The history of the **Battle of Midway**

Data Cleaning with **C#/.NET**

**Named Entity Recognition** via **Machine Learning**

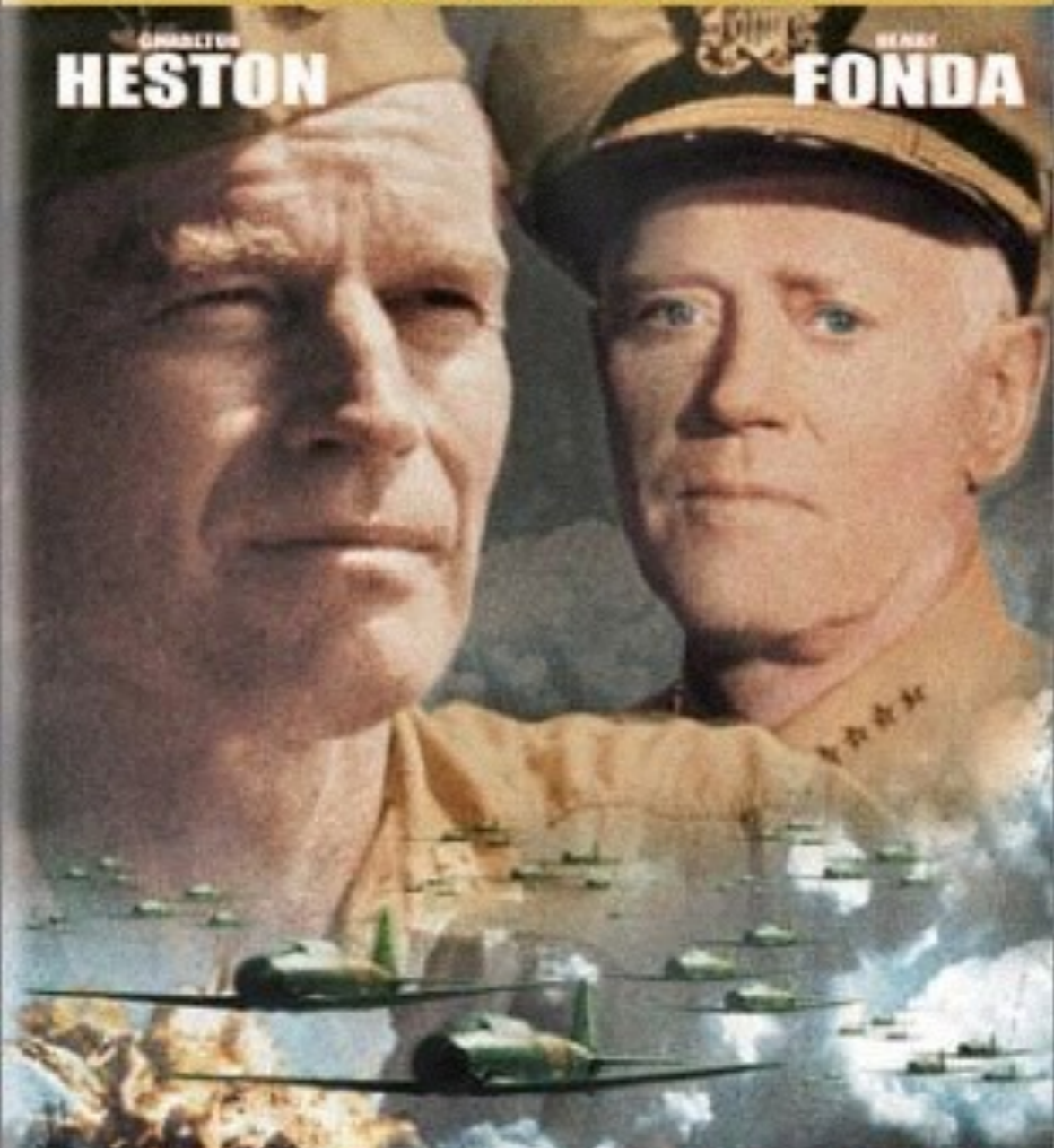History visualized in a **Xamarin.iOS** mobile app

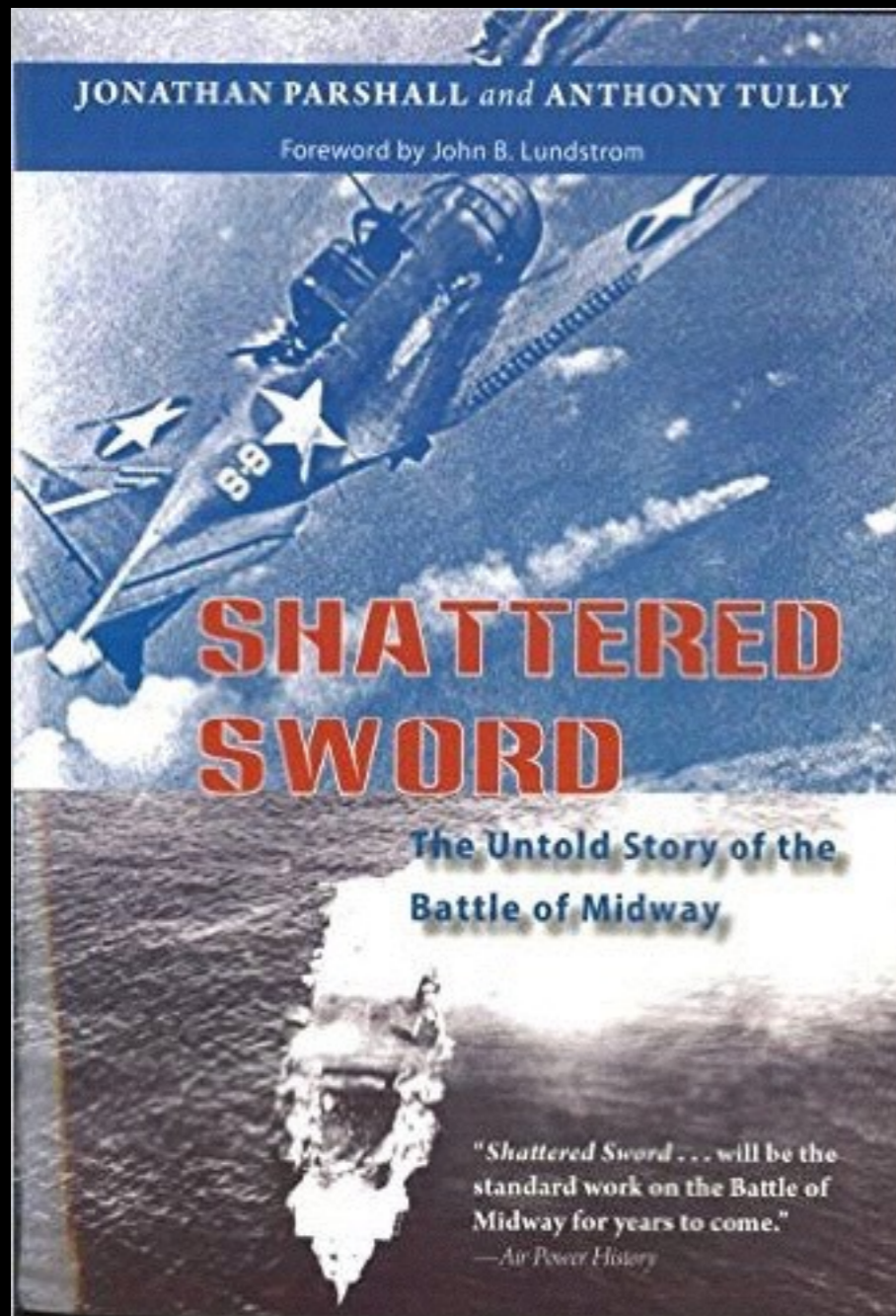Dan Edgar

New York Times Bestseller

# MIRACLE AT MIDWAY

## Gordon W. Prange
### with Donald M. Goldstein
### and Katherine V. Dillon

# Shattered Sword: The Untold Story of the Battle of Midway



Local Minneapolis author Jonathan Parshall

Great telling of the entire story of the Battle of Midway from the perspective of the Japanese

# Importance of Midway



Credit: https://en.wikipedia.org/wiki/Battle_of_Midway#/media/File:Midway_Atoll.jpg

# Carriers More Important Than Holding Midway

- Yorktown

- Enterprise

- Hornet

# Why Invade Midway?

# Historical View of Japanese Battle Plan for Invasion of Midway

It is necessary now to turn to an examination of Yamamoto's operational plan as it emerged in its final form [for the invasion of Midway], a task for which the reader would be well advised to pour a rather tall glass of spirits beforehand.

— Shattered Sword: The Untold Story of The Battle of Midway by Jonathan Parshall and Anthony Tully

# Japanese Navy - In need of R&R

Over **70%** of pilots at Midway were also at the raid on **Pearl Harbor**

Japan only had **2 ships with radar** in the entire fleet in 1942. The 'Mark 1 Eyeball' was the primary way to find enemy fleets.

Japan would construct only **56 attack aircraft** during all of 1942

At the same time Japan attacked Midway, they attacked the Aleutians. Nearly the entire Japanese Navy was committed to Midway or the Aleutians

# Losses at Midway - Significance

Only **2 U.S. Aircraft** losses were due to Japanese anti-aircraft fire.

More losses of aircraft due to landing accidents than anti-aircraft fire

U.S. dive bombers were faster than Japanese anti-aircraft guns

|  | U.S. | Japan |
|---|---|---|
| Casualties | 307 | 2,500 |
| **Carriers** | **1** | **4** |
| Heavy Cruiser | 0 | 1 |
| Destroyer | 1 | 0 |
| **Aircraft** | **147** | **332** |

