Click to add text

Building a Wide Reach Corpus for Secure Parser Development

LangSec 2020

May 21, 2020

Jet Propulsion Laboratory
California Institute of Technology

The Team

Chris Mattmann
Deputy CTO
JPL PI

Tim Allison
Files and Search

Tom Barber
Doer/Maker

Wayne Burke
Cognizant Engineer

Valentino Constantinou
Data Scientist

Edwin Goh
Data Scientist

Eric Junkins
Data Scientist

Anastasia Menshikova
Data Scientist

Mike Milano
Data Scientist

Phil Southam
Trouble (Fun?) Maker

Ryan Stonebraker
Data
Scientist/Alaskan

Virisha Timmaraju
Data Scientist

Debts of Gratitude

Sergey Bratus
Peter Wyatt and Duff Johnson, PDF Association
Dan Becker, John Kansky and team at Kudu Dynamics
Trail of Bits, Galois, BAE and SRI

Outline

1. Motivation for LangSec Corpus Development
2. Background and Related Work
3. Gathering Files
4. Extracting Features
5. Visualizing Features

Motivation

Who needs files?

- Inducing grammars
- Devtesting parsers during development
- Testing/profiling/tracing existing parsers
  - Literal files
  - Seeds for fuzzing

Motivation

But I have ‘wget’ and ‘curl’, how hard can it be?!

Hyperlinks -- noisy, broken...and cycles!
Hyperlink graph coverage
Javascript rendered pages
Connectivity/bandwidth issues
Needles, haystacks
Coverage, coverage, coverage

Background and Related Work

Related Work

- Govdocs1
- Common Crawl
- Apache Tika’s regression corpus

Gathering Files

Two Approaches

- Common Crawl
- APIs

Common Crawl

- Monthly open source crawls of large portions of the web: for December 2019, 2.45 trillion pages (234 TB).
- Available via Amazon Web Services Public Datasets
- Searchable indexes available

https://commoncrawl.org/

Common Crawl Formats

- WARC - Web ARChive Format, http headers and literal bytes retrieved (47 TB*)
- WAT - Metadata files about the crawl (18 TB*)
- WET - Text extracted from X?HTML/Text (8 TB*)
- URL Index files - metadata for each URL (0.3 TB*)

Sizes are the compressed sizes for the December, 2019 crawl.

CommonCrawl HttpHeader Information

{date=Wed, 03 Jun 2015 21:34:52 GMT, server=Apache/2.2.3 (CentOS), expires=Tue, 03 Jun 2014 21:34:58 +0000, vary=Accept-Encoding, content-encoding=gzip, x-highwire-cache-cache-control=no-cache, content-disposition=inline; filename="1606.full.pdf", x-highwire-filestream-for=http://pdf.highwire.org/stamped/brain/135/5/1606.full.pdf, x-highwire-cache=no-cache, x-highwire-sitecode=brain, connection=close, content-type=application/pdf, cache-control=no-cache, max-age=0, must-revalidate, proxy-revalidate}

Observed Limitations of Common Crawl

- Files are truncated at 1MB (22% of PDFs in the December, 2019 crawl)
- Detected mime type not available in older crawls
- Scale of the data

Detected Mimes on 200-Status Pages in the 12/2019 Crawl

File Type	Counts
text/html	1,916,642,639
application/xhtml+xml	536,459,845
text/plain	68,596,968
message/rfc822	4,197,870
application/rss+xml	3,503,936
image/jpeg	3,405,543
application/atom+xml	3,292,446
application/pdf	3,275,094
application/xml	1,898,145
text/calendar	1,083,796

Website coverage: one deep dive

Search Engine	Condition	Number of Files
Google	site:jpl.nasa.gov	1.2 million
Bing	site:jpl.nasa.gov	1.8 million
Common Crawl	*.jpl.nasa.gov	128,406
Google	site:jpl.nasa.gov filetype:pdf	50,700
Bing	site:jpl.nasa.gov filetype:pdf	64,300
Common Crawl	*.jpl.nasa.gov mime= pdf	7

