Tracing conceptual change in messy data (2): self-reliance as boon and bane

© Joris van Eijnatten
My question

• which popular concepts of “Europe” do newspapers reveal?

OR

• has football always been more important than Brussels?

AND

• how can we trace concepts of “Europe” in messily digitized newspapers?
My question

Three approaches:

1. The mundane
   - e.g. crossword puzzles, weather forecasts
2. Visions of/for the future
   - e.g. political ideals, philosophical views
3. Popular competition
   - e.g. Miss Europe, Eurovision Song Contest, Football
What I need

• A handy toolbox
  – to trace conceptual change
  – in ± big data
  – of not so very good quality
  – over a longer period of time
  – in more than one language

• comparative analysis over time and space
Data

• “messy data”

De "Europa" gereed.

De "Europa", het susleraJüp van de .üremer", welks U watcrlulng door den brand van Juli J.l. werd vci trautfd4* Maan &gt;i... -- 10 Hamburg tn tut duk g-ü-aa en ui man* &gt;oof het laatst »orden nagelen. De eerste ntoefvaart tal midden ■ Februari worden Keinaakt,- terwijl de ecrstrtil naar New York t’J Maan tal bceinnui

susleraJüp = <zusterschip>
.üremer“ = <“Bremer”>
U watcrlulng = <te waterlating>
vci trautfd4* = <vertraagd is>

Rotterdamsch nieuwsblad, 08-01-1930
Toolbox, January 2016

- which accessible & robust tools do we actually have?

nGrams (e.g. Delpher)
> insightful but too simple & too rigid
Toolbox, January 2016

• which accessible & robust tools do we actually have?

  semantic text-mining (e.g. Texcavator)
  > temporal dimension, proven, but restricted to KB corpus
• which **accessible** & **robust** tools do we actually have?

corpus linguistics (e.g. Antconc)
  > proven, but no temporal dimension

<table>
<thead>
<tr>
<th>Rank</th>
<th>Freq</th>
<th>Freq(L)</th>
<th>Freq(R)</th>
<th>Stat</th>
<th>Collocate</th>
<th>Stopword</th>
</tr>
</thead>
<tbody>
<tr>
<td>5241</td>
<td>6634</td>
<td>6462</td>
<td>172</td>
<td>729.98</td>
<td>west</td>
<td>#N/A</td>
</tr>
<tr>
<td>5052</td>
<td>2747</td>
<td>32</td>
<td>2715</td>
<td>775.37</td>
<td>cup</td>
<td>#N/A</td>
</tr>
<tr>
<td>6878</td>
<td>1822</td>
<td>1737</td>
<td>85</td>
<td>688.49</td>
<td>oost</td>
<td>#N/A</td>
</tr>
<tr>
<td>10072</td>
<td>1034</td>
<td>444</td>
<td>590</td>
<td>565.70</td>
<td>amerika</td>
<td>#N/A</td>
</tr>
<tr>
<td>14795</td>
<td>853</td>
<td>534</td>
<td>319</td>
<td>437.29</td>
<td>amerikaanse</td>
<td>#N/A</td>
</tr>
<tr>
<td>14651</td>
<td>789</td>
<td>592</td>
<td>197</td>
<td>443.68</td>
<td>landen</td>
<td>#N/A</td>
</tr>
<tr>
<td>9261</td>
<td>747</td>
<td>627</td>
<td>120</td>
<td>586.85</td>
<td>midden</td>
<td>#N/A</td>
</tr>
<tr>
<td>15226</td>
<td>745</td>
<td>521</td>
<td>224</td>
<td>430.26</td>
<td>nieuwe</td>
<td>#N/A</td>
</tr>
<tr>
<td>17198</td>
<td>694</td>
<td>328</td>
<td>366</td>
<td>388.85</td>
<td>nederland</td>
<td>#N/A</td>
</tr>
<tr>
<td>9401</td>
<td>639</td>
<td>528</td>
<td>111</td>
<td>574.02</td>
<td>kernwapens</td>
<td>#N/A</td>
</tr>
<tr>
<td>10061</td>
<td>598</td>
<td>447</td>
<td>151</td>
<td>568.20</td>
<td>raketten</td>
<td>#N/A</td>
</tr>
<tr>
<td>12938</td>
<td>575</td>
<td>236</td>
<td>339</td>
<td>482.67</td>
<td>verenigde</td>
<td>#N/A</td>
</tr>
<tr>
<td>13540</td>
<td>566</td>
<td>278</td>
<td>288</td>
<td>468.14</td>
<td>staten</td>
<td>#N/A</td>
</tr>
<tr>
<td>22340</td>
<td>562</td>
<td>270</td>
<td>292</td>
<td>299.19</td>
<td>jaar</td>
<td>#N/A</td>
</tr>
<tr>
<td>11774</td>
<td>499</td>
<td>374</td>
<td>125</td>
<td>508.09</td>
<td>avro</td>
<td>#N/A</td>
</tr>
</tbody>
</table>
• which **accessible** & **robust** tools do we actually have?

