TL;DR

1. Search is Easy
2. Search is Hard
3. Search has many shades of grey
About DigitalGov Search

- Search as a Service for ~1500 gov/mil sites
- Citizens get commercial search results augmented with customer-specific content
- Agencies get powerful and timely analytics
On the Search Side

- Many different document types, from tweets to PDFs
- Some small, some big (~1 Billion documents)
On the Analytics Side
Transparency: Tech meets Data

Databases
Indexes

Search
Exploration
Discovery

Downloads
Archives
Search is Easy

PUT /contacts/entry/1

{  "name": "National Security Agency",
  "city": "Fort Meade",
  "state": "MD",
  "notes": "summer intern job"}

GET /contacts/entry/_search?q=Agency

"National Security Agency"
That worked, but ...

{ "name": "National Security Agency",
  "city": "Fort Meade",
  "state": "MD",
  "notes": "summer intern job"}
Search is Hard

\[ y = \sqrt{x} \]
Recall & Relevancy

**recall:** fraction of *relevant* documents that are *retrieved*

**relevancy:** fraction of *retrieved* documents that are *relevant*
TF-IDF

- The more the term appears in a document, the higher the term frequency (TF).
- The more the term appears across the corpus, the lower the inverse document frequency (IDF).
- Additional signal can help improve relevancy.
Popular Search Software

Lucene

Solr

Elasticsearch
## Sprinkle Search Magic

```json
{
  "name": "National Security Agency",
  "city": "Fort Meade",
  "state": "MD",
  "notes": "summer intern job"
}
```

<table>
<thead>
<tr>
<th>Query term</th>
<th>Problem</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ft. Meade</td>
<td>Ft. vs Fort</td>
<td>synonyms</td>
</tr>
<tr>
<td>md</td>
<td>case</td>
<td>downcase</td>
</tr>
<tr>
<td>the National Security Agency</td>
<td>the</td>
<td>stopwords</td>
</tr>
<tr>
<td>National Securite Agéncy</td>
<td>spelling, accent</td>
<td>fuzziness, folding</td>
</tr>
<tr>
<td>interns</td>
<td>word form</td>
<td>stemming</td>
</tr>
</tbody>
</table>
GET /contacts/entry/_search?q=What%20%C3%85gencies%20in%20Maryland%20have%20interns%3F

“National Security Agency”
“summer intern job”
Ship it!

LOREN

ARE YOU WIZARD?
Demo Day

GET /contacts/entry/_search?q=Internal%20Revenue%20Service
Demo Day

GET /contacts/entry/_search?q=Internal%20Revenue%20Service

“summer intern job”
Demo Day

GET /contacts/entry/_search?q=Agency%20for%20International%20Development
Demo Day

GET /contacts/entry/_search?q=International%20Development

“summer intern job”
# A Snowball’s Chance in English

<table>
<thead>
<tr>
<th>Raw Term</th>
<th>Stemmed Token</th>
</tr>
</thead>
<tbody>
<tr>
<td>interns, internal, international</td>
<td>intern-</td>
</tr>
<tr>
<td>securities, security</td>
<td>secur-</td>
</tr>
<tr>
<td>Maine, main</td>
<td>main-</td>
</tr>
<tr>
<td>season, seasoning</td>
<td>season-</td>
</tr>
<tr>
<td>image, imaging</td>
<td>imag-</td>
</tr>
<tr>
<td>physics, physical</td>
<td>physic-</td>
</tr>
<tr>
<td>IRS</td>
<td>ir-</td>
</tr>
</tbody>
</table>
Best Practices

A search system is a database with an opinion. Where does it get that opinion?

- Sensible defaults get you pretty far.
- Analysis chain is where the effort goes.
- Refinement is ongoing.
- Make it easy to reindex.
Search APIs & Data Products

What do you expose?
● Schema?
● Filters?
● Lucene itself?

How much “search magic” is enough?
● Stemming, synonyms, stopwords, etc
Thank you!

http://search.digitalgov.gov
202-505-5315  |  @DG_Search