Common Crawl Takeaways

Extraordinarily useful for gathering heaps of files

No guarantees on coverage of the web

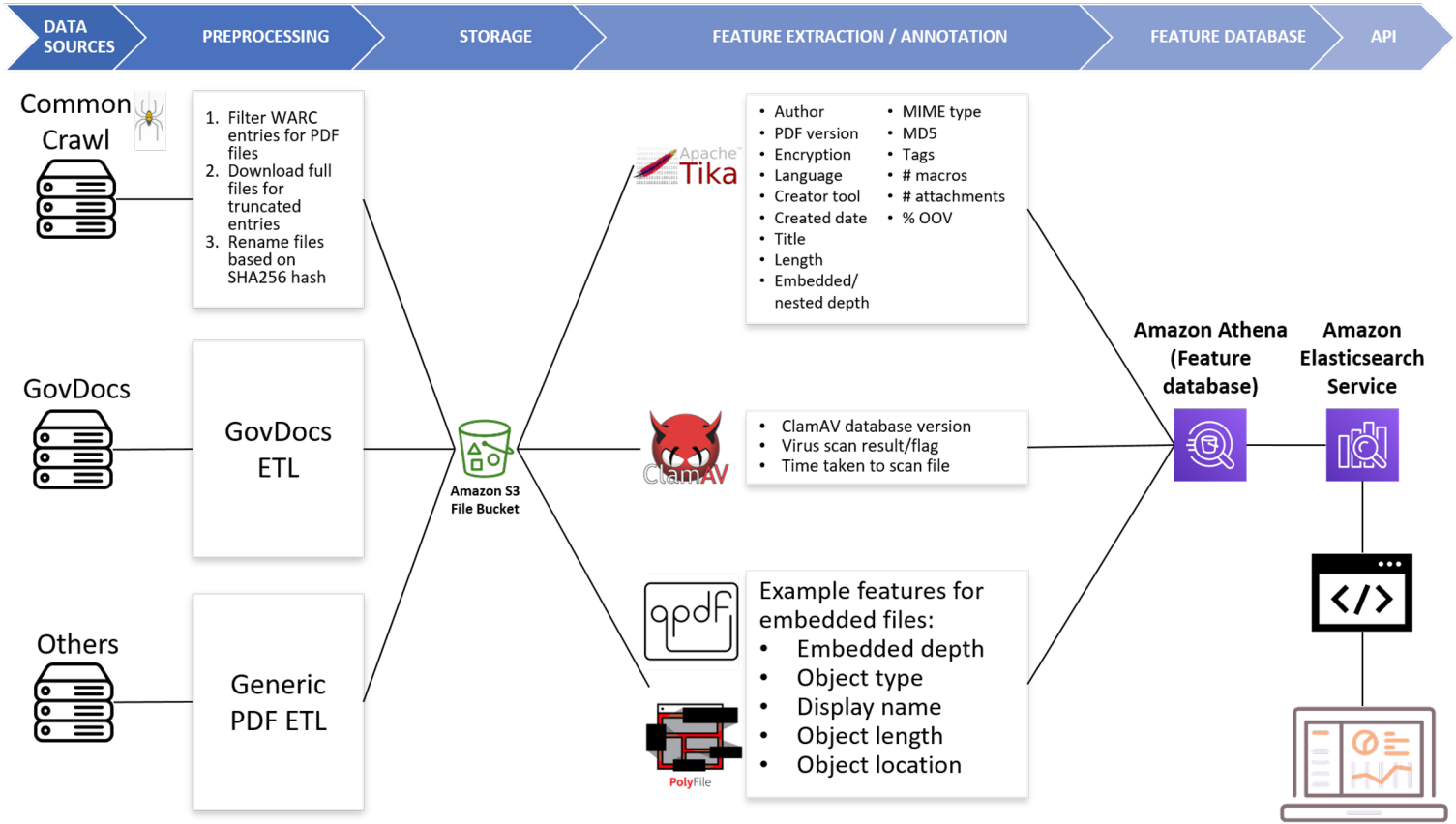
Some post processing/refetching required

Web crawling generally: No guarantees of representativeness of files in “typically” offline domains

Common Crawl: How we've used it

- Gathered 30 million unique PDFs to date
- Refetched the truncated PDFs
- Stored provenance (and WARC metadata) in AWS Athena

Architectural Flyby



Custom Crawlers/APIs

- Issue trackers can have non-optimal hyperlink structures
- We've used APIs for Bugzilla and JIRA based issue trackers so that we can query and gather issues with attachments.
- For a handful of sites, we have custom crawlers

Files, files and more files: Issue tracker data

```
ghostscript    mozilla_general  ooo       poi       sumatrapdf
libre_office   mozilla_pdfjs    openpdf   poppler   tika
librepdf       ocrmypdf         pdfbox    qpdf
```

- 27,000 PDFs (20 GB)
- Post-processed compression/package files:
  - **`PDFBOX-975-0.zip-3.pdf`**

Extracting Features

Features, features and more features

- Internal metadata (Apache Tika)
- ClamAV hits (ClamAV)
- PolyFile structural elements
- Error messages, exit values, processing times from standard commandline PDF processing tools: pdftotext, pdftops, pdfinfo, caradoc, pdfid

Status: Extracting Features into AWS

tika-annotate - Apache Tika Annotator

Goal: Generate an extensive set of descriptors for a targeted search of documents and capability test of performer solutions.

