08 – Building Language Resources from the Web
IA161 Advanced Techniques of Natural Language Processing

Vít Suchomel

NLP Centre, FI MU, Brno

November 26, 2020
Outline

1. Introduction: Web as a Language Resource
2. Efficient Web Crawling
3. Language Identification
4. Boilerplate Removal
5. Non-text removal
6. De-duplication
7. Plagiarism Detection
8. Task: Plagiarism Detection
Text Corpus

A corpus is a set of texts in a natural language.
Text Corpus

A corpus is a set of texts in a natural language.

Statistical NLP:
   a large amount of language use data
   situated within its textual context
Corpus Use

- generally: data for studying natural language
- linguists: analyses of language phenomena, language changes over time
- lexicographers, teachers: dictionaries, word meanings, examples of a typical use
- sociologists: style and theme, hot topics
- marketing experts: brands/product evaluation, sentiment analysis
- statistical NLP: language models for taggers, analysers, translation systems, predictive writing,...
Text Sources

- printed media: books, newspapers, magazines, poetry collections
- internet: articles, presentations, blogs, discussions, socnet messages (tweets, fb)
- speech: transcription of speech recordings, movie subtitles
- other: personal correspondence, school essays
Corpus Size Matters . . .

Most language phenomena follow the Zipfian distribution.
⇒ The more data the better.
Corpus Size Matters . . .

Most language phenomena follow the Zipfian distribution.
⇒ The more data the better.

Example: Modifiers of phrase ‘deliver speech’ (frequency):
- BNC (96 M words): major (8), keynote (6).
- ukWaC (1,32 G words): keynote (125), opening (12), budget (8), wedding (7).
- enTenTen12 (11,2 G words): keynote (813), acceptance (129), major (127), wedding (118), short (101), opening (97), famous (80).
- enTenTen15 (15,7 G words): keynote (3673), opening (684), welcome (413), key (257), major (255), acceptance (233), powerful (229), commencement (226), inspiring (210), inaugural (146).
- enClueWeb09 (70,5 G words): keynote (3802), acceptance (1035), opening (589), famous (555), commencement (356), impassioned (335), inaugural (333).
...But the Size Is Not Everything

A significant fraction of all web pages are of poor utility. ¹

¹ [Manning et al., 2008, Chapter 20]
...But the Size Is Not Everything

A significant fraction of all web pages are of poor utility. ¹

Why are qualitative aspects so important?

- web is the most used data source to obtain enough source texts – ‘Web as Corpus’
- web is garbage (by definition) – ‘garbage as corpus’?
- building language resources from the web requires extensive post-processing

¹ [Manning et al., 2008, Chapter 20]
Selected Issues of Building Web Corpora

- **language identification**
- character encoding detection
- **efficient web crawling**
- **boilerplate removal**
- **de-duplication** (removal of identical or nearly identical texts)
- **fighting web spam**
- text classification (topic, genre, language variety)
- authorship recognition & **plagiarism detection**
- storing & indexing of large text collections
Brno Processing Pipeline

1. web crawler SpiderLing – Suchomel, Pomikálek (2012)
2. character encoding detection (byte trigram model) – Pomikálek, Suchomel (2012)
3. language filtering (character trigram model)
4. boilerplate removal – Pomikálek (2011)
5. text tokenisation – Michelfeit, Suchomel (2014)
6. near duplicate paragraphs removal – Pomikálek (2011)
7. discerning (similar) languages – Suchomel (2019)
8. all data is stored and indexed by corpus manager Sketch Engine – Kilgarriff, Rychlí, Smrž, Tugwell (2004)

NLPC & Lexical Computing corpus tools: http://corpus.tools/
Outline

1. Introduction: Web as a Language Resource
2. Efficient Web Crawling
3. Language Identification
4. Boilerplate Removal
5. Non-text removal
6. De-duplication
7. Plagiarism Detection
8. Task: Plagiarism Detection
Web crawler

