

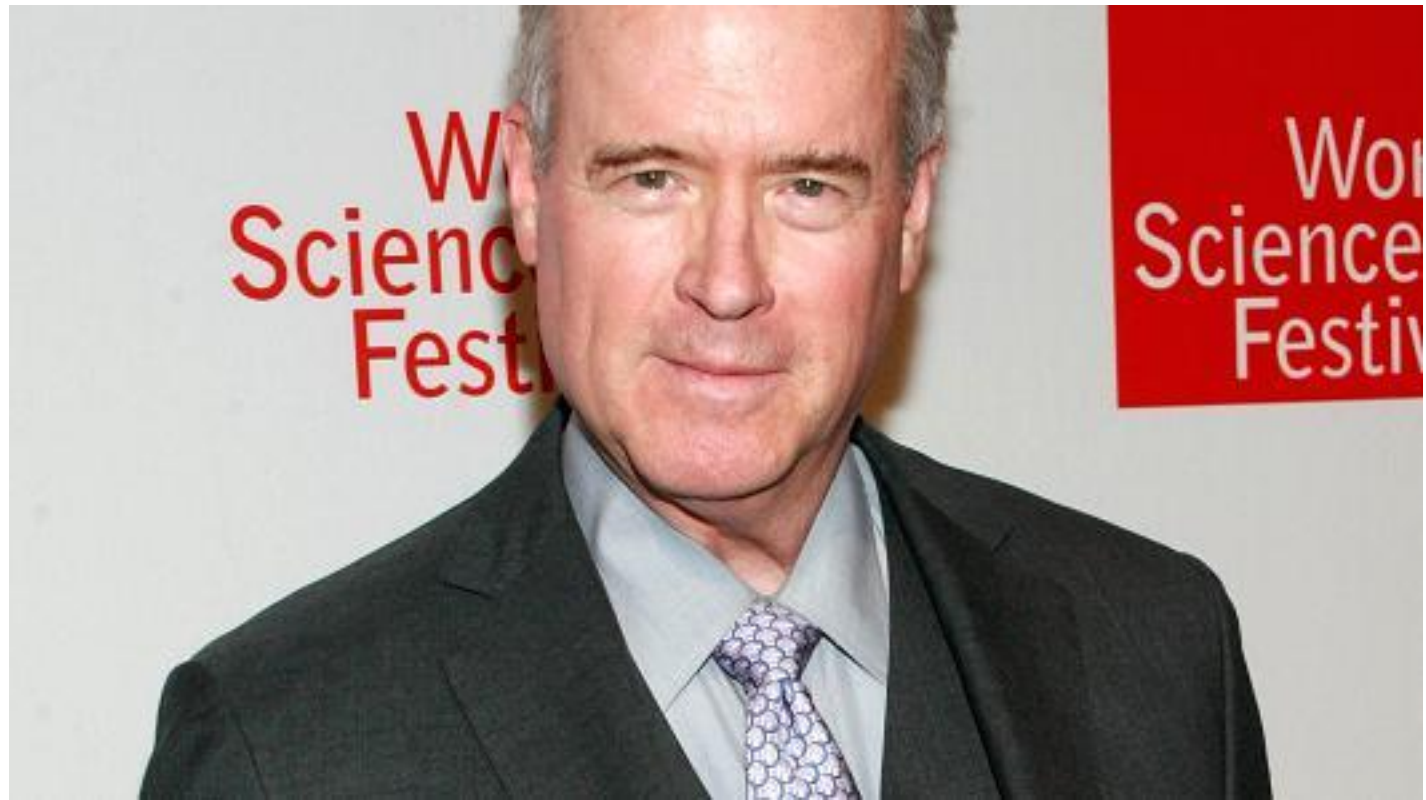
Corpus Acquisition from the Interwebs

Christian Buck,
University of Edinburgh

Motivation

“There is no data like more data”

(Bob Mercer, 1985)



Athènes, où l'industrie du barbier était fort en honneur, les plaideurs de la trempe de Figaro étaient rares, et si Solon avait négligé ce détail, la pratique avait bien été forcée de s'en occuper.