Method: Using the python wrapper for Apache Tika, a Java-based content detection and analysis framework.

Why Tika: Capable of extracting metadata and content for 1400 file formats.

Outcomes:

- Successfully scanned and generated the following descriptors (in the table) for the JPL workshop demo documents.

Author	U.S Government Printing Office
PDF Version	1.4
Digital Signature	False
Creator Tool	ACOMP.exe WinVer 1b43 jul 14 2003
Producer	Acrobat Distiller 4.0 for Windows
Application Type	PDF
Number of Pages	4
Number of Annotations	3

Descriptors extracted using tika-annotate with example output

Status: Extracting Features into AWS

av-annotate - ClamAV Go(lang) Annotator

Goal: develop a performant means of scanning and labeling documents for “malicious” documents against known signatures.

Method: use Go as a wrapper around the multi-threaded scanner daemon, clamd → rapid scanning of thousands of files.

Why ClamAV: benchmark of a currently-standard tool, another point of comparison for SafeDocs parsers and a helpful document annotation.

Outcomes:

- Works well against a set of malicious JPL emails used as part of the DARPA ASED program (many positive detections).
- Small amount of positive detections against GovDocs and JPL workshop demo documents (little positive detections).
  - We **need** SafeDocs parsers!

JPL Abuse Malicious Emails ***(n=3128)***	
Signature	*Count*
Doc.Macro.MaliciousHeuristic-6329080-0	34
Win.Trojan.Agent-5440575-0	26

Documents in Paper Corpus ***(n=~20000)***	
Signature	*Count*
Pdf.Exploit.CVE_2018_4882-6449963-0	1

Common Crawl WARC info

t	common_crawl__rec_headers.content-length	67326
t	common_crawl__rec_headers.content-type	application/http; msgtype=response
t	common_crawl__rec_headers.warc-block-digest	sha1:ZM35COC3SSMGLD43EYE6BKOYSSOJ4VVG
t	common_crawl__rec_headers.warc-concurrent-to	<urn:uuid:ffcf9128-27db-4a27-ac48-c4f72c5cce86>
🕘	common_crawl__rec_headers.warc-date	Apr 18, 2015 @ 16:48:53.000
✱ t	common_crawl__rec_headers.warc-ip-address	69.18.213.152
t	common_crawl__rec_headers.warc-payload-digest	sha1:3CWR2AIVNPJUMV6U6WWDERAHASNUQACG
t	common_crawl__rec_headers.warc-record-id	<urn:uuid:791a172e-9d35-4c08-8a3c-fc207f171fa2>
t	common_crawl__rec_headers.warc-target-uri	http://www.nyserda.ny.gov/About/Board-Governance/-/media/Files/About/Board-Governance/Board-and-committee-meetings/BoardAgendas/Board-Agenda-2009Dec.ashx
t	common_crawl__rec_headers.warc-type	response

Metadata extracted by Apache Tika

🕒	tika__created	Apr 22, 2014 @ 11:18:38.000
t	tika__creator_tool	Xerox WorkCentre 7345
#	tika__embedded_depth	0
t	tika__format	application/pdf; version=1.6
#	tika__inline_attachments_per_page	1
◑	tika__is_embedded	false
t	tika__lang_detected	eng
#	tika__lang_detected_conf	0.352
t	tika__mime	application/pdf
t	tika__mime_detailed	application/pdf
◑	tika__missing_content	false
🕒	tika__modified	Apr 22, 2014 @ 11:18:57.000
#	tika__num_alpha_tokens	261