- Traverses the internet (graph of pages and links).
- Downloads documents (content & meta information).
- Stores documents (or their parts) in various formats for further use.
- Crawlers for various purposes:
  - GoogleBot – web indexing,
  - Linkcrawler – links, broken links checking,
  - Heritrix – general crawler, (Java, multiple treads),
  - SpiderLing – text corpora, (Python, multiple sockets).
Basic crawler design

Source: http://en.wikipedia.org/wiki/Web_crawler
Advanced crawler implementation details

- Distributed vs. extensible.
- Multi-threaded vs. multi-socketed.
- Web traversal policy:
  - depth vs. breadth,
  - domain selection,
  - domain distance,
  - focused crawling (topic oriented) vs. general crawling,
  - yield ratio.
Focused crawler design

General unfocused crawling efficiency (Heritrix)

![Graph showing the relationship between downloaded data size and final data size with different yield rates.](image)

- yield rate = 0.1
- yield rate = 0.01
- yield rate = 0.001
Domain yield ratio optimised efficiency (SpiderLing)

![Graph showing the relationship between downloaded data size and final data size for different yield rates. The graph includes lines for yield rates of 0.1, 0.01, and 0.001, with data points scattered across the graph.](image-url)
Outline

1 Introduction: Web as a Language Resource
2 Efficient Web Crawling
3 Language Identification
4 Boilerplate Removal
5 Non-text removal
6 De-duplication
7 Plagiarism Detection
8 Task: Plagiarism Detection
Issues of Language Identification of Text from the Web

- multiple languages in a single web page, e.g. Maori/English
- similar languages, e.g. Danish vs. Norwegian
- language varieties, e.g. European vs. Brasilian Portuguese
Solution

- Google Compact Language Detector v. 3
  - neural network model
- langid.py
  - naive Bayes classifier over byte n-grams ($1 \leq n \leq 4$)
Outline

1. Introduction: Web as a Language Resource
2. Efficient Web Crawling
3. Language Identification
4. Boilerplate Removal
5. Non-text removal
6. De-duplication
7. Plagiarism Detection
8. Task: Plagiarism Detection
What is boilerplate

- Repeated parts of a web page (not containing a new text) – header, footer, navigation.
- Uninteresting text (too short or not continuous) – advertisement, lists of items, article previews.
- Hard to recognise: discussions.
What is boilerplate

Source: http://corpus.tools/attachment/wiki/Justext/Algorithm/cs_classification_example.png
Boilerplate removal approaches

- **Machine learning (SVM, CRF, neural networks, n-gram models):**
  - Annotated web pages required for training.
  - Victor (CRF),
  - Ncleaner (n-grams).

- **Heuristics:**
  - Rules for including/excluding sections of text.
  - BTE (tag density),
  - Boilerpipe (link/text ratio),
  - jusText (link/text ratio, frequent words, context sensitive – smoothing).
Site Style Tree [Yi et al., 2003]

- Represents both layout and content of a web page.
- Node importance = node entropy over the whole Site Style Tree.

Context sensitive paragraph classification:

Outline

1. Introduction: Web as a Language Resource
2. Efficient Web Crawling
3. Language Identification
4. Boilerplate Removal
5. Non-text removal
6. De-duplication
7. Plagiarism Detection
8. Task: Plagiarism Detection
What Is Wrong with this Text?

Now on the web stores are very aggressive price smart so there genuinely isn’t any very good cause to go way out of your way to get the presents (unless of course of program you procrastinated).
Now on the web stores are very aggressive price smart so there genuinely isn’t any very good cause to go way out of your way to get the presents (unless of course of program you procrastinated).