Books and Movies tell the
story from the top —> down

*Yorktown*            *Akagi*                              *Hornet*

*Woodson*

*Wasp*                                                *Weber*

*Hiryu*                              *Soryu*

*Saratoga*

*Spruance*                          *Thatch*            *VT-3*

*Fletcher*       *Bomber Squadron 3*              *Yamamoto*

History is also individual stories of
people from the bottom —> up

# Engineering Importance

- There were over 1,600 mechanics and aircraft engineers on the 4 Japanese carriers

| Kaga | Akagi |
|------|-------|
| Soryu | Hiryu |

# More than just combat stories…

https://en.wikipedia.org/wiki/Tooth-to-tail_ratio



16%

45%     39%

**Combat**

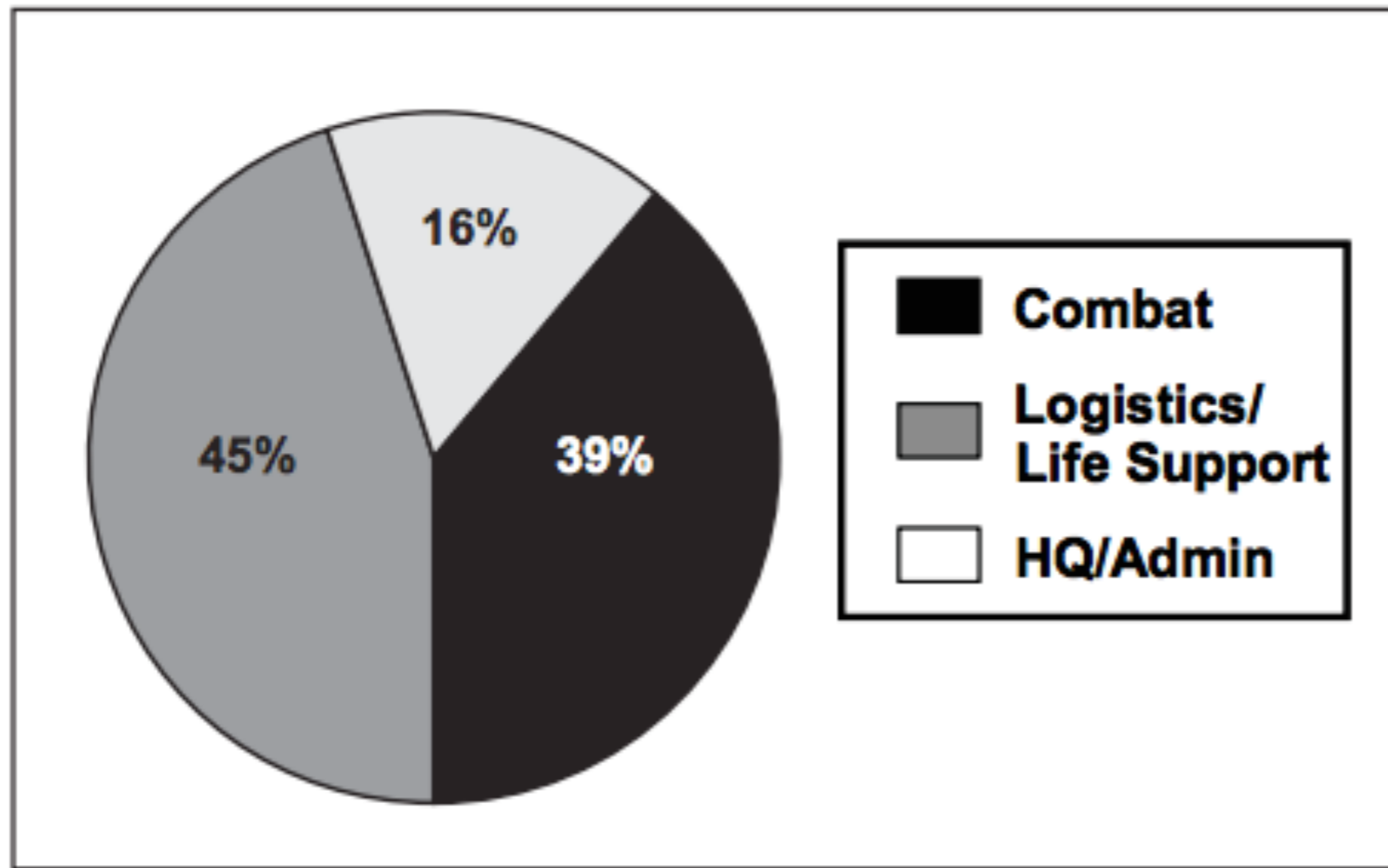**Logistics/
Life Support**

**HQ/Admin**

Figure 9. European Theater of Operations by Category, April 1945.[16]

Unlock the stories
from the **61%**

http://usacac.army.mil/cac2/cgsc/carl/download/csipubs/
mcgrath_op23.pdf

New York Times Bestseller

# MIRACLE AT MIDWAY

## Gordon W. Prange
### with Donald M. Goldstein
### and Katherine V. Dillon
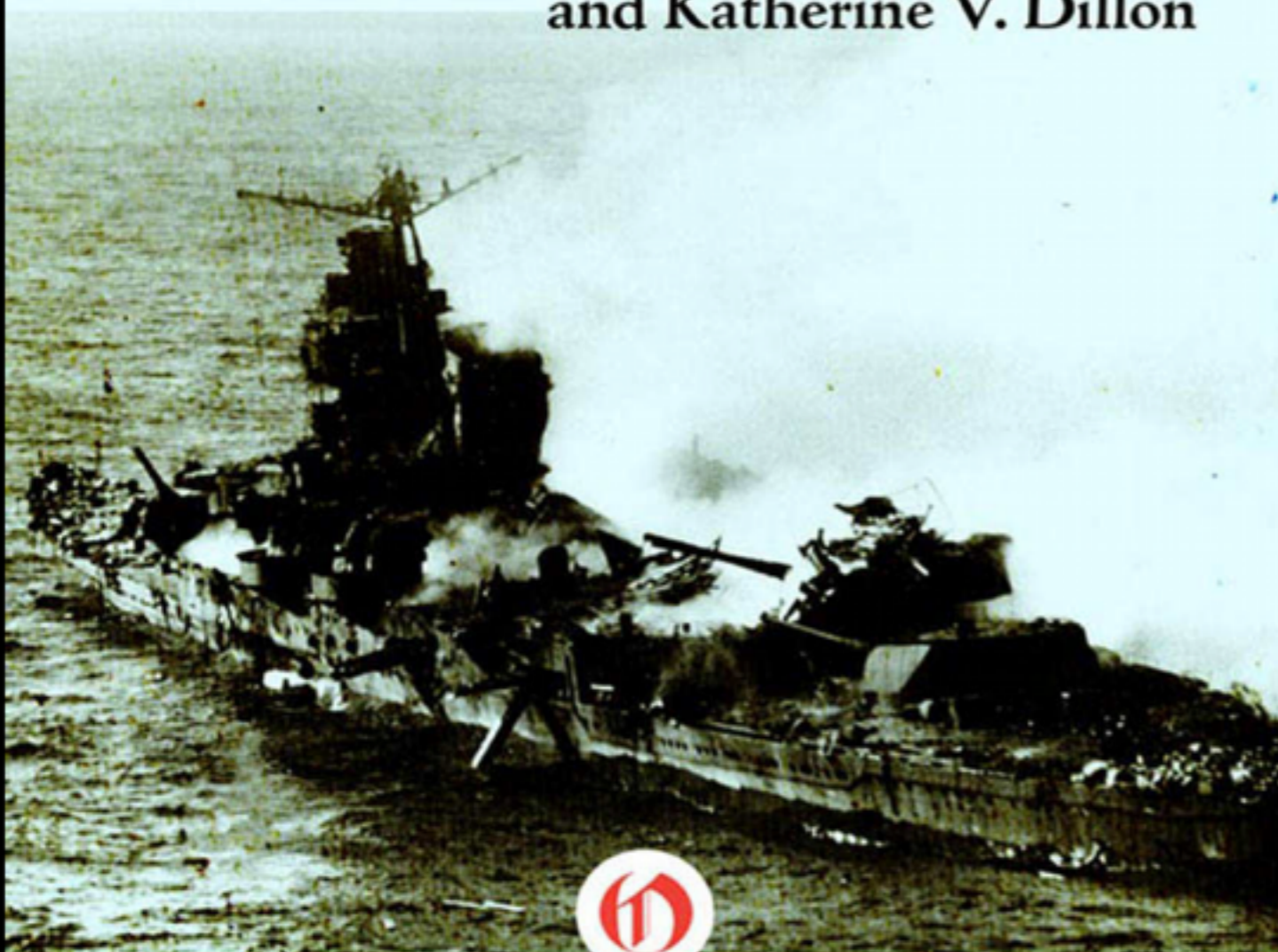
**The Good:**

Lots of facts

Includes some
maps / charts

**The Bad:**

It is a 'wall of text'
when history should be so
much more

# Inspiration

Everything is a remix....

# Visualizing Populations



http://www.fallen.io/ww2/
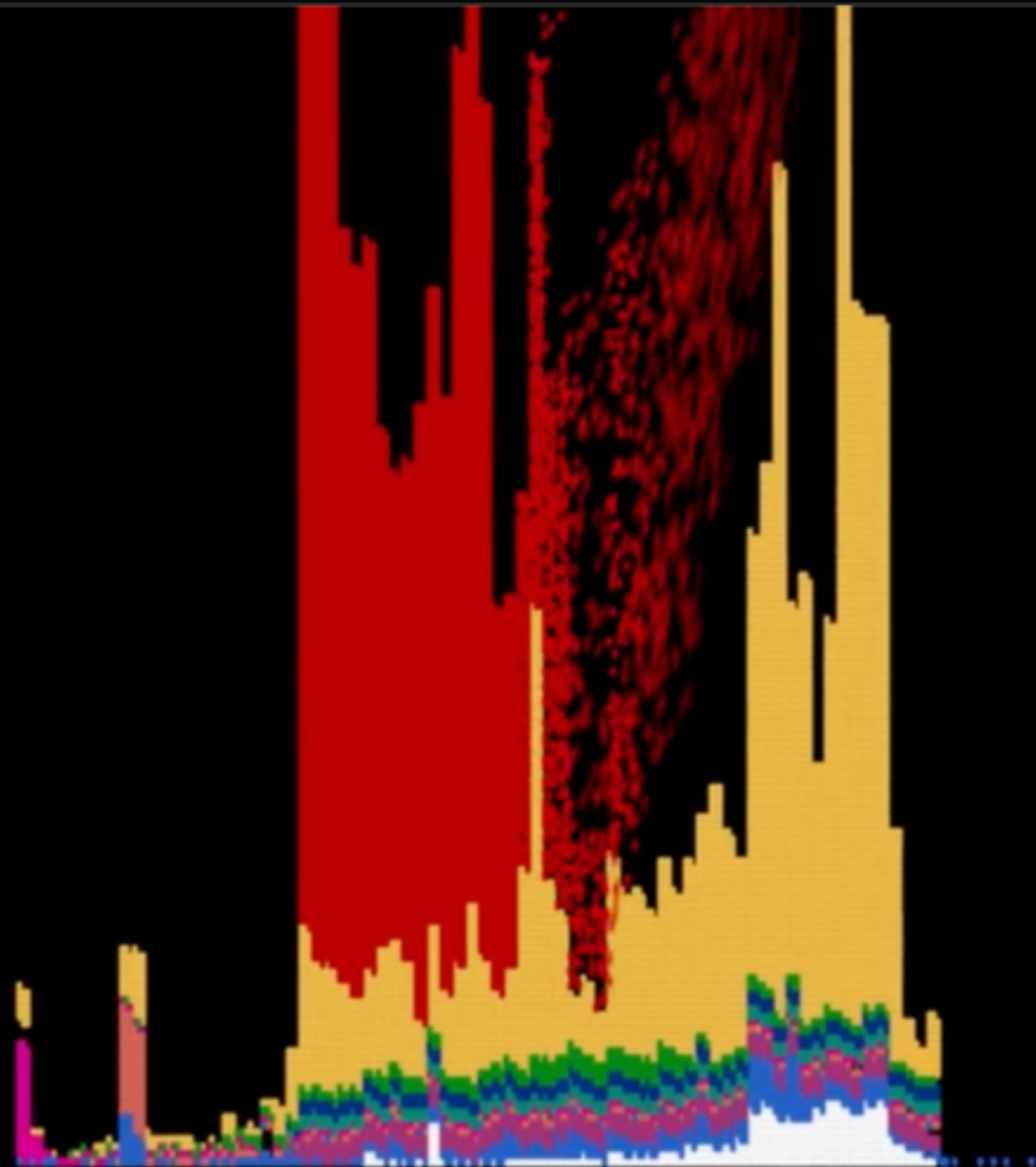
# Escaping Flatland - Visualizing Multiple Dimensions of Data