topic modelling (e.g. Mallet)
> proven, but no temporal dimension

<table>
<thead>
<tr>
<th>Id</th>
<th>words</th>
<th>topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>europa, cup</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>jaar, moskou</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>europa, terug</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>nieuwe, nederland</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>polen, miljoen</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>vs, isra, iran</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ton, eerste</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>land, blijft</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>redactie, voetbal</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>verslaggever, hans</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>nederland, landen</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>amerika, westen</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>west, reagan</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>grote, vrede</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>navo, raketten</td>
<td></td>
</tr>
</tbody>
</table>

**code, first name, weather, geography, defence, media, economy, sports, politics**
Toolbox, January 2016

- which accessible & robust tools do we actually have?

  text analytics package (e.g. SPSS Modeler)
  > proven, but no temporal dimension & black box

Category web of <pan-Europa>, article titles, N = 42,712 docs

Dutch territorial newspapers 1930-31
• which accessible & robust tools do we actually have?

vector-space modelling (e.g. ShiCo)
> temporal dimension, unproven & black box
Case: analysing the mundane

assignment
• “determine the frequency of locations mentioned in weather forecasts and plot them on a dynamic, time-based graph”

hypothesis
• “weather forecasts offer insight into the geographical dimensions of a popular concept of Europe”
WEER IN EUROPA
Door het zich naar Zuid west-Ierland terugtrekken van een hoge-drukgebied stroomt minder koude lucht van de Oceaan West- en Midden-Europa binnen. Koud blijft het nog in het zuiden, met temperatuur in Spanje en aan de IJsvlêra van vijf tot tien graden. Ook in het Alpengebied blijft het nog winters met enige sneeuwval, maar later zal de zachtere lucht verder naar Midden-Europa doordringen.
Problem

no immediately available tool could help me to perform this simple assignment

so I tried a little *self-reliance*
Weather forecasts: methodology

• at some point in your research, write down your methodology
• it helps you to better understand what you have been doing

• weather forecasts required a 4-step methodology
Methodology: Step 1

1. Extract dataset of weather forecasts
   1. Extract per decade from the KB terrabyte all articles and advertisements published between 1880 and 1990 containing the word “temperatu*”
   2. Reduce the dataset to records that are very probably weather forecasts, using the query
   3. Further reduce the dataset to records that are exclusively weather forecasts
   4. Divide dataset into subsets based on newspaper titles (different newspapers have different OCRs)
Methodology: Step 2

2. Get a list of place names and further refine dataset

5. Differentiate between
   • towns/cities: Leuven, Sint Job in ’t Goor, Silly, Corny
   • countries: Ireland, Germany, Lichtenstein
   • regions within a country: Friesland, Bavaria
   • regions across countries: Lapland, the Alps

6. Generate a basic list of place names to further refine the dataset

7. Remove all records that mention only De Bilt ("d[\s\S]*?bilt")

8. Get list of names for each category (see 5):
   - e.g. github.com/David-Haim /CountriesToCitiesJSON
Methodology: Step 3