Web spam, computer generated text – Not a good evidence of natural language phenomena
Web Spam Definition – Text Corpus Point of View

Good content: fluent, natural, consistent text (regardless its purpose)
Bad content – computer generated text
  • machine translation
  • keyword stuffing
  • phrase stitching
  • synonym replacement
  • automated summaries
  • any incoherent text

Varieties of spam removable by existing tools dealt with by other means
  • duplicate content
  • link farms
  • redirection
Approaches to Web Spam Removal

1. trustworthy websites only
2. website rules in the crawler: distance from the seeds, hostname
3. supervised classification
4. semi-manual filtering of websites

Suchomel: Better Web Corpora For Corpus Linguistics And NLP, doctoral thesis, Masaryk university, Brno, 2020

[Suchomel, 2020]
Trustworthy Websites Only

- works well but not perfect
- limited amount/size of trustworthy sources $\Rightarrow$ unsuitable for small languages
Website Distance from the Seed (Trustworthy) Websites

Text Quality by Domain Distance from Seed Domains in Estonian Web

- Web pages in the distance
- Bad (non-text or other reasons)
- Good

<table>
<thead>
<tr>
<th>Domain distance from seed domains</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web pages in the distance</td>
<td>503,831</td>
<td>3,852,489</td>
<td>954,303</td>
<td>296,598</td>
<td>44,593</td>
</tr>
<tr>
<td>Bad (non-text or other reasons)</td>
<td>1%</td>
<td>14%</td>
<td>10%</td>
<td>20%</td>
<td>29%</td>
</tr>
<tr>
<td>Good</td>
<td>99%</td>
<td>86%</td>
<td>90%</td>
<td>80%</td>
<td>71%</td>
</tr>
</tbody>
</table>
Supervised Classification – Data & Method

- 146 spam pages of 1630 manually classified web pages
- various web sources, 2006 to 2015
  - phrase and sentence level incoherency
  - frequent spam topics: medication, financial services, essay writing
  - other non-text, various techniques

- FastText supervised classifier (Mikolov, 2016)
- applied to a large English web corpus from 2015
- 35% most ‘spam-like’ documents removed
- recall: 70.5%
- precision: 71.5%
## Supervised Classification – Evaluation – Wordlist

<table>
<thead>
<tr>
<th>Document count</th>
<th>Original corpus</th>
<th>Clean corpus</th>
<th>Kept</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token count</td>
<td>58,438,034</td>
<td>37,810,139</td>
<td>64.7%</td>
</tr>
<tr>
<td></td>
<td>33,144,241,513</td>
<td>18,371,812,861</td>
<td>55.4%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phrase</th>
<th>Original hits/M</th>
<th>Clean hits/M</th>
<th>Kept</th>
</tr>
</thead>
<tbody>
<tr>
<td>viagra</td>
<td>229.71</td>
<td>3.42</td>
<td>0.8%</td>
</tr>
<tr>
<td>cialis 20 mg</td>
<td>2.74</td>
<td>0.02</td>
<td>0.4%</td>
</tr>
<tr>
<td>aspirin</td>
<td>5.63</td>
<td>1.52</td>
<td>14.8%</td>
</tr>
<tr>
<td>oral administration</td>
<td>0.26</td>
<td>0.23</td>
<td>48.8%</td>
</tr>
<tr>
<td>loan</td>
<td>166.32</td>
<td>48.34</td>
<td>16.1%</td>
</tr>
<tr>
<td>payday loan</td>
<td>24.19</td>
<td>1.09</td>
<td>2.5%</td>
</tr>
<tr>
<td>cheap</td>
<td>295.31</td>
<td>64.30</td>
<td>12.1%</td>
</tr>
<tr>
<td>interest rate</td>
<td>14.73</td>
<td>9.80</td>
<td>36.7%</td>
</tr>
<tr>
<td>essay</td>
<td>348.89</td>
<td>33.95</td>
<td>5.4%</td>
</tr>
<tr>
<td>essay writing</td>
<td>7.72</td>
<td>0.32</td>
<td>2.3%</td>
</tr>
<tr>
<td>pass the exam</td>
<td>0.34</td>
<td>0.36</td>
<td>59.4%</td>
</tr>
<tr>
<td>slot machine</td>
<td>3.50</td>
<td>0.99</td>
<td>15.8%</td>
</tr>
<tr>
<td>playing cards</td>
<td>1.01</td>
<td>0.67</td>
<td>36.8%</td>
</tr>
<tr>
<td>play games</td>
<td>3.55</td>
<td>3.68</td>
<td>53.9%</td>
</tr>
</tbody>
</table>
Supervised Classification – Evaluation – Collocates/Lexicography