## Napoleon's March to Moscow
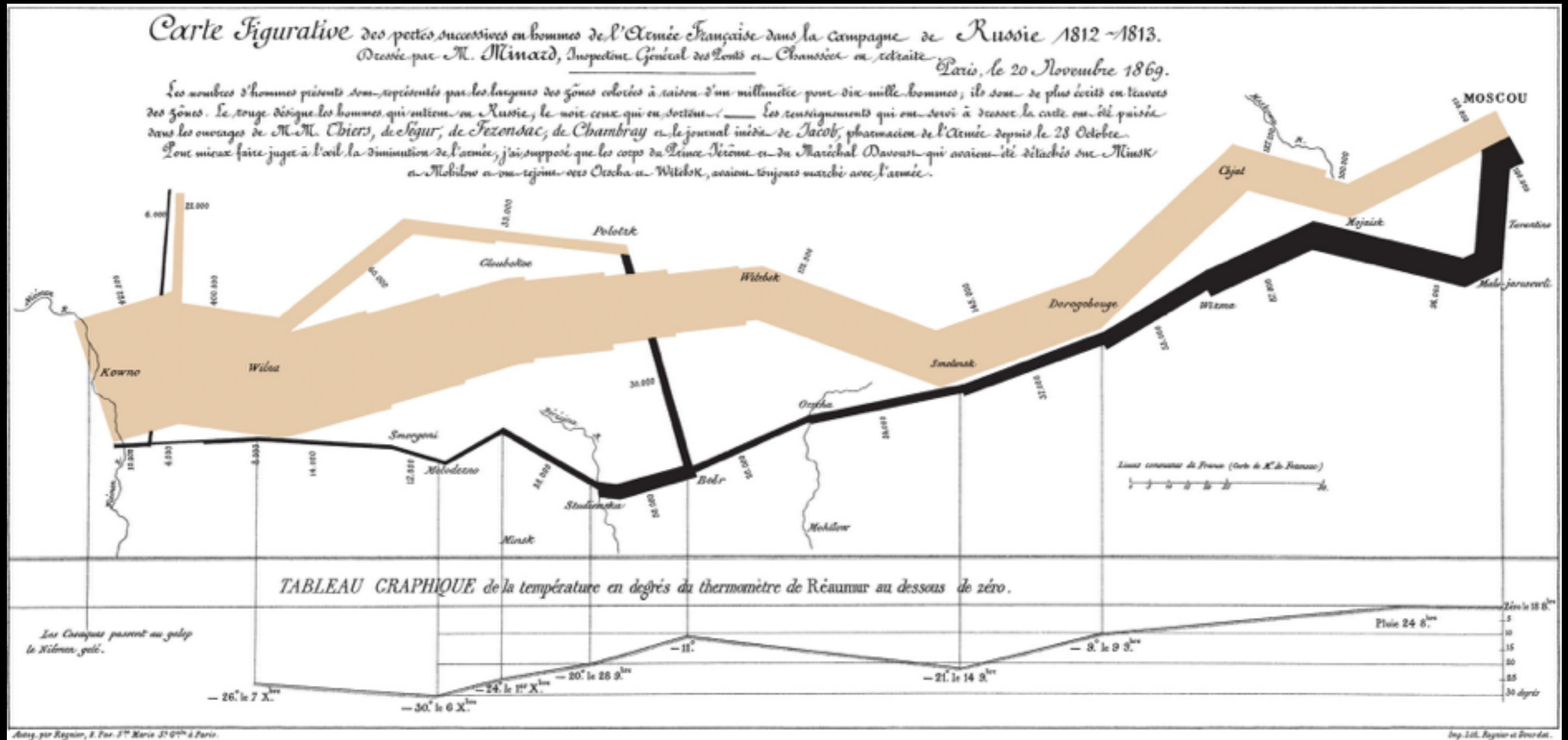


- By Charles Minard (1781-1870) - see upload log, Public Domain, https://commons.wikimedia.org/w/index.php?curid=297925

# Character + Event Timeline



A New Hope

Luke Skywalker
Owen Lars
Beru Lars
Obi-Wan Kenobi
General Tagge
Admiral Motti
Grand Moff Tarkin
Chewbacca
Han Solo
Greedo
Jabba The Hutt
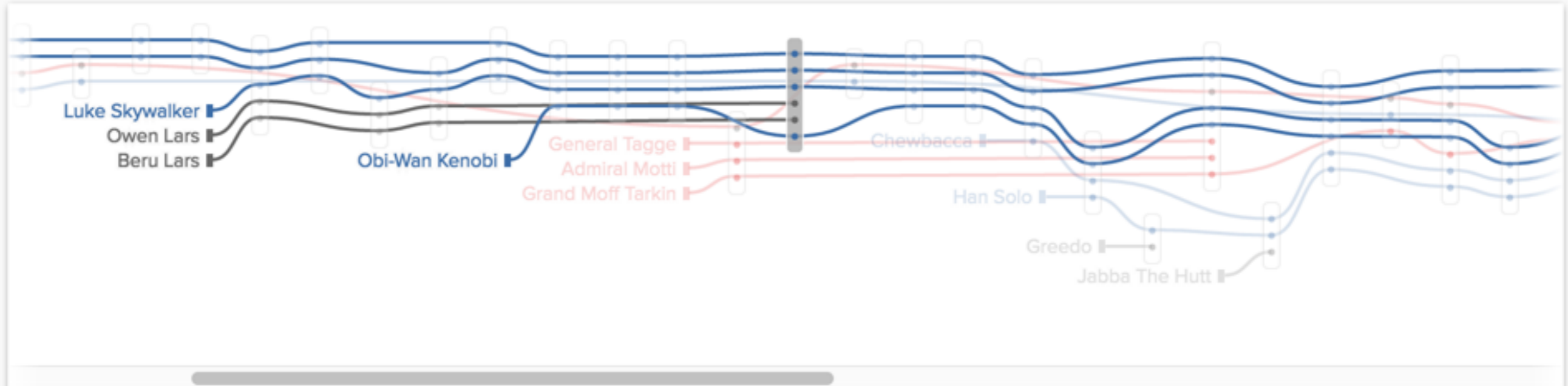
< All junctions > View all

Torched Homestead

Luke Skywalker, Obi-Wan Kenobi, R2-D2 and C-3PO find the Jawa transport destroyed. After realising Stormtroopers were responsible, Luke returns to his home to find it torched and his Uncle Lars and Aunt Meru dead.

http://tinyurl.com/z3ycwx9

# The Tech Tree

Crazy Idea from Miracle at Midway

Find Data Source

DANFS 'Dead Tree' Book

OCR

DANFS

Github - Python Data Pull

SQLite Containing Per Ship DANFS HTML

C#/ .NET

System.Xml.Linq / XDocument

Stanford NER

IKVM - .NET to Java Bridge

Machine Learning
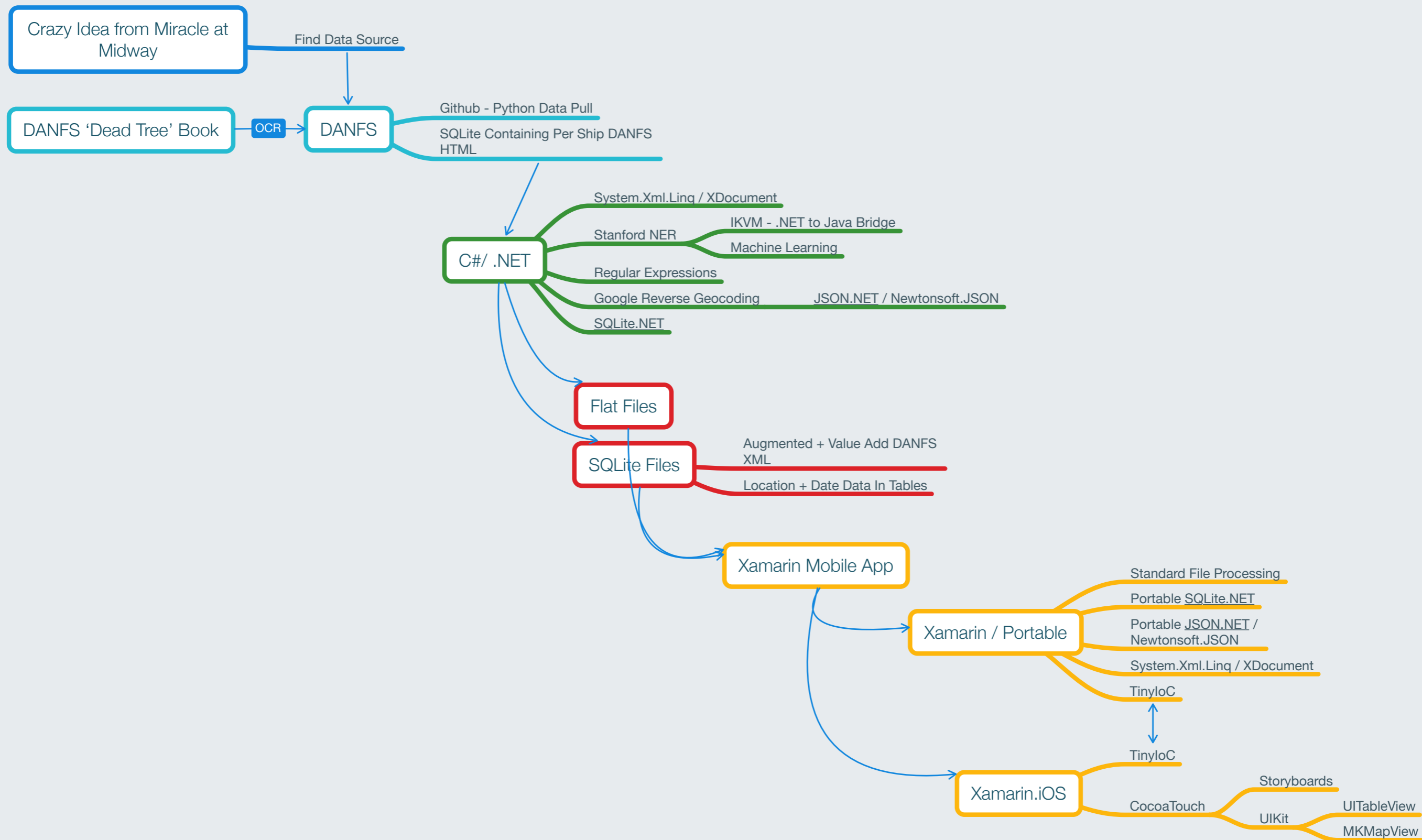
Regular Expressions

Google Reverse Geocoding

JSON.NET / Newtonsoft.JSON

SQLite.NET

Flat Files

SQLite Files

Augmented + Value Add DANFS XML

Location + Date Data In Tables

Xamarin Mobile App

Xamarin / Portable

Standard File Processing

Portable SQLite.NET

Portable JSON.NET / Newtonsoft.JSON

System.Xml.Linq / XDocument

TinyIoC

TinyIoC

Xamarin.iOS

TinyIoC

CocoaTouch

Storyboards

UIKit

UITableView

MKMapView

# We need data!

- Want naval ship logs in chronological order.