D'ailleurs il y avait des cas où l'observation de la loi eût été impossible en fait ou en droit. S'il s'agissait d'un enfant au berceau, d'un absent ou d'une femme, il était nécessaire que quelqu'un prît en main la cause de ce plaideur, qui ne pouvait ou ne devait pas présenter lui-même sa défense. Si l'État était directement intéressé à une affaire, si par exemple il s'agissait d'un procès de haute trahison, il ne fallait pas qu'en un cas aussi grave la défense de l'intérêt public fût abandonnée à l'initiative privée. La république désignait donc des orateurs, chargés de la représenter, et de soutenir l'accusation. Dans tous ces cas, il avait bien fallu déroger à la règle, et admettre que le plaideur se pouvait faire représenter par autrui.

De même, s'il s'agissait d'un laboureur, d'un vigneron, d'un matelot ou d'un soldat, dont la langue indocile se montrait rebelle à la parole, il devenait bien difficile d'appliquer la loi. C'eût été vraiment perdre le temps des juges, et aussi se moquer d'eux que de leur produire un plaideur absolument incapable d'exposer le premier mot de son affaire. N'était-ce pas d'ailleurs une injustice criante que de pauvres vieux soldats blanchis sous le harnais, qui dans maints combats s'étaient couverts d'une glorieuse sueur, fussent exposés sans défense aux attaques d'un vil sycophante, et se vissent ravir par une condamnation l'argent qui devait servir à leur assurer des funérailles décentes? Il avait paru juste que ce plaideur inexpérimenté pût appeler à son aide quelque ami bienveillant et disert, dont la langue officieuse voulût bien porter secours à sa détresse. Et comme les amis habiles et désintéressés étaient aussi rares à Athènes qu'ailleurs, il avait bien fallu s'adresser à quelque rhéteur ou sophiste de profession, dont on reconnaissait en secret les services par le paiement d'un honoraire.

Finding Monolingual Text

Simple Idea:

1. Download many websites
2. Extract text from HTML
3. Guess language of text
4. Add to corpus
5. Profit

Turns out all these are quite involved



Crawling the Web

Non-profit organization

Data:

Publicly available on Amazon S3

E.g. January 2015: 140TB / 1.8B pages

Crawler:

Apache Nutch

collecting pre-defined list of URLs

Extracting text

[ABOUT](#) [CONTACT US](#) [FORUMS](#) [HOME](#) [LINUX HOW-TO & TUTORIALS](#) [SHELL SCRIPTS](#)

[RSS/FEED](#) 



Bash Shell: Find Out Linux / FreeBSD / UNIX System Load Average

by [NIXCRAFT](#) on MARCH 23, 2005 · [8 COMMENTS](#) · LAST UPDATED AUGUST 8, 2013
in [LINUX](#), [MONITORING](#), [SYS ADMIN](#)

Yes, I know we can use the `uptime` command to find out the system load average. The `uptime` command displays the current time, the length of time the system has been up, the number of users, and the load average of the system over the last 1, 5, and 15 minutes. However, if you try to use the `uptime` command in script, you know how difficult it is to get correct load average. As the time since the last reboot moves from minutes, to hours, and an even day after system rebooted. Just type the `uptime` command:

```
$ uptime
```



Google™ Custom Search



nixCraft

 Follow

+1

+ 137,320

 [LATEST LINUX/UNIX Q & A](#)

[How To Patch and Protect OpenSSL Vulnerability # CVE-2015-0291 CVE-2015-0204 \[19/March/2015 \]](#)

[load-average.html#comments](#) rel= nofollow >0 commentsLAST UPDATED

class="updated" title="2013-08-08">August 8, 2013</abbr><p

class='headline_meta'> in Linux, Monitoring, Sys admin

</p></div><div

class="format_text entry-content"><p><span

class="drop_cap">Yes, I know we can use the **<kbd>uptime</kbd>** command to find out the system load average. The uptime command displays the current time, the length of time the system has been up, the number of users, and the load average of the system over the last 1, 5, and 15 minutes. However, if you try to use the uptime command in script, you know how difficult it is to get correct load average. As the time since the last, reboot moves from minutes, to hours, and an even day after system rebooted. Just type the uptime command:<br