Top collocate objects of verb ‘buy’ before and after spam removal

<table>
<thead>
<tr>
<th>Original corpus</th>
<th>Cleaned corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td>lemma</td>
<td>frequency</td>
</tr>
<tr>
<td>viagra</td>
<td>569,944</td>
</tr>
<tr>
<td>ciali</td>
<td>242,476</td>
</tr>
<tr>
<td>essay</td>
<td>212,077</td>
</tr>
<tr>
<td>paper</td>
<td>180,180</td>
</tr>
<tr>
<td>levitra</td>
<td>98,830</td>
</tr>
<tr>
<td>uk</td>
<td>93,491</td>
</tr>
<tr>
<td>ticket</td>
<td>85,994</td>
</tr>
<tr>
<td>product</td>
<td>105,263</td>
</tr>
<tr>
<td>cialis</td>
<td>71,359</td>
</tr>
<tr>
<td>car</td>
<td>75,496</td>
</tr>
<tr>
<td>house</td>
<td>70,204</td>
</tr>
<tr>
<td>propecia</td>
<td>55,883</td>
</tr>
</tbody>
</table>
Semi-manual Website Filtering

Data:
- 1,000 Estonian 2019 web sites, manually checked by Kristina Koppel (Tartu University)
- 16% marked as computer generated non-text, mostly machine translated, 6% marked as poor quality

Method:
- FastText supervised classifier
- probability threshold set to aim for a high recall

Evaluation:
- 100 positive & 100 negative random pages for manual evaluation
- recall: 97.1%, precision: 66.7%
- quite efficient method – just several man-days of manual work
Outline

1. Introduction: Web as a Language Resource
2. Efficient Web Crawling
3. Language Identification
4. Boilerplate Removal
5. Non-text removal
6. De-duplication
7. Plagiarism Detection
8. Task: Plagiarism Detection
De-duplication

- Quite straightforward for full duplicates.
- What about similar documents?
- People copy just parts of the document: original vs. copy
- Or copy and modify: original vs. modified
- Or copy and extend: original vs. extended
N-gram shingling algorithm
[Manning et al., 2008, Chapter 19]

- ‘Shingles’ of length of n words.
- N-grams represented by hashes.
Algorithm inspired by Broder’s shingling algorithm:

- Make n-grams of words for every structure,
- every n-gram is represented by its hash,
- the current structure is a duplicate $\iff$ at least $p\%$ of n-gram hashes is duplicate (has been observed before).
- Default options: structure = paragraph, $n = 7$, $p = 50$, smoothing.

---

Outline

1 Introduction: Web as a Language Resource
2 Efficient Web Crawling
3 Language Identification
4 Boilerplate Removal
5 Non-text removal
6 De-duplication
7 Plagiarism Detection
8 Task: Plagiarism Detection
Main and related tasks in plagiarism detection

- **Plagiarism detection**: Given a document, identify all plagiarized sources and boundaries of re-used passages.
- **Author identification**: Given a document, identify its author.
- **Author profiling**: Given a document, extract information about the author (e.g. gender, age).

External vs. Intrinsic plagiarism detection
[Potthast et al., 2010]

**External plagiarism detection**
Given a set of suspicious documents and a set of source documents the task is to find all text passages in the suspicious documents which have been plagiarized and the corresponding text passages in the source documents.