- Optimally **Dates / Times** mixed with **Locations**

# DANFS

**D**ictionary of **A**merican **N**aval **F**ighting **S**hips

Over 12,000 American Naval ship histories trapped in HTML generated via Optical Character Recognition.
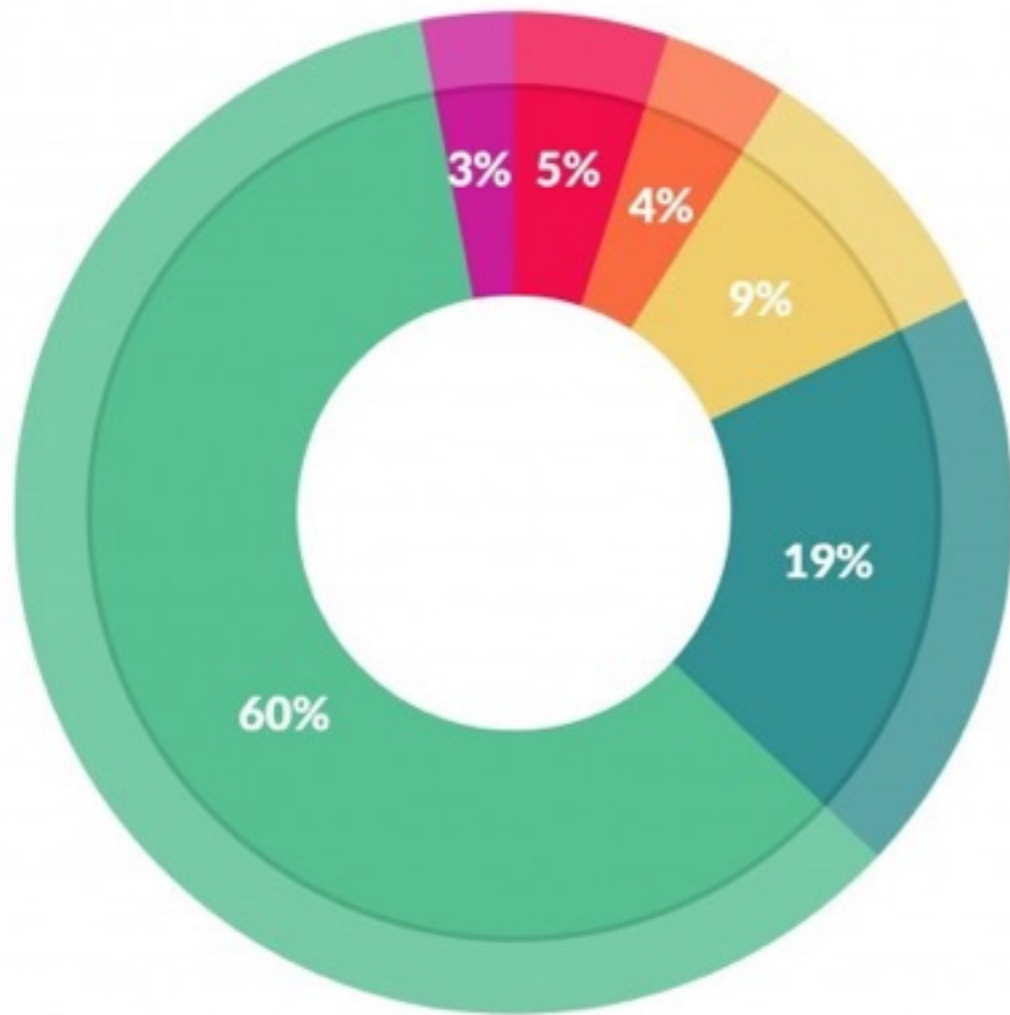
# Thank the Maker for Python and Pythonistas!

https://github.com/jrnold/danfs

https://s3.amazonaws.com/data.jrnold.me/danfs/danfs.sqlite3

# Cleaning the Data



What data scientists spend the most time doing

- Building training sets: 3%
- Cleaning and organizing data: 60%
- Collecting data sets; 19%
- Mining data for patterns: 9%
- Refining algorithms: 4%
- Other: 5%

From: http://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#6bf865c17f75

# General Ship Text Regions

## Cabezon (SS-334)

Ship Metrics

(SS-334: dp. 1,526; l. 311'9"; b. 27'3"; dr. 15'3"; s. 20 k., cpl. 66; a.: 1 5"; 10 21" tt.; cl. *Gato*)

Reason for Name of Ship

A saltwater fish of the sculpin family inhabiting the North Atlantic and North Pacific Oceans.

Ship Log

*Cabezon* (SS-334) was launched 27 August 1944 by Electric Boat Co., Groton, Conn.; sponsored by Mrs. T. R. Cooley; and commissioned 30 December 1944, Commander G. W. Lautrup in command.

*Cabezon* departed New London, Conn., 19 February 1945 for Key West, Fla., where she underwent 3 weeks of training and providing services for the Fleet Sound School. She then sailed via the Panama Canal to Pearl Harbor, arriving 15 March 1945.

From 25 May to 11 July 1945 *Cabezon* conducted her first war patrol in the Sea of Okhotsk, sinking a 2,631-ton Japanese cargo vessel on 19 June. She refitted at Midway until 4 August, then departed for Saipan to serve as target ship for surface force training exercises. From 7 September 1945 until 12 January 1946 she engaged in local operations and training in Philippine waters, based at Subic Bay.

On 6 February 1946 *Cabezon* arrived at San Diego, operating from that port until her base was changed to Pearl Harbor. Subsequent to her arrival there on 20 November 1946, she participated in local operations and training cruises for submariners of the Naval Reserve there and on the west coast with intervening cruises to the South Pacific, the North Pacific, and across the Arctic Circle. She also made two cruises to the Far East (18 March-29 July 1950 and 21 April-16 October 1952), the second of which included a reconnaissance patrol in the vicinity of La Perouse Strait, between Hokkaido, Japan, and Sakhalin, U.S.S.R. She sailed for Mare Island 21 April 1953 to start inactivation and was placed out of commission in reserve there 24 October 1953.

*Cabezon* received one battle star for service in World War II. Her single war patrol was designated "successful." She is credited with having sunk a total of 2,631 tons of shipping.