</span

id="more-631"><br

<code>\$ uptime</code><br

<pre>1:09:01 up 29 min, 1 user, load average: 0.00, 0.00, 0.00</pre>

<code>\$ uptime</code><br

<pre>2:13AM up 34 days, 16:15, 36 users, load averages: 1.56, 1.89, 2.06</pre><p>Traditionally, UNIX administrators used sed and other shell command in scripting to get correct value of load average. Here is my own modified hack to save the time<br

<code>\$ uptime | awk -F'load averages:' '{ print \$2 }'</code><br

<code>\$ uptime | awk -F'[a-z]:' '{ print \$2 }'</code><br

<pre> 1.24 1.34 1.35</pre><p>Output taken from my OS X desktop:</pre><pre> 0.00, 0.01, 0.05</pre><p>Output taken from my Ubuntu Linux server:</pre><pre> 0.24, 0.27, 0.21</pre><p>Output taken from my RHEL based server:</pre><pre> 0.71, 0.71, 0.58</pre><p>Please note that command works on all variant of UNIX operating systems.</pre><h2>See also</h2>

See chksysload.bash script to

HTML-2-Text v1: Strip Tags

LAST UPDATED August 8, 2013 in Linux , Monitoring , Sys admin Yes, I know we can use the uptime command to find out the system load average. The uptime command displays the current time, the length of time the system has been up, the number of users, and the load average of the system over the last 1, 5, and 15 minutes. However, if you try to use the uptime command in script, you know how difficult it is to get correct load average. As the time since the last, reboot moves from minutes, to hours, and an even day after system rebooted. Just type the uptime command: `$ uptime` Sample outputs: `1:09:01 up 29 min, 1 user, load average: 0.00, 0.00, 0.00`

HTML-2-Text v2: HTML5 parser



LAST UPDATED August 8, 2013
in Linux, Monitoring, Sys admin

Y

es, I know we can use the uptime command to find out the system load average. The uptime command displays the current time, the length of time the system has been up, the number of users, and the load average of the system over the last 1, 5, and 15 minutes. However, if you try to use the uptime command in script, you know how difficult it is to get correct load average. As the time since the last, reboot moves from minutes, to hours, and an even day after system rebooted. Just type the uptime command:

```
$ uptime
```

Sample outputs: 1:09:01 up 29 min, 1 user, load average: 0.00, 0.00, 0.00

Detecting Language

Muitas intervenções alertaram para o facto de a política dos sucessivos governos PS, PSD e CDS, com cortes no financiamento das instituições do Ensino Superior e com a progressiva desresponsabilização do Estado das suas funções, ter conduzido a uma realidade de destruição da qualidade do Ensino Superior público.

Detecting Language

Muitas intervenções alertaram para o facto de a política dos sucessivos governos PS, PSD e CDS, com cortes no financiamento das instituições do Ensino Superior e com a progressiva desresponsabilização do Estado das suas funções, ter conduzido a uma realidade de destruição da qualidade do Ensino Superior público.

Example langid.py

```
$ echo "Muitas intervenções alertaram" | \
    /home/buck/.local/bin/langid
('pt', -90.75441074371338)
```

Example langid.py

```
$ echo "Muitas intervenções alertaram" | \  
    /home/buck/.local/bin/langid  
( 'pt', -90.75441074371338)
```

```
echo "Muitas intervenções" |  
    /home/buck/.local/bin/langid  
( 'pt', -68.2461633682251)
```