PolyFile and QPDF keys (for now)

t	polyfile__keys	/Lang, /MarkInfo, /Marked, /Metadata, /Names, /Pages, /StructTreeRoot, /Type, /ViewerPreferences, /DisplayDocTitle, /Length, /Subtype, /IDTree, /Count, /Kids, /ClassMap, /K, /ParentTree, /ParentTreeNextKey, /CM1, /O, /StartIndent, /TextAlign, /CM2, /LineHeight, /TextIndent, /CM3, /CM4, /CM5, /CM6, /SpaceAfter, /CM7, /Nums, /Obj, /Pg, 5, /P, /S, 20, /ID, /A, /C, /Annots, /Contents, /MediaBox, /Parent, /Resources, /Font, /T1_0, /ProcSet, /PDF, /ImageB, /XObject, /Im0, /Rotate, /StructParents, /Tabs, /BitsPerComponent, /ColorSpace, /DecodeParms, /Columns, /Filter, /Height, /Width, /BaseFont, /Encoding, /BaseEncoding, /Differences, /notequal, /greaterequal, /space, /FirstChar, /LastChar, /ToUnicode, /Widths, /BS, /W, /Border, /Rect, /StructParent, /URI, /, /governancemeetings.asp, /EndIndent, /ListNumbering, /CreationDate, /Creator, /ModDate, /NCCL_DocId, /Producer, /Title, /NCCL_App, /NCCL_Standard, /NCCL_Status
t	qpdf__err_txt	-
t	qpdf__keys	/A, /Annots, /BS, /BaseEncoding, /BaseFont, /BitsPerComponent, /Border, /C, /CM1, /CM2, /CM3, /CM4, /CM5, /CM6, /CM7, /ClassMap, /ColorSpace, /Columns, /Contents, /Count, /CreationDate, /Creator, /DecodeParms, /Differences, /DisplayDocTitle, /Encoding, /EndIndent, /Filter, /FirstChar, /Font, /Height, /ID, /IDTree, /Im0, /Info, /K, /Kids, /Lang, /LastChar, /Length, /LineHeight, /ListNumberi

Features, features and more features

An oversimplification of structural hierarchy

Use	Text and metadata	Rendering	Interactivity

Embedded Resources

XMP, XFA, JS, fonts, multimedia, ICC profiles...and?

Document Graph

Putting the objects together. Issues: orphaned objects, infinite loops in references...

Object/Stream Parsing

```
32 0 obj stream
...endstream...
...endstream...
endobj
33 0 obj
```

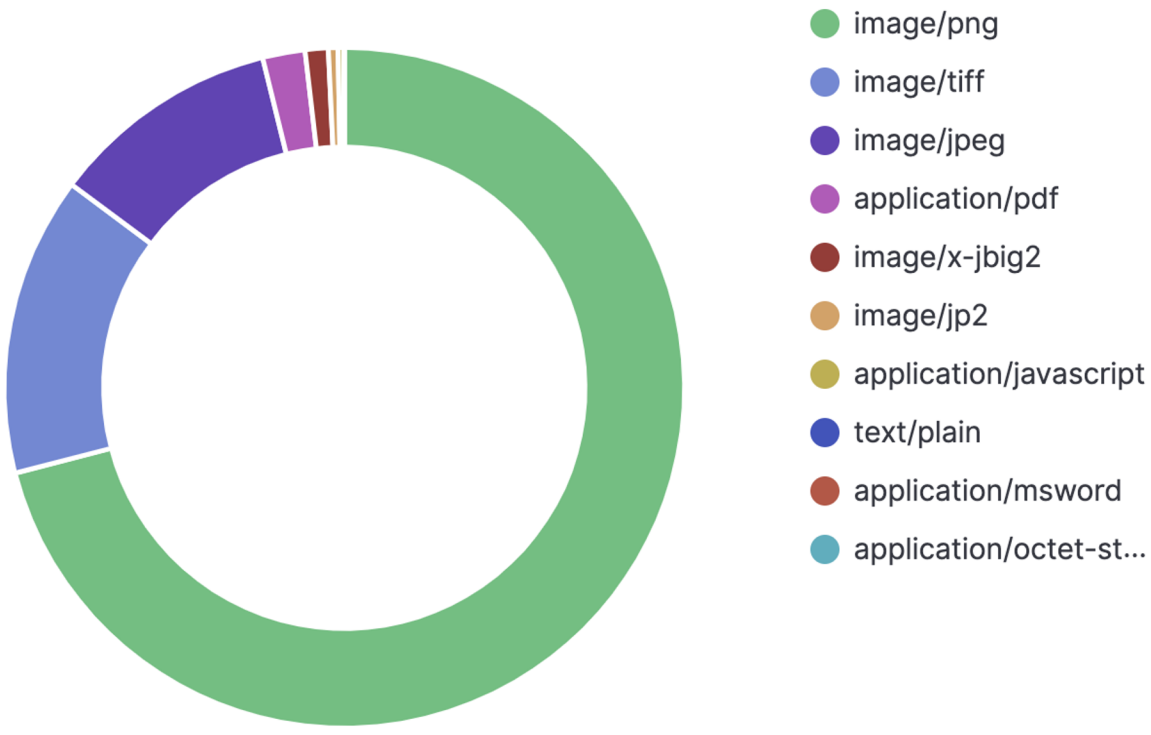
→ Where does the stream actually end?