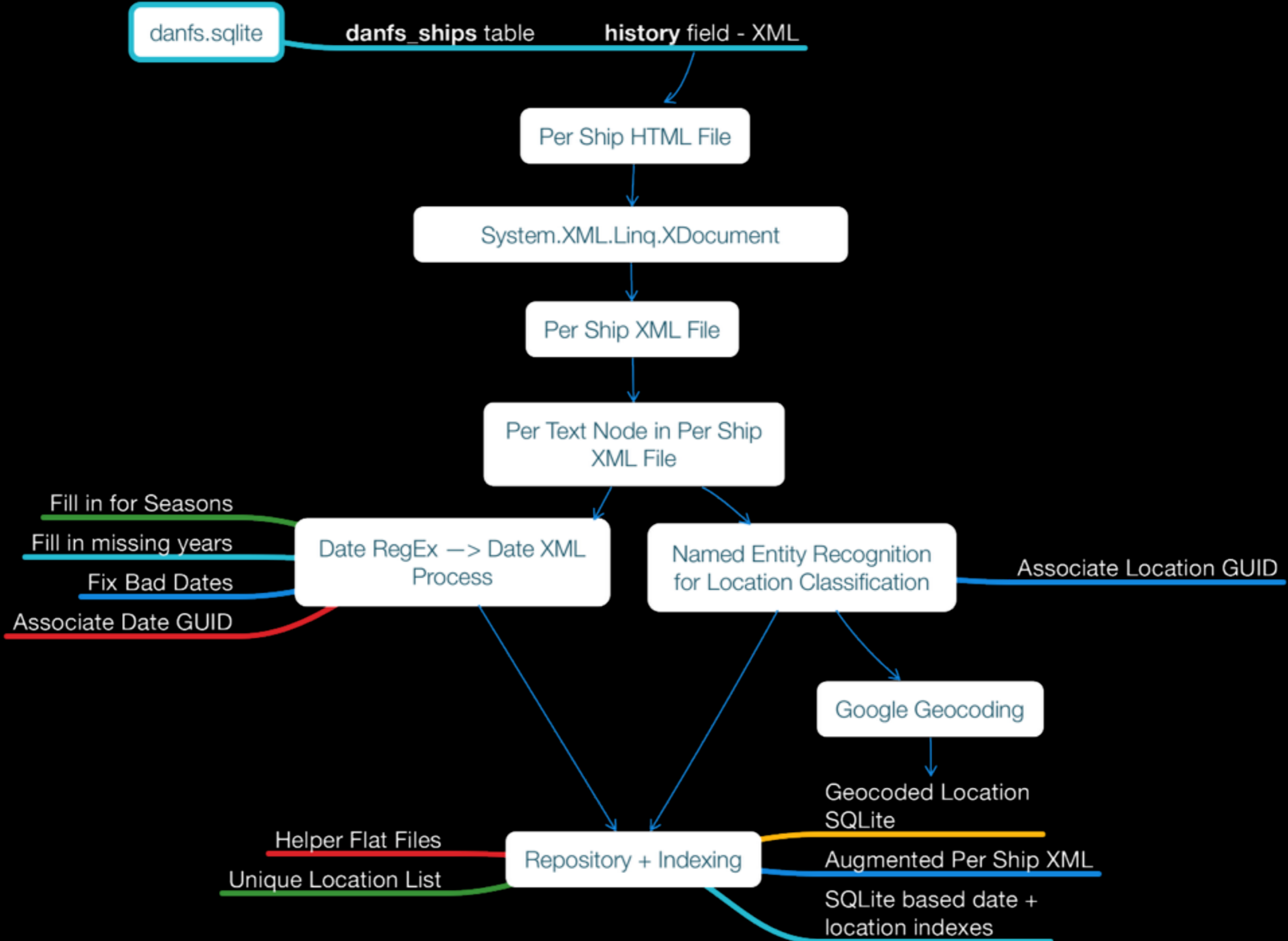Published:Thu Apr 21 03:06:02 EDT 2016

# DANFS Data Sample

The seventh Enterprise (CV-6) was launched **3 October 1936** by Newport News Shipbuilding and Drydock Co., <u>Newport News, Va</u>.; sponsored by Mrs. Claude A. Swanson, wife of the Secretary of the Navy; and commissioned **12 May 1938**, Captain N. H. White in command.
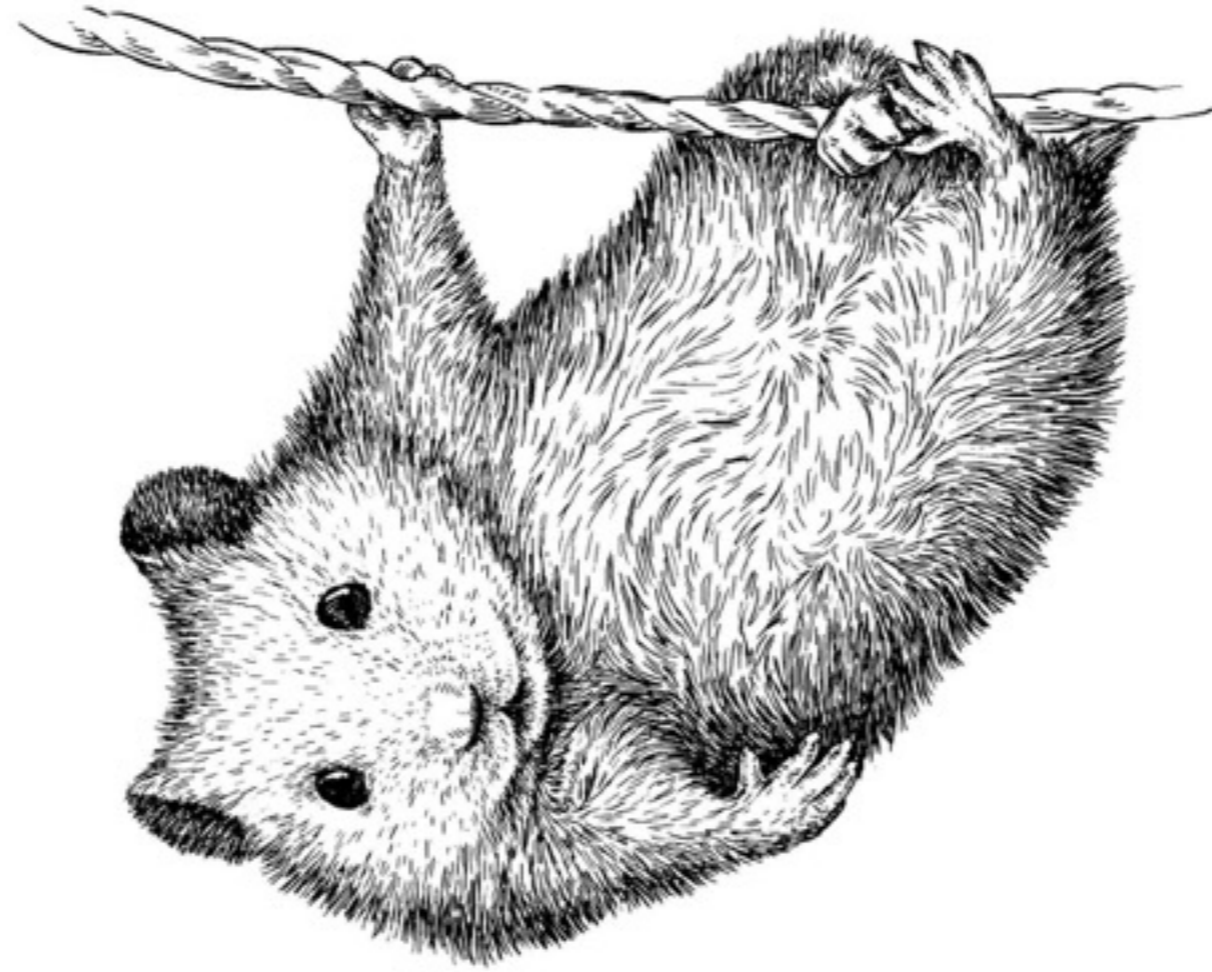
*Enterprise* sailed south on a shakedown cruise which took her to <u>Rio de Janeiro</u>, <u>Brazil.</u> After her return she operated along the east coast and in the <u>Caribbean</u> until **April of 1939** when she was ordered to duty in the Pacific. Based first on <u>San Diego</u> and then on <u>Pearl Harbor</u>, the carrier trained herself and her aircraft squadrons for any eventuality, and carried aircraft among the island bases of the Pacific.

# Cleaning the Data with C# / .NET Framework

danfs.sqlite — **danfs_ships** table — **history** field - XML

Per Ship HTML File

System.XML.Linq.XDocument

Per Ship XML File

Per Text Node in Per Ship XML File

Fill in for Seasons
Fill in missing years
Fix Bad Dates
Associate Date GUID

Date RegEx —> Date XML Process

Named Entity Recognition for Location Classification

Associate Location GUID

Google Geocoding

Geocoded Location SQLite

Helper Flat Files
Unique Location List

Repository + Indexing

Augmented Per Ship XML

SQLite based date + location indexes

The Internet will do the remembering for you

# Googling for the Regex

*Every. Damn. Time.*

O RLY?                                                    @ThePracticalDev

*Combining slashes and dots until a thing happens*

*Expert*

# Regex by Trial and Error

@ThePracticalDev

# DANFS - RegEx for Dates

- Used online **RegEx 101** to test a series of regex options for processing DANFS dates.

# DANFS - Inline XML Dates after Cleaning

```
<date year="1943" month="January" day="8">
8 January 1943
</date>

<date month="August" day="11" year="1944">
11 August
</date>
```

How did we get that year when we only had limited date data?

```
<date month="September" year="1945">
September
</date>
```

# Data Cleaning - Date Fun

Invalid date: <date **month="June" day="31"** year="1972" date_guid="839e81fd-2cec-4856-b3e9-6c95a2e61251">31 June</date> *koelsch* String was not recognized as a valid DateTime.

Invalid date: <date **month="April" day="31"** year="1846" date_guid="943ce8bb-2c1d-48e3-9cf7-568c9f114baf">31 April</date> *lawrence-ii* String was not recognized as a valid DateTime.

Invalid date: <date **month="February" day="29"** year="1863" date_guid="f1948754-c828-4742-80ac-45b06be925bb">29 February</date> *osage-i* String was not recognized as a valid DateTime.

# What next?
# Extract out Locations

- I want a world map so I need locations between all those dates.

- But locations are locked in plain text!

Early in the Civil War, Victoria, a side-wheel steamer built at **Elizabeth, Pa.**, in 1858 and based at **St. Louis**, was acquired by the Confederate Government for service as a troop transport on the waters of the *Mississippi River* and its tributaries. In the spring of 1862, Union warships of the Western Flotilla, commanded at first by Flag Officer Andrew H. Foote and then by Flag Officer Charles H. Davis, relentlessly fought their way downstream from **Cairo, Ill**. On 6 June, they met Southern river forces in the Battle of Memphis and won a decisive victory which gave the North control of the *Mississippi* above **Vicksburg**. Later that day, the Union gunboats found and took possession of several Confederate vessels moored at the wharf at **Memphis**.

Machine Learning To The Rescue!

# Stanford Named Entity Recognizer
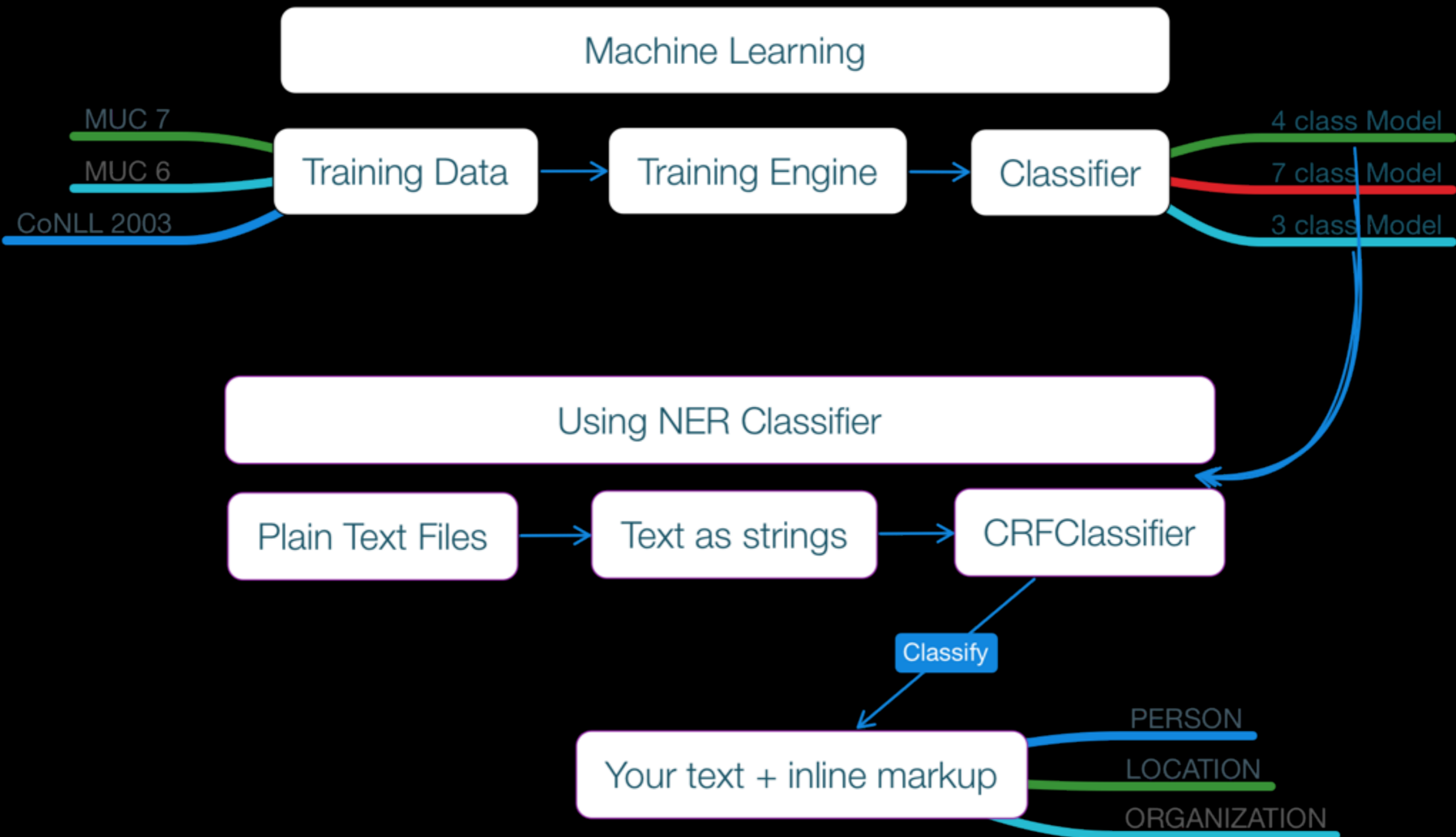
Hey! You said C# / .NET! This is all Java!

Yup, but thanks to IKVM we can use it from .NET via the NuGet Package.

Follow **@sergey_tihon** on Twitter, but beware he is one of *those F# People*!

# Machine Learning in One Slide

- Get **lots and lots** of **known definitions for data**

  - Example: Lots of lists of known world locations.

- Run known definitions for data through a **training engine**.

- Output a **model** that lets a machine take its best guess at finding other stuff that matches your known definitions for data.