**Intrinsic plagiarism detection**
Given a set of suspicious documents the task is to identify all plagiarized text passages, e.g., by detecting writing style breaches. The comparison of a suspicious document with other documents is not allowed in this task.
Plagiarism techniques [Potthast et al., 2015]

- Random text operations. Random shuffling, insertion, replacement, or removal of characters, phrases or sentences. Replacement of characters with look-alike UTF characters.
- Semantic word variation. Random replacement words with synonyms, antonyms, hyponyms, or hypernyms.
- Part-of-speech-preserving word shuffling. Shuffling of phrases while maintaining the original POS sequence.
- Improvement of previous synthetic techniques: Insertions, replacements and variations may be obtained from context documents.
- Machine translation, cyclic translation. Automatic translation of a text passage from one language via a sequence of other languages to the original language.
- Summarization. Summaries of long text passages.
- Improvement of machine translation and summarization techniques: Manually corrected output.
Basic techniques for revealing similar documents

Bag of words

Full fingerprint methods
Overlapping substrings of length k in words from the beginning of the document.

Selective Fingerprint methods
Non-overlapping substrings of length k in words from the beginning of the document.

Rarest-in-document
All substrings are sorted according to their document frequency, then the rarest are selected as representatives of the document.

Selected Anchors
The document is reduced to pre-selected short chunks of characters.

Symmetric Similarity measure
\[ SS(X, Y) = \frac{|d(X) \cap d(Y)|}{|d(X) \cup d(Y)|} \]

where \( d(X) \) is a set of fingerprints of \( X \).

Outline

1. Introduction: Web as a Language Resource
2. Efficient Web Crawling
3. Language Identification
4. Boilerplate Removal
5. Non-text removal
6. De-duplication
7. Plagiarism Detection
8. Task: Plagiarism Detection
Task: Plagiarists vs. plagiarism detectors

Either:
Create 5 documents (with a similar topic) and 5 plagiarisms of these
documents, 10 documents total.\(^4\)

- 100 words \(\leq\) document length \(\leq\) 500 words
- 20% \(\leq\) plagiarism content \(\leq\) 90%
- POS tagged text:
  - Czech: asteria04:/opt/majka_pipe/majka-czech_v2.sh | cut -f1-3.
  - English: asteria04:/opt/treetagger_pipe/tt-english_v2.1.sh.
- For each plagiarism:
  1. describe plagiarism technique(s) used
  2. which detection methods might be able to reveal it – give reasons
  3. which detection methods might not be able to reveal it – give reasons

The minimal homework.

\(^4\) For the sake of simplicity: A plagiarism cannot have more sources here.
Task: Plagiarists vs. plagiarism detectors

Or:
Select a detection algorithm and implement it in Python.

- Input format: A POS tagged vertical consisting of structures doc with attributes author, id, class, source. Pair author, id is unique. Class is "original" or "plagiarism". Source is the id of the source (in case of plagiarism) or own id (in case of original).\(^5\)

- Output format: One plagiarism per line: id TAB detected source id TAB real source id. Evaluation line: precision, recall F1 measure.

- ./plagiarism_simple.py < training_data.vert

- Your script will be evaluated using data made by others.

- Describe which plagiarism detection technique(s) were implemented.

The right homework if you want to learn something.

---

\(^5\) For the sake of simplicity: A plagiarism cannot have more sources here.
Task: Input data example

<doc author="Já První" id="1" class="original" source="1">
<s>
Dnes dnes k6eAd1
je být k5eAaImIp3nS
pěkný pěkný k2eAgInSc4d1 pěkný
den den k1gInSc4 den
</s>
</doc>

<doc author="Já První" id="2" class="plagiarism" source="1">
<s>
Dnes dnes k6eAd1
je být k5eAaImIp3nS
ale ale k9
pěkný pěkný k2eAgInSc4d1 pěkný
den den k1gInSc4 den
</s>
</doc>
Task: Output example

\[
\begin{array}{c}
2 & 1 & 1 \\
1.00 & 1.00 & 1.00
\end{array}
\]


References II

Better Web Corpora for Corpus Linguistics and NLP. 

Eliminating noisy information in web pages for data mining. 
In Proceedings of the ninth ACM SIGKDD international conference on 
Knowledge discovery and data mining, pages 296–305. ACM.