Example langid.py

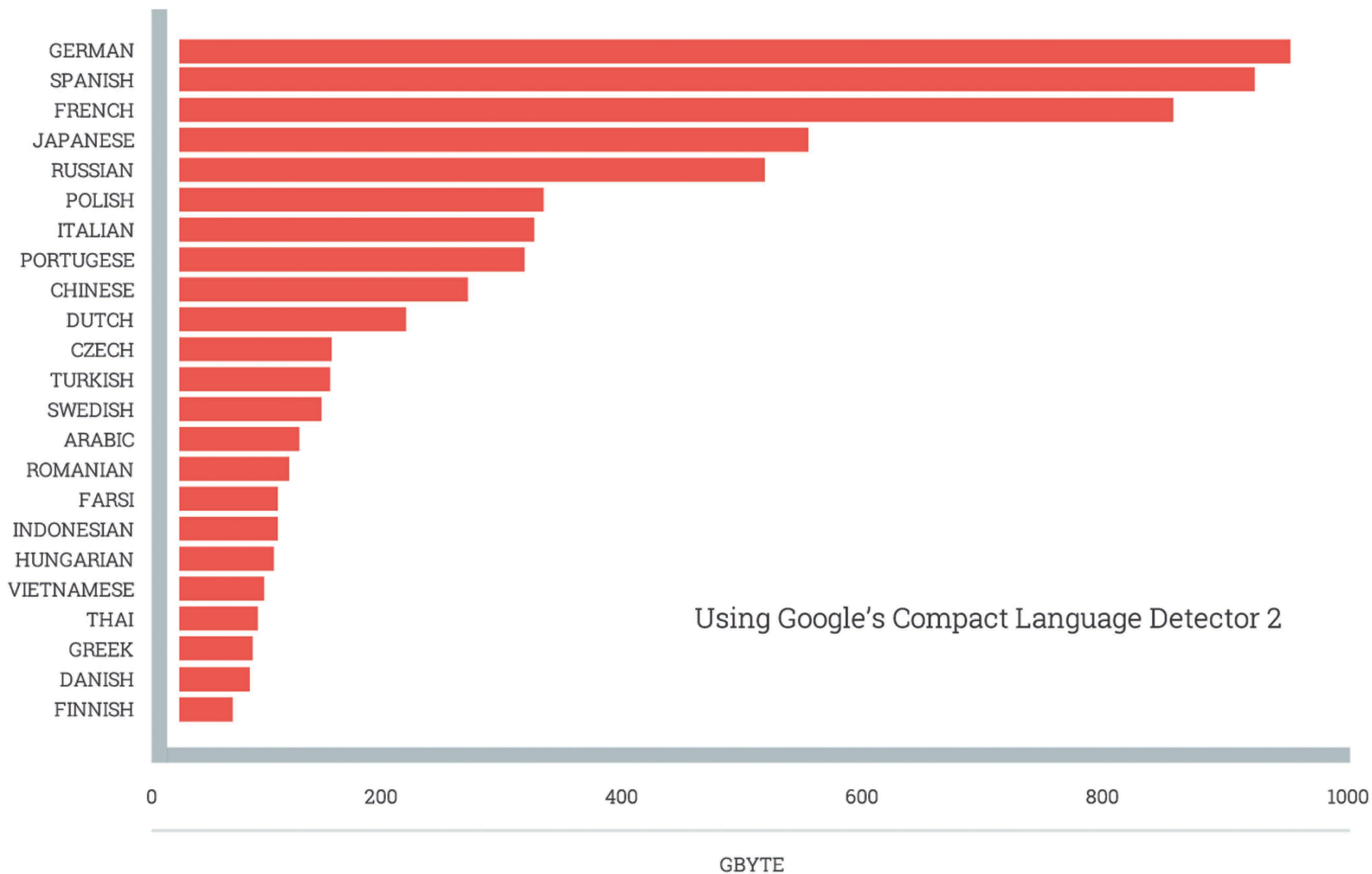
```
$ echo "Muitas intervenções alertaram" | \  
    /home/buck/.local/bin/langid  
( 'pt', -90.75441074371338)
```

```
echo "Muitas intervenções" |  
    /home/buck/.local/bin/langid  
( 'pt', -68.2461633682251)
```

```
echo "Muitas" | /home/buck/.local/bin/langid  
( 'en', 9.061840057373047)
```

Language Identification Tools

- langid.py (Lui & Baldwin, [ACL 2012](#))
1-4 grams, NaiveBayes, Feature Selection
- TextCat (based on Cavnar & Trenkle, 1994)
similar to langid.py
no Feature Selection
- Compact/Chromium Language Detector 2
takes hints from tld, meta data
super fast! By Google.
detects spans



Distribution of non-English languages in 2012/2013 CommonCrawl prior to de-duplication (Buck and Heafield, 2014)