Tokenization

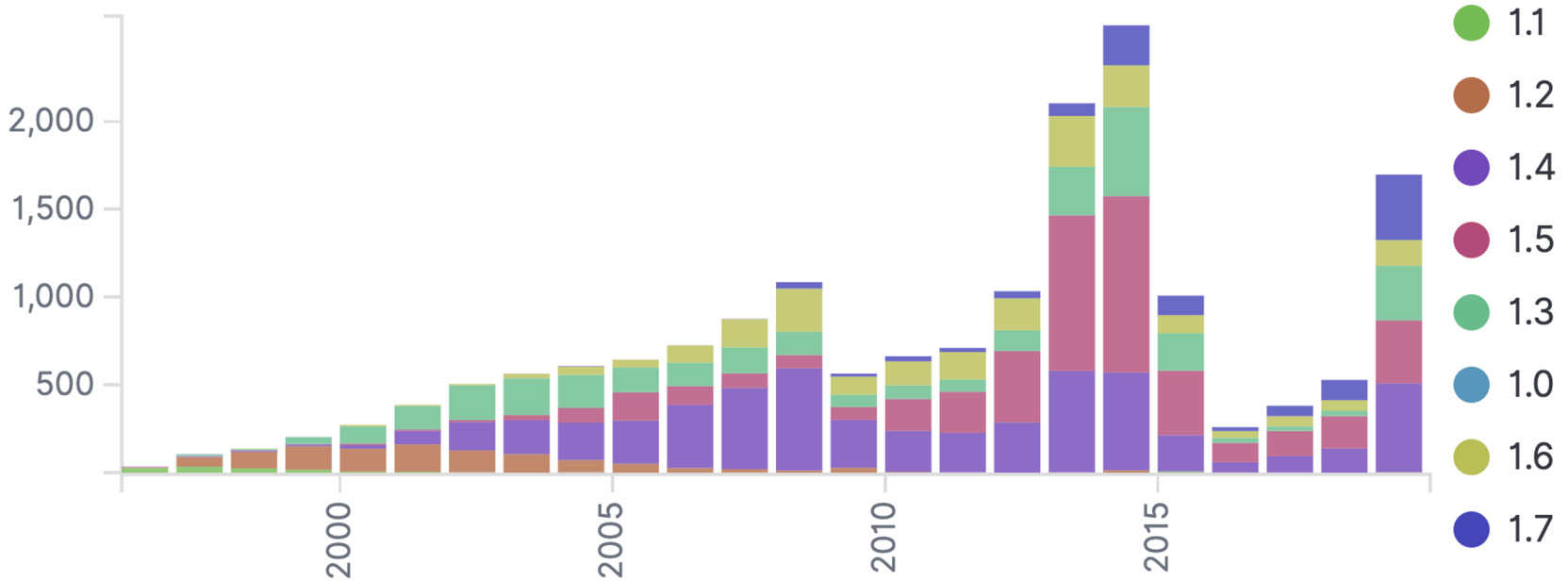
`...32 0 R %comment 33 0 R` → Reference Token: “32 0 R”