# Using Stanford NER

Machine Learning

MUC 7
MUC 6
CoNLL 2003

Training Data → Training Engine → Classifier

4 class Model
7 class Model
3 class Model

Using NER Classifier

Plain Text Files → Text as strings → CRFClassifier

Classify

Your text + inline markup

PERSON
LOCATION
ORGANIZATION

# Stanford NER - Easy Inline XML Classification

```
//Default: Easy convenience method from Stanford NER.

  var classifierResult =
  classifier.classifyWithInlineXML(textValue);
```

# Stanford NER - Inline XML Default Output

```
<i>Abele</i>left
<LOCATION>
Pearl Harbor
</LOCATION>,
bound for Iwo Jima. After sailing via
<ORGANIZATION>
Eniwetok
</ORGANIZATION> and
<LOCATION>
Guam
</LOCATION> with Task Group 51.5, the ship arrived off
<LOCATION>
Iwo Jima
</LOCATION> on
<date month="February" day="20" year="1945">20 February </date>and began laying a
torpedo net.
She remained in the area for eight days laying nets and fleet moorings before
getting underway
on the 28th and heading for
<LOCATION>
Saipan
</LOCATION>
```

# Uh, Oh! Classification Problems

`<ORGANIZATION>`
`Eniwetok`
`</ORGANIZATION>`

**Eniwetok** is a *location* not an organization

# Stanford NER - Adding Probability to Output

```csharp
//Custom: Complete deconstruction and C# re-implementation
of
//classifyWithInlineXML convenience method.
var sentences = classifier.classify(textValue);

var sb = new StringBuilder();
for (var itr = sentences.iterator(); itr.hasNext();)
{
    var sentence = itr.next() as java.util.List;

    var cliqueTree =
classifier.getCliqueTree(sentence);

    //Special custom method that custom implements inline XML
    //to merge probabilities into the XML output.
    printAnswersInlineXML(sentence , sb, cliqueTree);
}

var classifierResult = sb.ToString();
```

# Stanford NER - Probability Added to Output

```
<i>Abele</i>left
<LOCATION PROBABILITY="0.998654781217597">
Pearl Harbor
</LOCATION>,
bound for Iwo Jima. After sailing via
<ORGANIZATION PROBABILITY="0.434755597252728">
Eniwetok
</ORGANIZATION> and
<LOCATION PROBABILITY="0.939351891155514">
Guam
</LOCATION> with Task Group 51.5, the ship arrived off
<LOCATION PROBABILITY="0.534756216597627">
Iwo Jima
</LOCATION> on
<date month="February" day="20" year="1945">20 February </date>and began laying a torpedo
net.
She remained in the area for eight days laying nets and fleet moorings before getting
underway
on the 28th and heading for
<LOCATION PROBABILITY="0.989145268947393">
Saipan
</LOCATION>
```
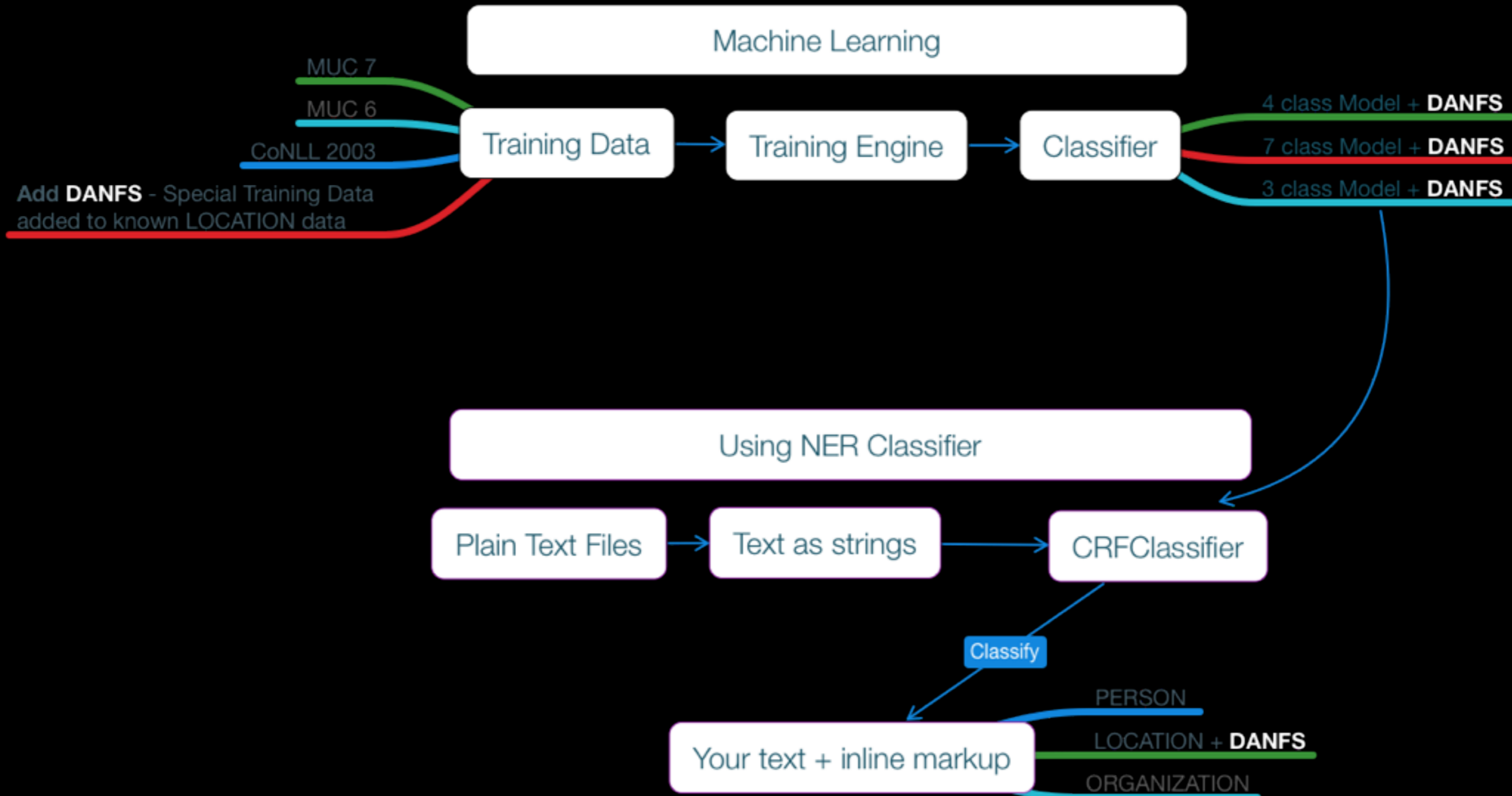
# Stanford NER - Future - Improving Classification for DANFS via Training

# Other Machine Learning Resources

- <u>Google Prediction API</u>

- Stanford Natural Language Processing

  - The base for Stanford NER

- <u>Machine Learning In a Year</u>

  - <u>Machine Learning In a Week</u>

# Google Geocoding

http://maps.googleapis.com/maps/api/
geocode/json?
address=Eniwetok&sensor=false

Can only geocode about 150 locations per day per IP address.

We have 20,000 unique locations

# When Geocoding try not to end up on Null Island



0°,0°

NULL ISLAND

# Google Geocoding JSON Return
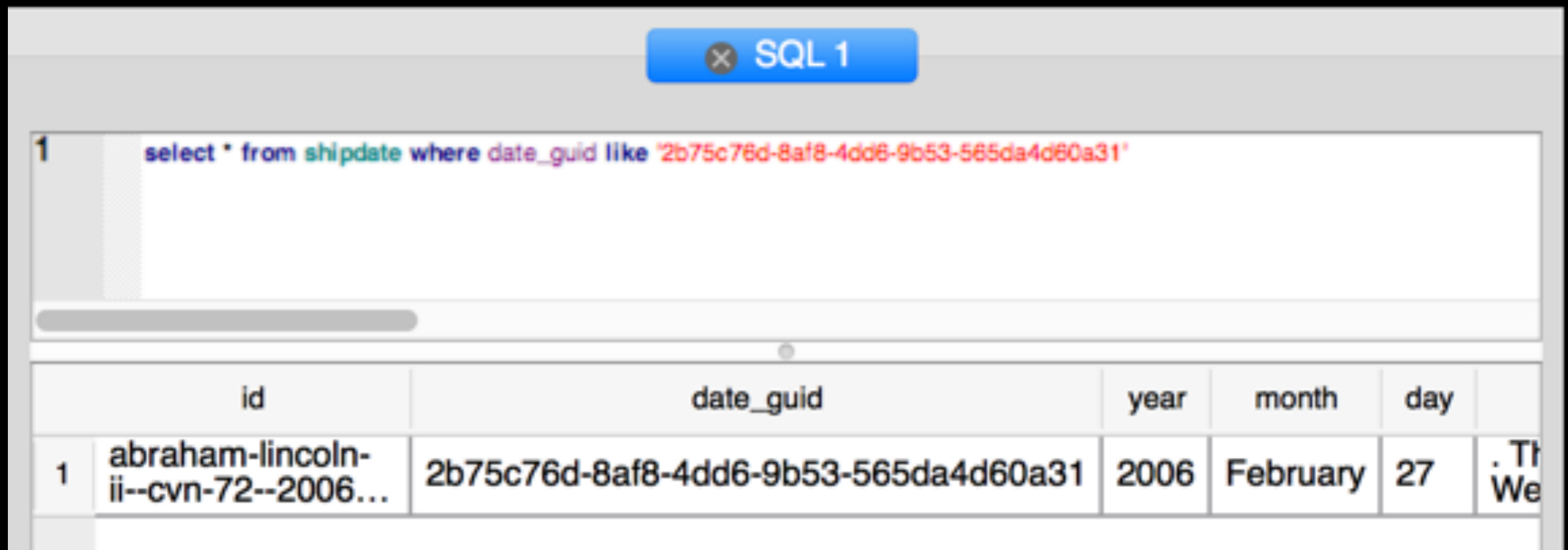
```json
{
  "results" : [
    {
      "geometry" : {
        "bounds" : {
          "northeast" : {
            "lat" : 11.3603022,
            "lng" : 162.3477857
          },
          "southwest" : {
            "lat" : 11.3357145,
            "lng" : 162.3176258
          }
        },

        "location" : {
          "lat" : 11.3415658,
          "lng" : 162.3266731
        },
        "location_type" : "APPROXIMATE",
        "viewport" : {
          "northeast" : {
            "lat" : 11.3603022,
            "lng" : 162.3477857
          },
          "southwest" : {
            "lat" : 11.3357145,
            "lng" : 162.3176258
```

# Document to Database Linkage via GUID

```
<date year="2006" month="February" day="27"
date_guid="2b75c76d-8af8-4dd6-9b53-565da4d
60a31">27 February 2006</date>
```

http://www.howtogeek.com/wp-content/uploads/gg/up/
sshot4f0de139724c3.jpg

If your data cleaning code doesn't resemble
the above, you may have done it wrong.

We have hit a data wall
and are at a crossroads

# Data cleanliness probably not so good

- We have 11,000 ship story XML files that contain:

  - Dates categorized at about 90% (that's a guess)

  - Locations categorized at about 70% (that's a guess)

  - Date + Location associations at about 50-60% (if that)

  - Google Geocode results do return multiple values, but we have enough to 'just pick one'.

# Eyeballing the Date / Location Data

```
<date month="February" day="20" year="1945">20 February </date>
   and began laying a torpedo net. She remained in the area for eight days laying ne
   and
   fleet moorings before getting underway on the 28th and heading for
 <LOCATION >Saipan</LOCATION>
  to prepare for the upcoming
 <LOCATION >Okinawa</LOCATION>
  invasion.</p> <p>After a brief period spent in the
 <LOCATION>Leyte Gulf</LOCATION>
      staging area, Abele arrived off
 <LOCATION>Kerama Retto</LOCATION> on
<date month="March" day="26" year="1945">26 March </date>
   to begin laying net defenses. Although she was attacked by Japanese suicide boats
and
   aircraft during the next seven weeks, she suffered no damage. On
<date month="April" day="18" year="1945">18 April</date>,
   the ship assisted in the downing of one enemy airplane. On
<date month="May" day="12" year="1945">12 May</date>
   , she sailed to Nagagusuku Wan,
 <LOCATION>Okinawa</LOCATION>,
   and assisted in laying five miles of heavy antitorpedo nets across the
   harbor entrance. She also claimed credit for downing one Japanese "Val" on
<date month="June" day="11" year="1945">11 June</date>.
```

# Where to go next? Tool up and Visualize with iOS app

- Let's keep moving forward even though our data ain't that good.