Count (M)	Line
1374.44	Add to
816.33	Share
711.68	Unblock User
68.31	Sign in or sign up now!
61.26	Log in
54.77	Privacy Policy
45.18	April 2010
34.35	Load more suggestions
19.84	Buy It Now Add to watch list
16.64	Powered by WordPress.com

Most common English lines

Language	Lines (B)	Tokens (B)	Bytes
English	59.13	975.63	5.14 TiB
German	3.87	51.93	317.46 GiB
Spanish	3.50	62.21	337.16 GiB
French	3.04	49.31	273.96 GiB
Russian	1.79	21.41	220.62 GiB
Czech	0.47	5.79	34.67 GiB

	BLEU			
	2012	Δ	2013	Δ
Baseline	35.8		30.9	
+ 50M lines	36.3	0.5	31.5	0.6
+ 100M lines	36.5	0.7	31.5	0.6
+ 200M lines	36.6	0.8	31.8	0.9
+ 400M lines	37.0	1.2	31.8	0.9
+ 800M lines	37.3	1.6	31.8	0.9
+ 1.3B lines	37.7	1.9	32.0	1.1

Impact of LM size on English-Spanish MT quality

Mining Bilingual Text

"Same text in different languages"

- Usually: one side translation of the other
- Full page or interface/content only
- Potentially translation on same page
 - Twitter, Facebook posts
- Human translation preferred

Pipeline

1. Candidate Generation
2. Candidate Ranking
3. Filtering
4. Optional: Sentence Alignment
5. Evaluation

STRAND (Resnik, 1998, 1999)