Visualizing Features with Kibana

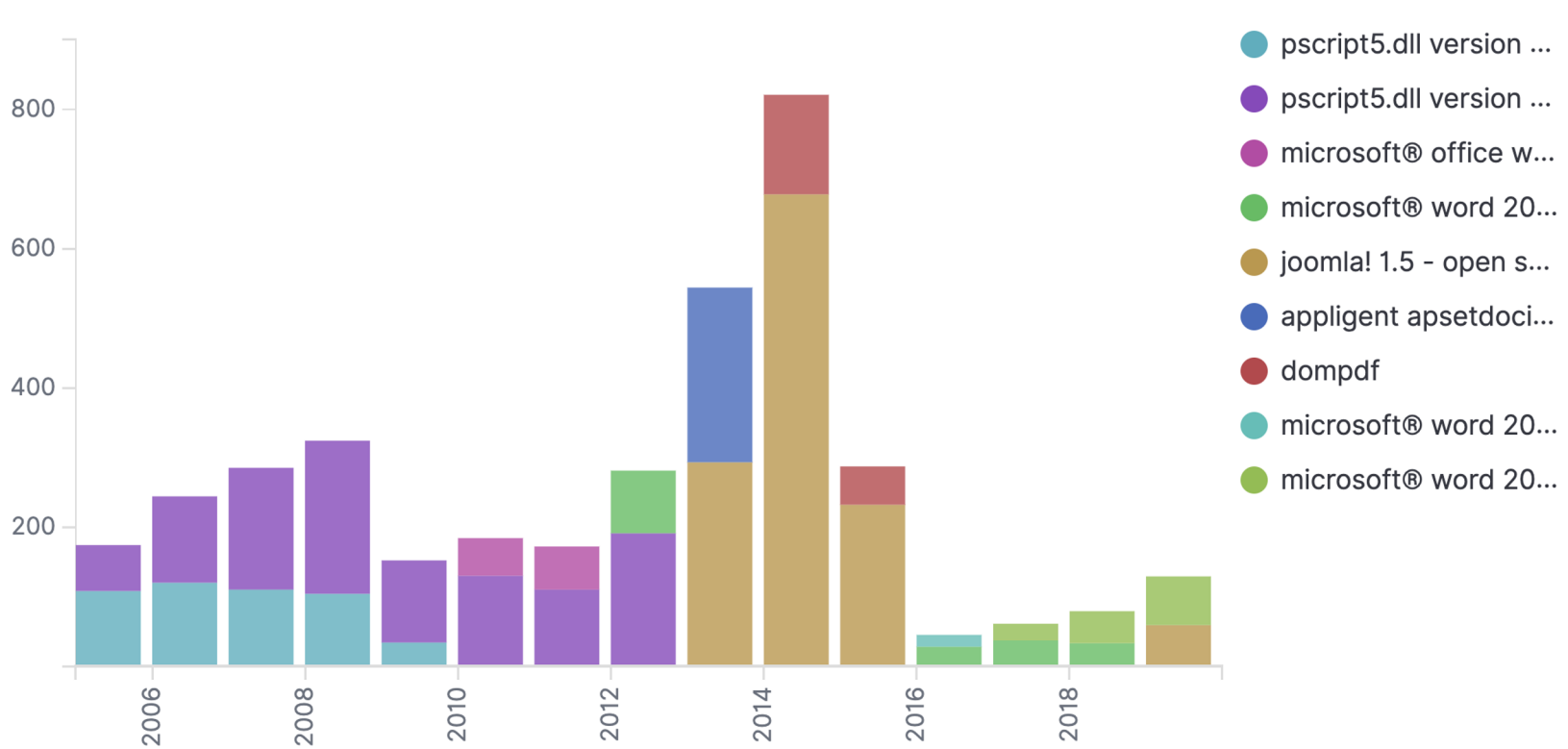
File types: Containers and embedded files