- Let's get started on visualizing the ship + date + location data even if it isn't accurate.

- Visualizing will help us determine our success rate to our final goal.

- Visualizing will lead to possibly better overall tooling and help in determining next steps.

# Check Performance

- 230,805 dates recorded in SQLite and marked up in XML

- ~20,000 locations recorded in SQLite and marked up in XML

- 299,021 date + location associations recorded into SQLite.

No backing service… Can we just run it on device?

# Full DANFS iOS App Demo

# Why Xamarin.iOS and not Obj-C / Swift?

- LINQ, LINQ, LINQ

  - Did I forget to mention LINQ?

- System.Xml.Linq

- SQLite.NET + LINQ

- Newtonsoft.JSON / JSON.NET

- Portable Libraries - Android and Windows Future

# Why Swift / Obj-C and not Xamarin.iOS?

- External Code Dependencies

  - CocoaPods

  - Carthage

  - Third Party Frameworks like Google VR SDK may not be wrapped or wrappable

- Newer APIs may not be wrapped or may be buggy

- Startup Performance - But is less of a factor now

# Storyboard Centric App Structure

- Main Storyboard

  - Stock + Custom View Controllers

- Segues to bind everything together
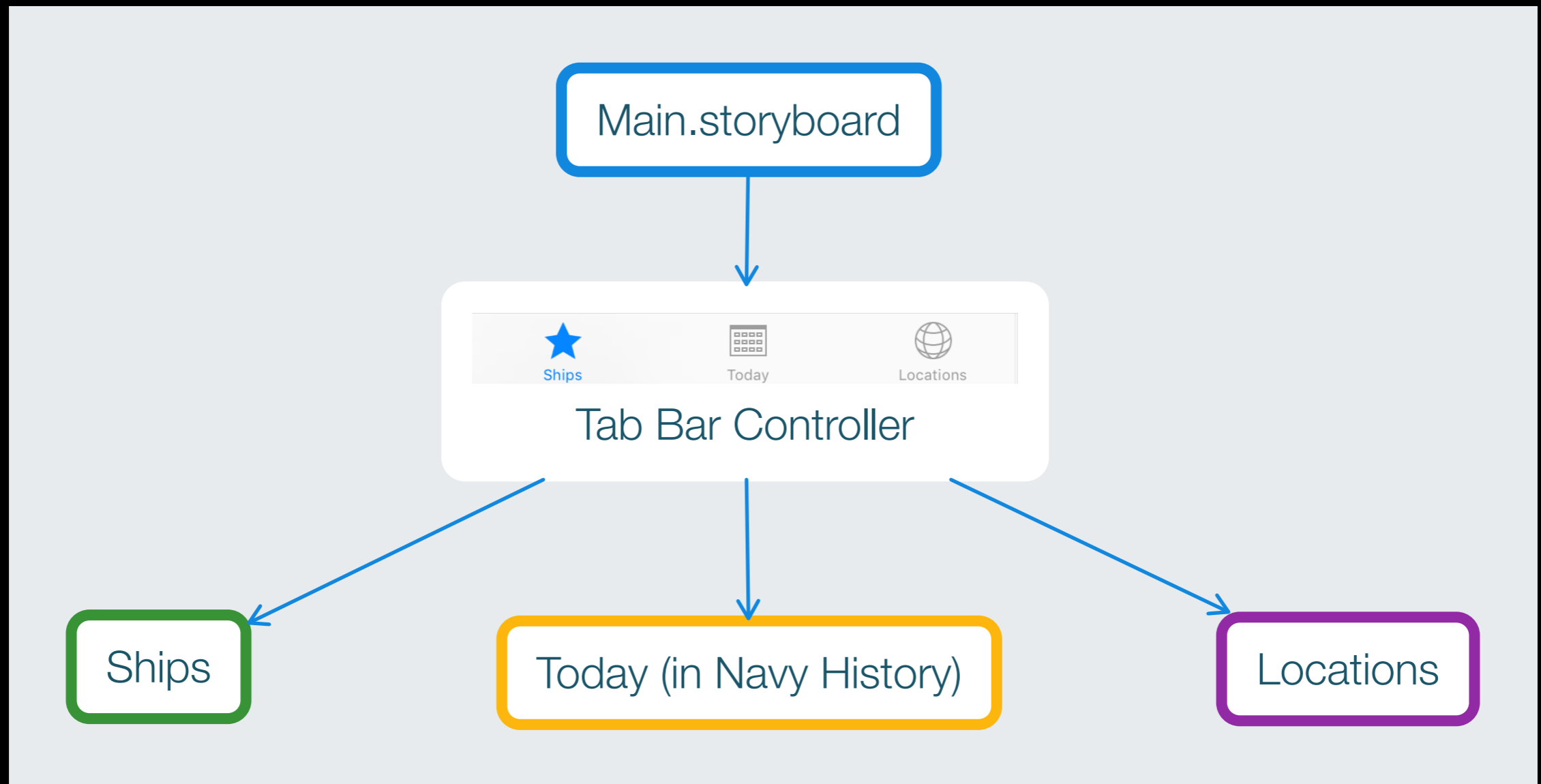
  - Show