**Structural
Translation
Recognition,
Acquiring
Natural
Data**



STRAND: parent pages

A page that links to different language versions

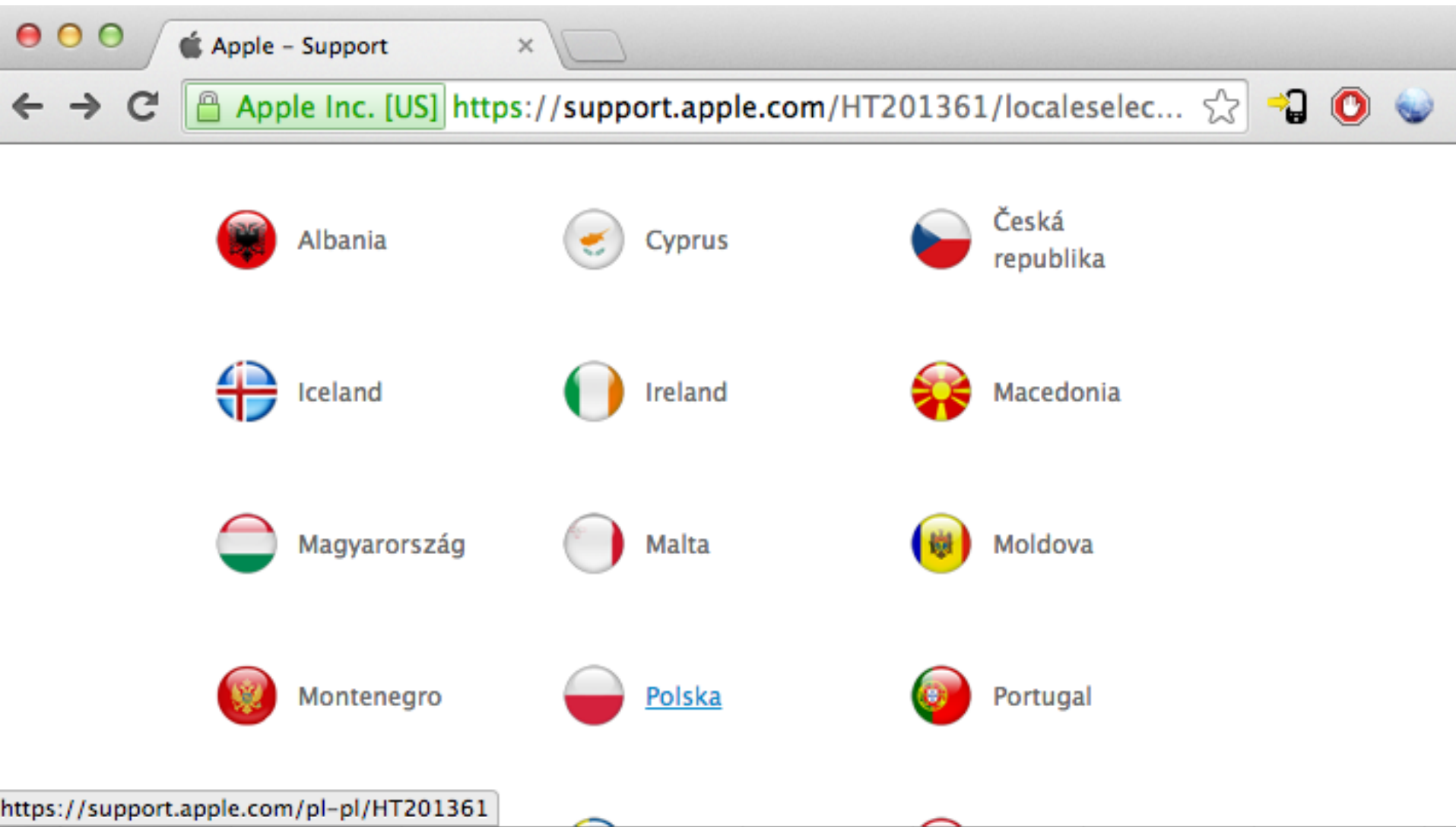


`x.com/en/cat.html`

`x.com/fr/chat.html`

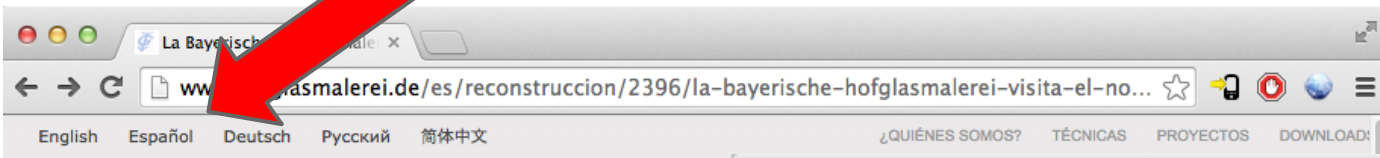
Require that links are close together

Example parent page



STRAND: sibling pages

A page that links to itself in another language



GUSTAV VAN TREECK
BAYERISCHE HOFGLASMALEREI

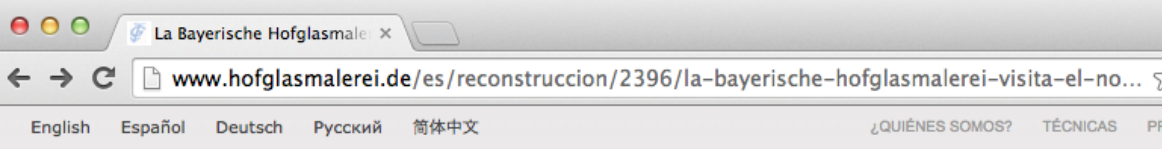
COMIENZ



LA BAYERISCHE HOFGLASMALEREI VISITA EL NORTE

Con especial atención a las Catedrales y trabajos de restauración en

Una delegación del Taller visitó en noviembre, las ciudades del norte español como L



GUSTAV VAN TREECK
BAYERISCHE HOFGLASMALEREI

COMIENZO

VIDRIO MODERNO

RESTAURACIONES



Candidate Generation without links

1. Find and download multilingual sites
2. Find some URL pattern to generate candidate pairs

xyz.com/en/	xyz.com/fr/
xyz.com/bla.htm	xyz.com/bla.htm?lang=FR
xyz.com/the_cat	xyz.fr/le_chat

Grep'ing for .*=EN (with counts)

545875	lang=en	33503	lang=eng
140420	lng=en	19421	uil=English
126434	LANG=en	15170	ln=en
110639	hl=en	14242	Language=EN
99065	language=en	13948	lang=EN
81471	tlng=en	12108	language=english
56968	l=en	11997	lang=engcro
47504	locale=en	11646	store=en
33656	langue=en		

Grep'ing for lang.*=.* (with counts)