PDF Version by Created Date



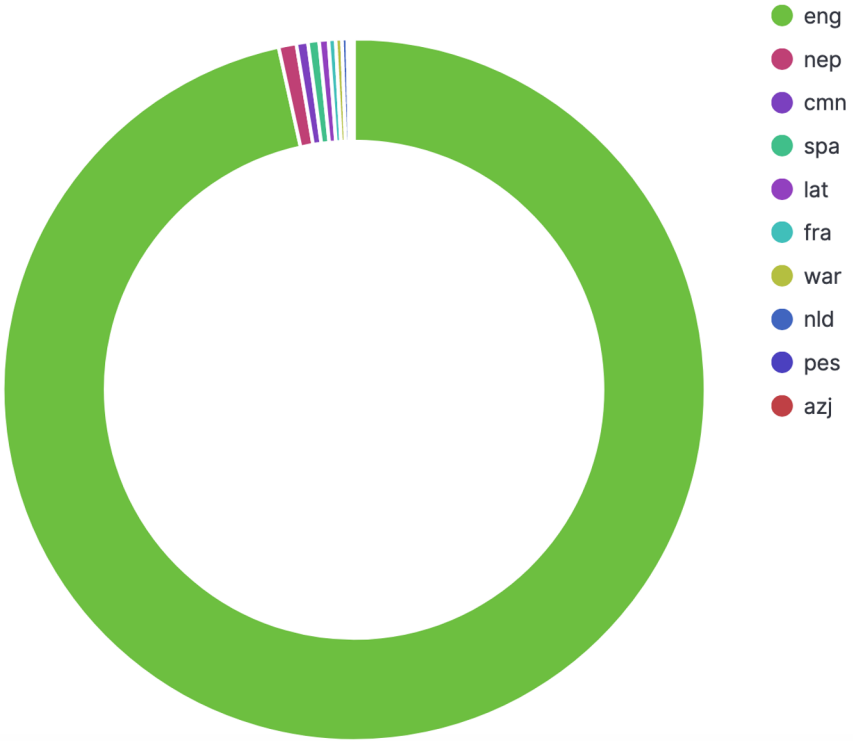
Creator tools by year



Detected Languages

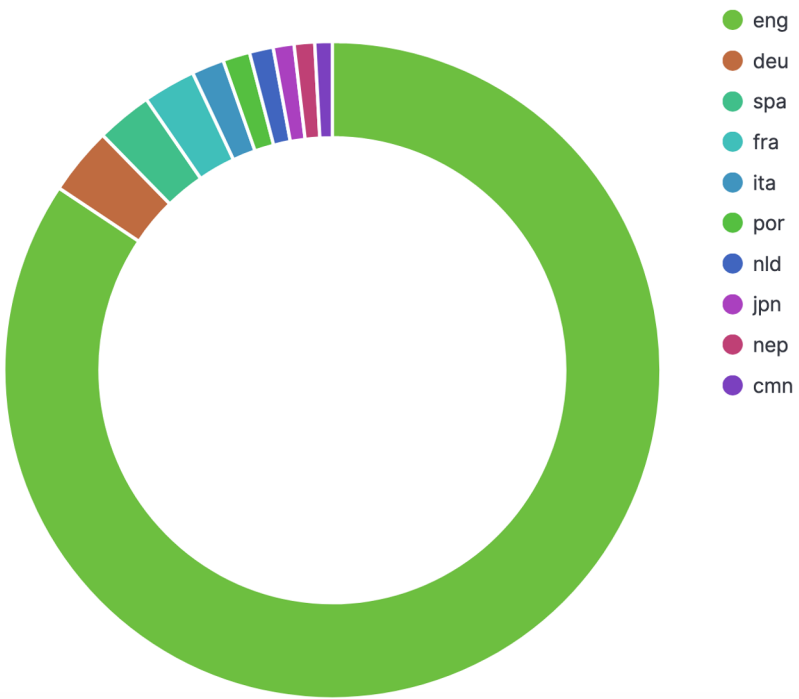
Govdocs1

collection: govdocs × | embedded_depth: 0 × | + Add filter

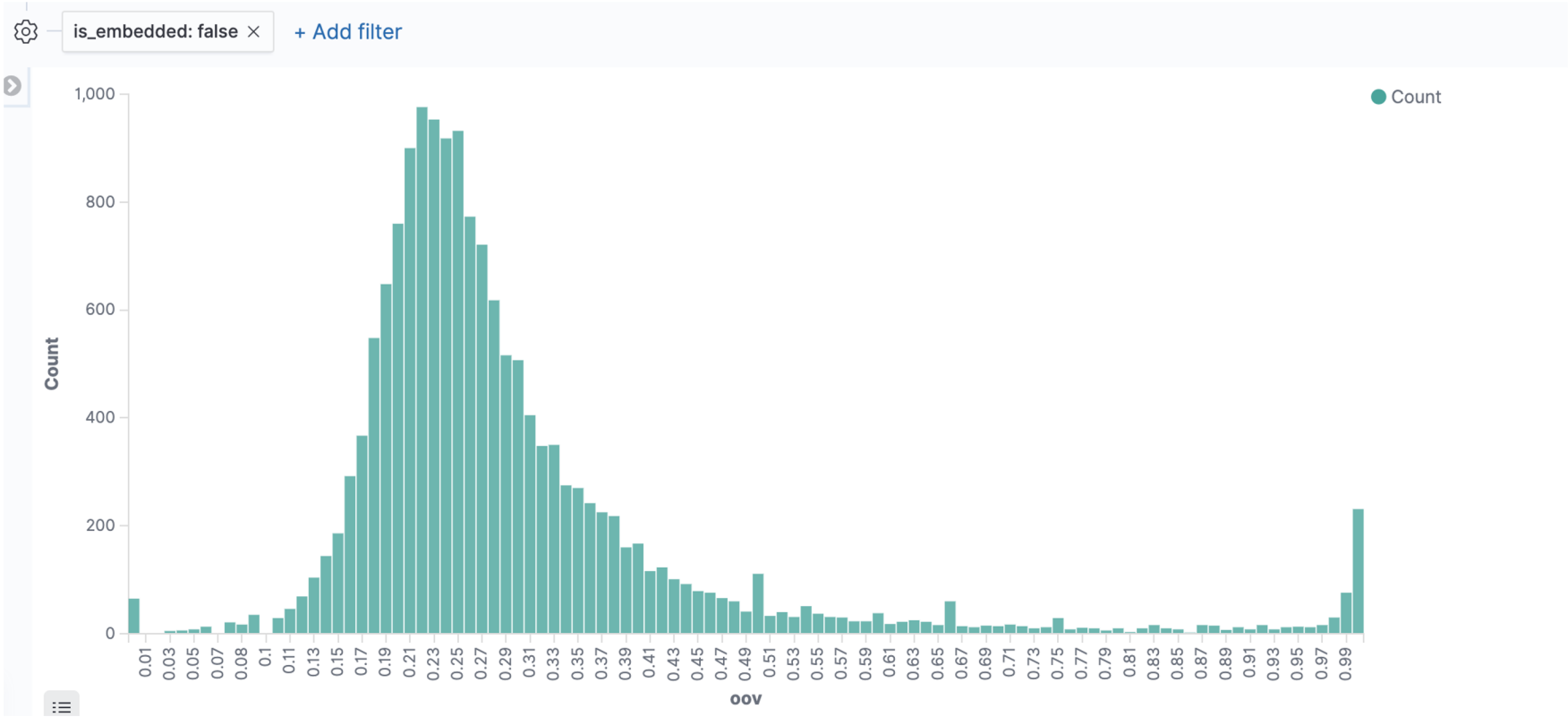


Common Crawl

collection: common_crawl2 × | embedded_depth: 0 × | + Add filter



Histogram of Out of Vocabulary (OOV) %



Sort by OOV% descending

Significant Terms -- What Keys Appear More Frequently in Version 1.7 vs 1.6

```
GET safedocs-document-meta/_search
{
  "query" : {
    "match" : {
      "tika__pdf_version" : {
        "query":"1.7"
      }
    }
  },
  "size" :0,
  "aggregations" : {
    "significant_queries" : {
      "significant_terms" : {
        "field" : "polyfile__keys.keyword",
        "size": 50,
        "chi_square": {
          "background_is_superset" : false
        },
        "background_filter": {
          "match" : {
            "tika__pdf_version" : {
              "query":"1.6"
            }
          }
        },
        "min_doc_count": 10
      }
    }
  }
}
```

```
  "hits" : {
    "total" : {
      "value" : 927,
      "relation" : "eq"
    },
    "max_score" : null,
    "hits" : [ ]
  },
  "aggregations" : {
    "significant_queries" : {
      "doc_count" : 927,
      "bg_count" : 1711,
      "buckets" : [
        {
          "key" : "/DisplayDocTitle",