3. Determine frequencies of place names in datasets

9. Create a list of name variants using regular expressions
   • Zürich = z[üë]{1,2}rich|[üë]{1,2}rich|zÃ¼rch|z[lt]irich|zurieh
   • West Germany = west[\s\-]?deuts.{1,5}d|west[\-\s]?du[\s]?it[\s\s]{0,2}land

10. Determine frequencies of place names in data set

11. Use and augment a list of geographical stopwords (= false place names, n= 3,135)
   • “alexander”, “hjo”, “zomergem”

12. Use and augment list of mistaken identities
   • “middelburg; Belgium”, “los angeles; Spain”, “china; Russia”

13. Repeat 10 through 12
Methodology: Step 4

4. Normalize and plot on map
   14. Normalize frequencies of place names over the total number of records per dataset
   15. Obtain coordinates for place names (lat & long)
   16. Plot place names and frequencies on map
   17. Take a very long vacation
Methodology: which tools?

1. Texcavator
2. SPSS Modeler
3. Excel (manual)
4. SPSS Modeler
5. Excel
6. Python 3 script
7. Python 3 script
8. Browser
9. Regex editor
10. Python 3 script
11. Python 3 script
12. Python 3 script
13. Python 3 script
14. Python 3 script
15. Hamster maps
16. Carto DB / Google Fusion Tables
17. EasyJet
import pandas as pd
import numpy as np

# get input
name_file = pd.read_csv('OriginalNames_Regex.csv', sep=(';'), names=['Concept', 'Concept+', 'Category', 'Country', 'Regex', 'Latitude', 'Longitude'], encoding='ISO-8859-1')
name_pat = (name_file['Regex'])

prompt1 = input('Please insert the name of the TRUE data file you want to process 
')
prompt2 = input('Please insert the name of your RAW output frequency file 
')
weather_file = pd.read_csv(prompt1, sep=(';'), names=('_id', 'paper_dc_date', 'paper_dc_language', 'paper_dc_title', 'paper_dc_publisher', 'paper_dcterms_issued', 'paper_dcterms_temporal', 'paper_dcx_issuenumber', 'article_dc_subject', 'article_dc_title', 'text_content'), encoding='ISO-8859-1')