13948	lang=EN	12003	lang=cz
13456	language=ca	11997	lang=engcro
13098	switchlang=1	11635	lang=sl
12960	language=zh	11578	lang=d
12890	lang=Spanish	11474	lang=lv
12471	lang=th	11376	lang=NL
12266	langBox=US	11349	lang=croeng
12108	language=english	11244	lang=English

Filtering Candidates: Length

Extract texts and compare lengths (Smith 2001)

$$\text{Length}(E) \approx C * \text{Length}(F)$$



learned,
language-specific parameter

Document- or sentence-level

Filtering Candidate: Structure

```
<html>  
  <body>  
    <h1>  
      Where is the cat?  
    </h1>  
    The cat sat on  
    the mat.  
  </body>  
</html>
```

```
<html>  
  <body>  
  
    El gato se sentó  
    en la alfombra.  
  </body>  
</html>
```

Filtering Candidate: Structure

```
<html>  
  <body>  
    <h1>  
      Where is the cat?  
    </h1>  
    The cat sat on  
    the mat.  
  </body>  
</html>
```

```
<html>  
  <body>  
    El gato se sentó  
    en la alfombra.  
  </body>  
</html>
```

Linearized Structure

[Start:html]

[Start:body]

[Start:h1]

[Chunk:17bytes]

[End:h1]

[Chunk:23bytes]

[End:body]

[End:html]

[Start:html]

[Start:body]

[Chunk:32bytes]

[End:body]

[End:html]

Levenshtein Alignment

[Start:html]	Keep
[Start:body]	Keep
[Start:h1]	Delete
[Chunk:17bytes]	Delete
[End:h1]	Delete
[Chunk:23bytes]	23 Bytes -> 32 Bytes
[End:body]	Keep
[End:html]	Keep

Variables characterizing alignment quality

dp	% inserted/deleted tokens
n	# aligned text chunks of unequal length
r	(Pearson) correlation of lengths of aligned text chunks
p	significance level of r

Variables characterizing alignment quality

dp	$\frac{3}{8} = 37.5\%$
n	1
r	$r = r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$ <p>... undefined</p>
p	... also undefined

Beyond structure

23 Bytes -> 32 Bytes

The cat sat on the mat.


El gato se sentó en la alfombra.

Content Similarity

23 Bytes -> 32 Bytes

The cat sat on the mat.

El gato se sentó en la alfombra.



The diagram illustrates the alignment between the English sentence "The cat sat on the mat." and the Spanish sentence "El gato se sentó en la alfombra." using thin gray lines. The connections are as follows: "The" connects to "El", "cat" connects to "gato", "sat" connects to "se", "on" connects to "sentó", "the" connects to "en", and "mat." connects to "alfombra".