          "doc_count" : 85,
          "score" : 98.33224376865637,
          "bg_count" : 21
        },
        {
          "key" : "/Extensions",
          "doc_count" : 47,
          "score" : 88.3233456446408,
          "bg_count" : 0
        },
        {
          "key" : "/BaseVersion",
          "doc_count" : 46,
          "score" : 86.4107751674724,
          "bg_count" : 0
        },
```

Next Steps

Corpora

- “Publish” issue tracker PDFs

Features

- More tools, more commandline options

Analysis and visualization

- Correlations, clustering of features and visualizations

Long term

- Corpus minimization (cmin) (thank you, John Kansky)

Questions/Discussion

- Thank you!

- Contact info:
- timothy.b.allison@jpl.nasa.gov (@_tallison)
- vconstan@jpl.nasa.gov

Jet Propulsion Laboratory
California Institute of Technology

jpl.nasa.gov

Extras

Features, features and more features

An oversimplification of structural hierarchy

Use	Text and metadata	Rendering	Interactivity

Embedded Resources

XMP, XFA, JS, fonts, multimedia, ICC profiles...and?

Document Tree

Putting the objects together. Issues: orphaned objects, infinite loops in references...

Object/Stream Parsing

```
32 0 obj stream
...endstream...
...endstream...
endobj
33 0 obj
```

→ Where does the stream actually end?

Tokenization

```
...32 0 R %comment 33 0 R
```

→ Reference Token: “32 0 R”