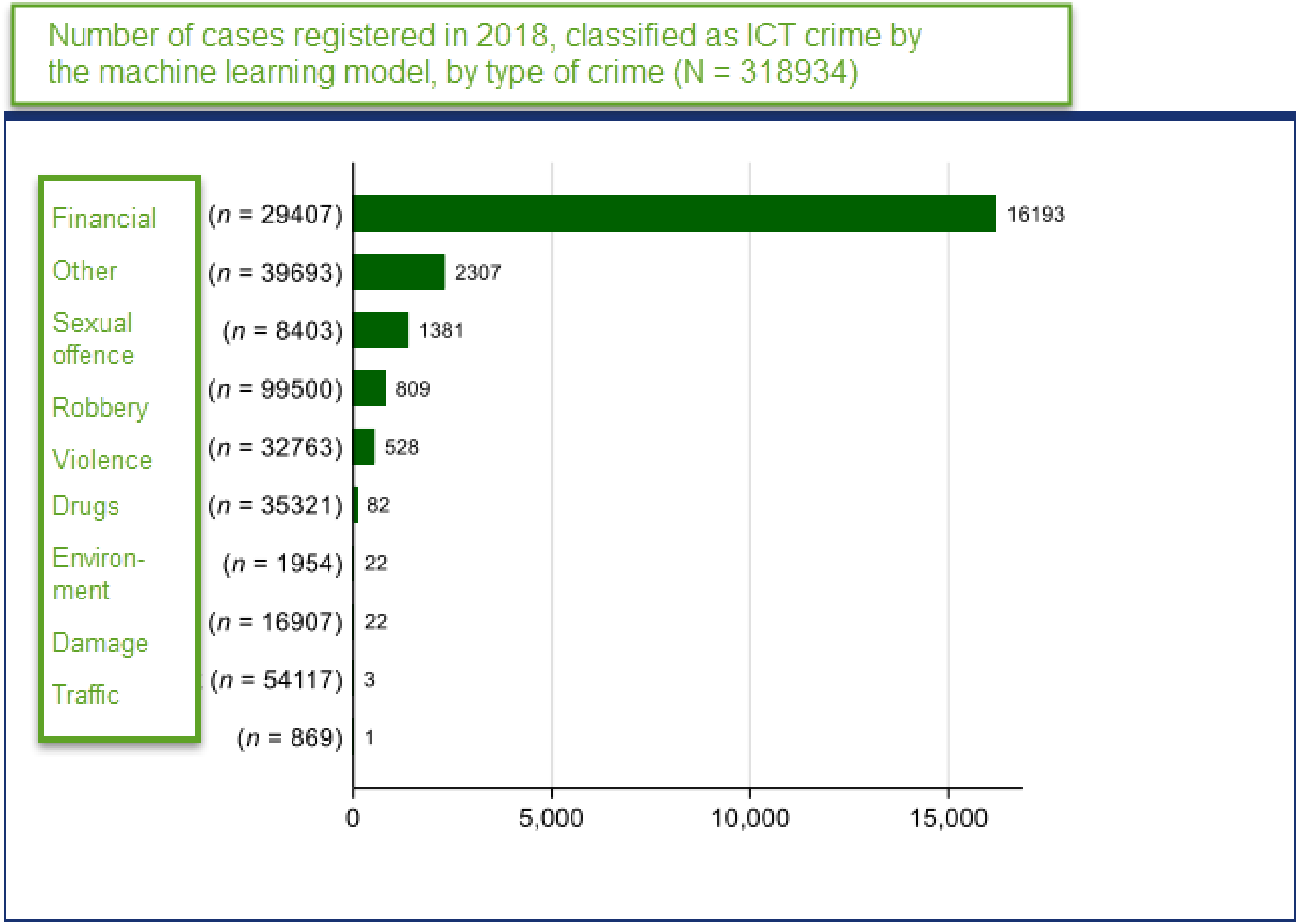
Results cont'd:

- 21 500 cases of a total of 334 544 was categorised by the model as ICT crime.

Proportion of ICT crime according to manual coding, police mode registration and the machine learning model prediction, by crime type (N = 1072)



And some more results…



A brief summary

- So – what now?
- Well, it’s a question of knowledge
- Most NLP techniques are rather easy to understand
- All you need is an old laptop, Python & a (somewhat) bright mind
- Just go for it!