Content Similarity

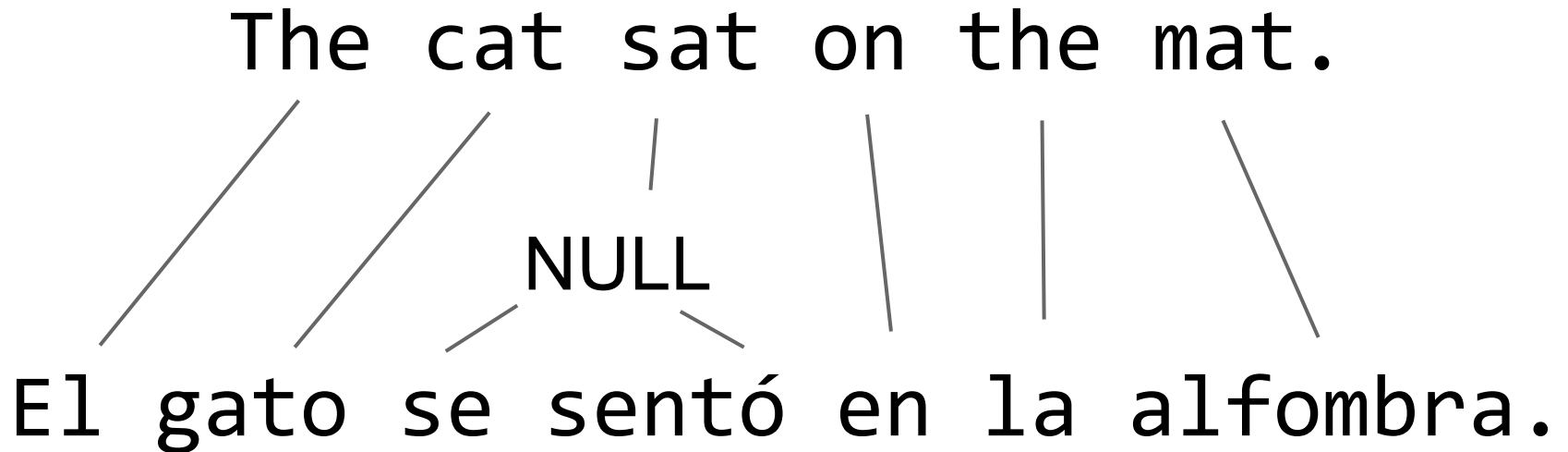
23 Bytes -> 32 Bytes

The cat sat on the mat.

El gato se sentó en la alfombra.

Diagram illustrating word alignment between the English sentence "The cat sat on the mat." and the Spanish sentence "El gato se sentó en la alfombra.". Lines connect corresponding words: "The" to "El", "cat" to "gato", "sat" to "se", "on" to "sentó", "the" to "en", and "mat." to "alfombra.". A "NULL" label is positioned between "se" and "sentó", with lines connecting to both words, indicating a shift or padding in the alignment.

Content Similarity



$$\text{tsim} = \frac{\text{two-word-links}}{\text{all links}} = \frac{5}{8}$$

Filtering with Features



Idea: Learn good/bad decision rule

Training data:

- Ask raters for content equivalence
- Positive examples easy

Challenges:

- Representative negative examples?
- Class skew
- Evaluation metric

Challenges

Translations on other sites

- siemens.com vs. siemens-systems.de
- News reported by different outlets

Machine Translation found

- Too high scores look suspicious

Partial Translations

SEO (keywords in URLs)



What Google does (or did in 2010)

For each non-English document:

1. Translate everything to English using MT
2. Find distinctive ngrams:
 - a. rare, but not too rare (5-grams)
 - b. used for **matching** only
3. Build inverted index: ngram -> documents

[cat sat on] -> {[doc_1, ES], [doc_3, DE], ...}

[on the mat] -> {[doc_1, ES], [doc_2, FR], ...}

Matching using inverted index

[cat sat on] -> {[doc_1, ES], [doc_3, DE], ...}

[on the mat] -> {[doc_1, ES], [doc_2, ES], ...}

[on the table] -> {[doc_3, DE]}

For each n-gram:

Generate all pairs where:

document list short (≤ 50)

source language different

{[doc_1, doc_3], ...}

Scoring using forward index

Forward index maps documents to n-grams

$n = 2$ for higher recall

For each document pair $[d_1, d_2]$:

- collect scoring n-grams for both documents

- build IDF-weighted vector

- distance: cosine similarity

Scoring pairs

$\text{ngrams}(d_1) = \{n_1, n_2, \dots, n_r\}$

$\text{ngrams}(d_2) = \{n'_1, n'_2, \dots, n'_r\}$

$\text{idf}(n) = \log(|D| / \text{df}(n))$

where: $|D|$ = number of documents

$\text{df}(n)$ = number of documents with n

$v_{1,x} = \text{idf}(n_x)$ if n_x in $\text{ngrams}(d_1)$, 0 oth.

$v_{2,x} = \text{idf}(n_x)$ if n_x in $\text{ngrams}(d_2)$, 0 oth.

$\text{score}(d_1, d_2) = v_1 \cdot v_2 / \|v_1\| * \|v_2\|$

Conclusion

General pipeline:

- Find pairs
 - Within a single site / All over the Web
 - URL restrictions
 - IR methods
- Extract features
 - Structural similarity
 - Content similarity
 - Metadata
- Score pairs



Reading Material

Uszkoreit et al:

[Large Scale Parallel Document Mining for Machine Translation](#), 2010

Resnik and Smith:

[The Web as a Parallel Corpus](#), 2003

Buck and Heafield:

[N-gram Counts and Language Models from the Common Crawl](#), 2014