#generate frequencies of regex names
content = pd.Series(weather_file['text_content'])
name_dict = {}
for name in name_pat:
    name_dict[name] = 0
result_list = []
for item in name_dict:
    result = content.str.contains(item, case=False, regex=True).sum()
    if result == 0:
        pass
    elif result > 0:
        result_list.append([item, result])
<table>
<thead>
<tr>
<th>city/country</th>
<th>category</th>
<th>latitude</th>
<th>longitude</th>
<th>frequency 50-55 norm</th>
<th>50-59 norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>amsterdam; Netherlands</td>
<td>Town</td>
<td>52.370.215</td>
<td>4.895.167</td>
<td>788</td>
<td>690.018</td>
</tr>
<tr>
<td>geneva; Switzerland</td>
<td>Town</td>
<td>46.204.390</td>
<td>6.143.157</td>
<td>554</td>
<td>485.114</td>
</tr>
<tr>
<td>de bilt; Netherlands</td>
<td>Town</td>
<td>52.109.271</td>
<td>5.180.967</td>
<td>185</td>
<td>161.996</td>
</tr>
<tr>
<td>paris; France</td>
<td>Town</td>
<td>48.856.614</td>
<td>2.352.221</td>
<td>135</td>
<td>118.214</td>
</tr>
<tr>
<td>london; United Kingdom</td>
<td>Town</td>
<td>51.507.350</td>
<td>-0.127758</td>
<td>130</td>
<td>113.835</td>
</tr>
<tr>
<td>oslo; Norway</td>
<td>Town</td>
<td>59.913.868.800</td>
<td>10.752.245.400</td>
<td>117</td>
<td>102.452</td>
</tr>
<tr>
<td>groningen; Netherlands</td>
<td>Town</td>
<td>53.219.383</td>
<td>6.566.501</td>
<td>113</td>
<td>98.949</td>
</tr>
<tr>
<td>berlin; Germany</td>
<td>Town</td>
<td>52.520.006</td>
<td>13.404.953</td>
<td>105</td>
<td>91.944</td>
</tr>
<tr>
<td>stockholm; Sweden</td>
<td>Town</td>
<td>59.329.323.500</td>
<td>18.068.580.800</td>
<td>93</td>
<td>81.436</td>
</tr>
<tr>
<td>helsinki; Finland</td>
<td>Town</td>
<td>60.169.855.700</td>
<td>24.938.379.000</td>
<td>91</td>
<td>79.685</td>
</tr>
<tr>
<td>luxembourg; Luxembourg</td>
<td>Town</td>
<td>49.611.621</td>
<td>6.131.934</td>
<td>90</td>
<td>78.809</td>
</tr>
<tr>
<td>innsbruck; Austria</td>
<td>Town</td>
<td>47.269.212.400</td>
<td>11.404.102.400</td>
<td>75</td>
<td>65.674</td>
</tr>
<tr>
<td>madrid; Spain</td>
<td>Town</td>
<td>40.416.775.400</td>
<td>-3.703.790.200</td>
<td>75</td>
<td>65.674</td>
</tr>
<tr>
<td>nice; France</td>
<td>Town</td>
<td>43.710.172.800</td>
<td>7.261.953.200</td>
<td>75</td>
<td>65.674</td>
</tr>
<tr>
<td>zurich; Switzerland</td>
<td>Town</td>
<td>47.376.886</td>
<td>8.541.694</td>
<td>60</td>
<td>52.539</td>
</tr>
<tr>
<td>copenhagen; Denmark</td>
<td>Town</td>
<td>55.676.096.800</td>
<td>12.568.337.100</td>
<td>59</td>
<td>51.664</td>
</tr>
<tr>
<td>den helder; Netherlands</td>
<td>Town</td>
<td>52.956.280</td>
<td>4.760.797</td>
<td>59</td>
<td>51.664</td>
</tr>
<tr>
<td>locarno; Switzerland</td>
<td>Town</td>
<td>46.166.998</td>
<td>8.794.264</td>
<td>57</td>
<td>49.912</td>
</tr>
<tr>
<td>algiers; Algeria</td>
<td>Town</td>
<td>36.753.770</td>
<td>3.058.792</td>
<td>57</td>
<td>49.912</td>
</tr>
<tr>
<td>leeuwarden; Netherlands</td>
<td>Town</td>
<td>53.201.233.400</td>
<td>5.799.913.300</td>
<td>56</td>
<td>49.037</td>
</tr>
<tr>
<td>grenoble; France</td>
<td>Town</td>
<td>45.188.529</td>
<td>5.724.523</td>
<td>53</td>
<td>46.41</td>
</tr>
<tr>
<td>bordeaux; France</td>
<td>Town</td>
<td>44.837.789</td>
<td>-0.579179</td>
<td>48</td>
<td>42.032</td>
</tr>
<tr>
<td>eindhoven; Netherlands</td>
<td>Town</td>
<td>51.441.641</td>
<td>5.469.722</td>
<td>43</td>
<td>37.653</td>
</tr>
<tr>
<td>brussels; Belgium</td>
<td>Town</td>
<td>50.850.339</td>
<td>4.351.710</td>
<td>42</td>
<td>36.778</td>
</tr>
</tbody>
</table>
Normalised frequency of towns in weather forecasts

*De Telegraaf*

1950-1959

n = 1,142
Lessons learnt, September 2016

self-reliance: boon or bane?

PROS or CONS
- programming is pleasant (great fun!)
- method is mandatory (write up your stuff)
- numbers are needed (you can’t avoid statistics)
- autonomy is arduous (time, time, time)
- halting is hard (where or when to draw the line?)
Lessons learnt, September 2016

there isn’t really very much out there (yet)
• the available tools are great but they are only a first step
remain in control
• insight into tools is crucial but involves blood, sweat, tears, pain, suffering, anguish, despair, depression... and time
help is expensive
• if you are poor, be self-reliant
get results
• you won’t be taken seriously otherwise
small is better
• work with specific tools in a circumscribed area
innovation often sucks
• funding institutions need to think about their language