  - Embed

  - Modal

  - Exit

# **Ctrl + Drag + Drop**
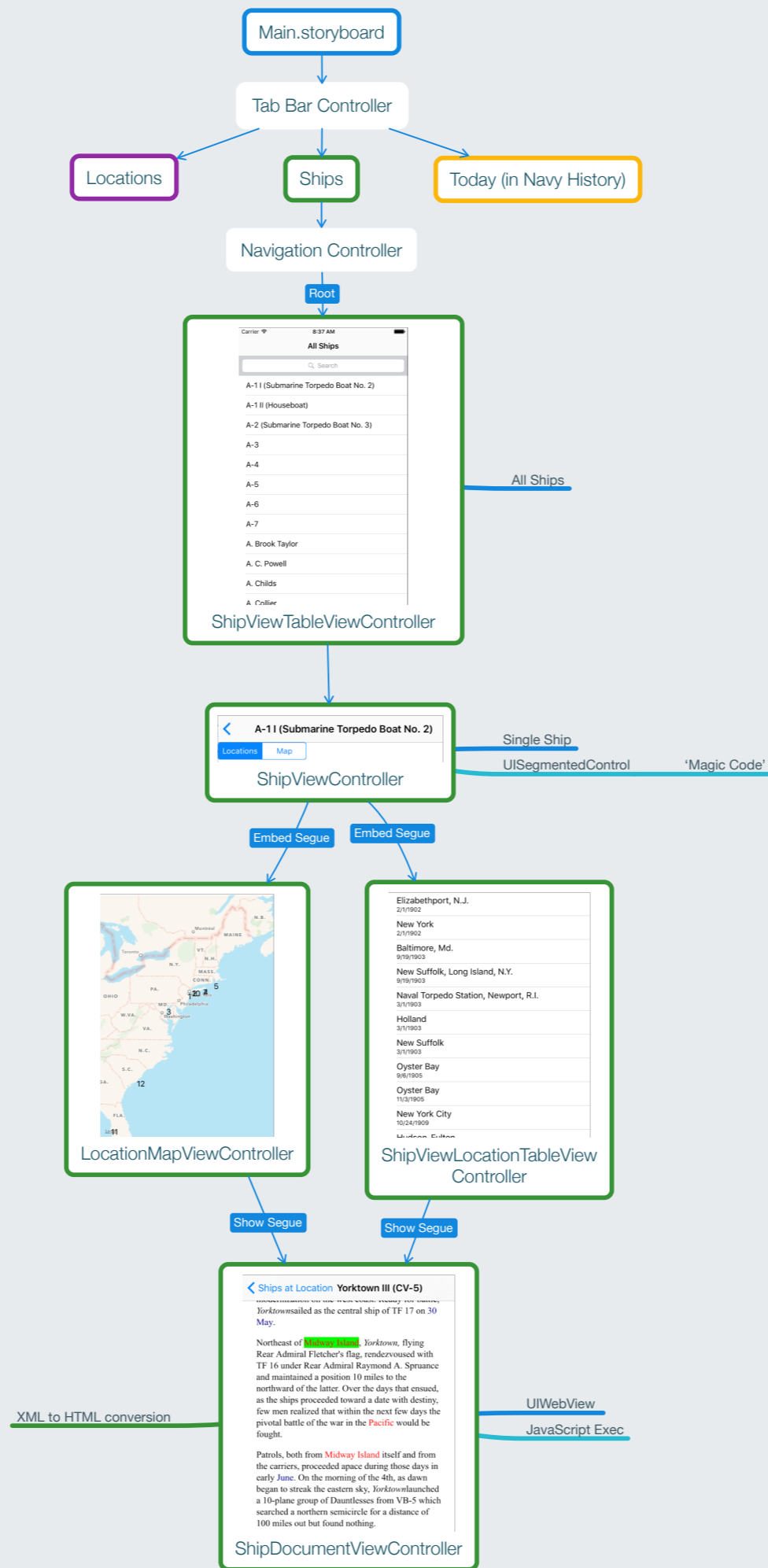## One Xcode Gesture To Rule Them All!

- UI Element to Storyboard Segues for app navigation without code

- Auto Layout without code

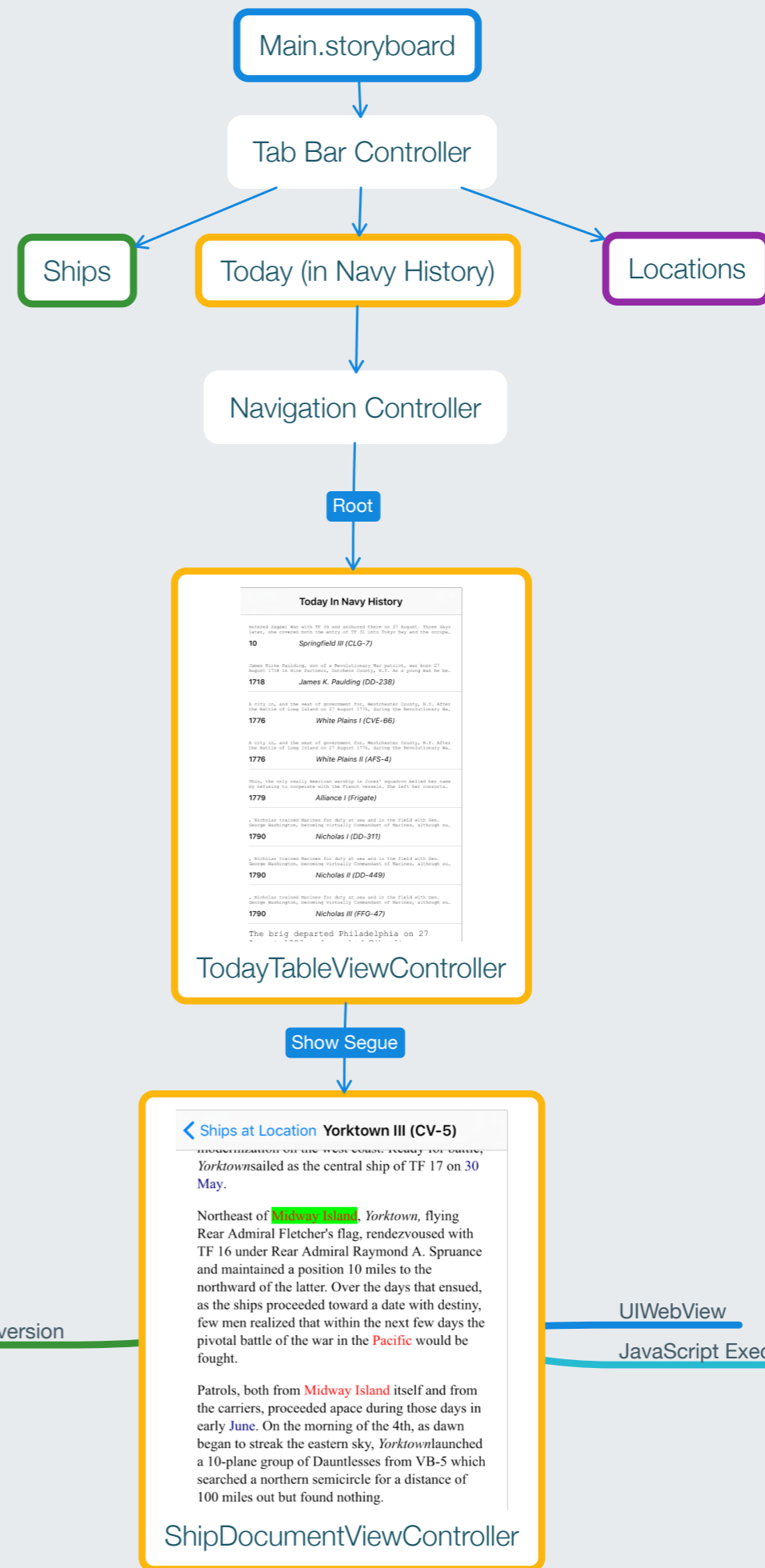- Associate view to code behind

- Associate events to code behind

# Storyboard Root

Main.storyboard

Tab Bar Controller

★ Ships   Today   Locations

Ships

Today (in Navy History)

Locations

# Ships

Main.storyboard

Tab Bar Controller

Locations | Ships | Today (in Navy History)

Navigation Controller

Root



All Ships

ShipViewTableViewController

Single Ship

ShipViewController

UISegmentedControl          'Magic Code'

Embed Segue          Embed Segue



LocationMapViewController



ShipViewLocationTableView Controller
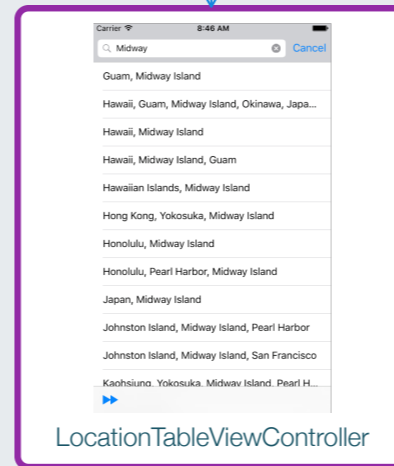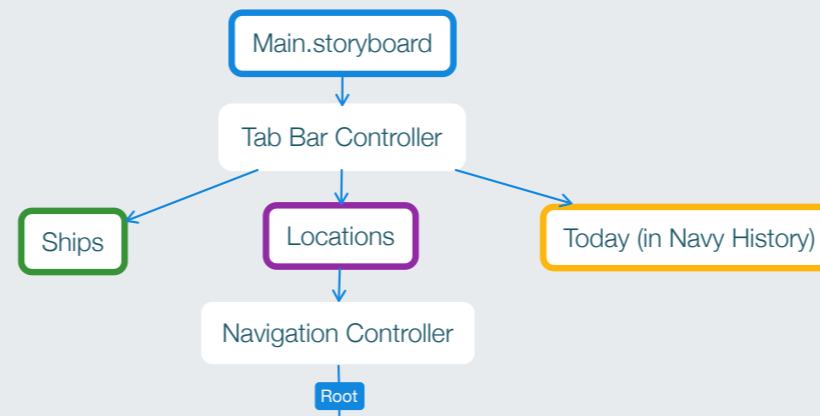
Show Segue          Show Segue



XML to HTML conversion          UIWebView          JavaScript Exec

ShipDocumentViewController

# Today In Navy History

Main.storyboard

Tab Bar Controller

Ships

Today (in Navy History)

Locations

Navigation Controller

Root



TodayTableViewController

Show Segue



ShipDocumentViewController

XML to HTML conversion

UIWebView

JavaScript Exec

# Locations

Main.storyboard

Tab Bar Controller

Ships

Locations

Today (in Navy History)

Navigation Controller

Root



LocationTableViewController

Show Segue



LocationShipTableViewController
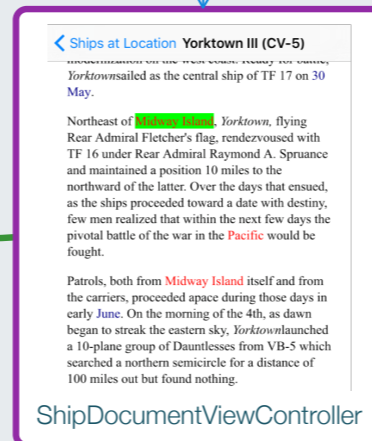
Modal Segue
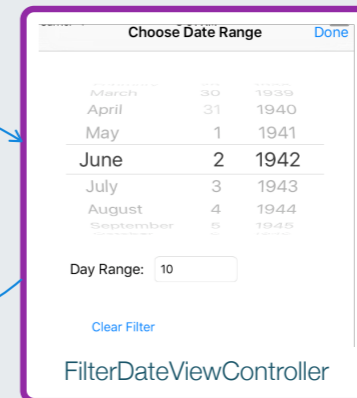
Exit Segue



FilterDateViewController

Show Segue

XML to HTML conversion

UIWebView

JavaScript Exec



ShipDocumentViewController

# Using HTML and UIWebView

- Don't over complicate XML to HTML generation.

```
private static void ProcessShipElement(XElement element, StringBuilder sb)
{
    if (element.Name == "date")
    {
        //Just insert a date marker.
        sb.Append("<span id=\"date-");
        sb.Append(element.Attribute("date_guid").Value);
        sb.Append("\" style=\"color:navy\">");
    }
```

```
    foreach (var childNode in element.Nodes())
    {
        if (childNode is XText)
        {
            sb.Append((childNode as XText).Value);
        }
        else if (childNode is XElement)
        {
            ProcessShipElement((childNode as XElement), sb);
        }
    }
```

```
    if (element.Name == "date" || element.Name == "LOCATION")
    {
        sb.Append("</span>");
    }
```

# Using HTML and UIWebView

- Just put in raw HTML with `UIWebView` `LoadHtmlString`

```
var htmlString = await dataAccess.GetDisplayableHTMLForShip(shipIDToShow);

ActiveWebView.LoadHtmlString(htmlString, null);
```
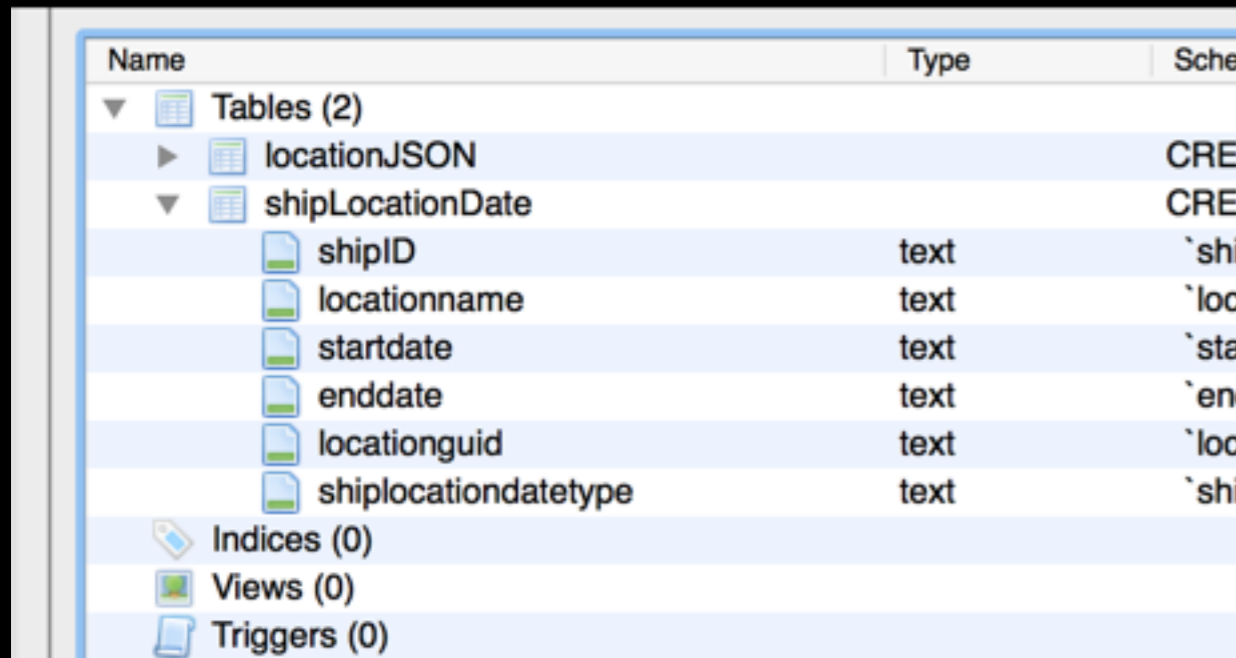
# Using HTML and UIWebView

- Invoke inline JavaScript in `LoadingFinished` off of `UIWebViewDelegate`

```
[Export("webViewDidFinishLoad:")]
public void LoadingFinished(UIWebView webView)
{
    if (ShipEvent != null)
    {
        ActiveWebView.EvaluateJavascript($"document.getElementById('date-{ShipEvent.date_guid}').scrollIntoView()");
        ActiveWebView.EvaluateJavascript($"document.getElementById('date-{ShipEvent.date_guid}').style.backgroundColor = \"#00FF00\"");
    }
    if (LocationGuidToHighlight != null)
    {
        ActiveWebView.EvaluateJavascript($"document.getElementById('location-{LocationGuidToHighlight}').scrollIntoView()");
        ActiveWebView.EvaluateJavascript($"document.getElementById('location-{LocationGuidToHighlight}').style.backgroundColor = \"#00FF00\"");
    }
}
```

# Portable SQLite + LINQ

## From DB Browser for SQLite

| Name | Type | Sche |
|---|---|---|
| ▼ 📊 Tables (2) | | |
| ▶ 📋 locationJSON | | CREA |
| ▼ 📋 shipLocationDate | | CREA |
|   📄 shipID | text | `ship |
|   📄 locationname | text | `loc |
|   📄 startdate | text | `sta |
|   📄 enddate | text | `end |
|   📄 locationguid | text | `loc |
|   📄 shiplocationdatetype | text | `ship |
| 🏷️ Indices (0) | | |
| 🖼️ Views (0) | | |
| 📜 Triggers (0) | | |

## C# code

```csharp
public class shipLocationDate
{
    public string shipID { get; set; }
    public string locationname { get; set; }
    public string startdate { get; set; }
    public string enddate { get; set; }
    public string locationguid { get; set; }
    public string shiplocationdatetype { get; set; }
}
```

```csharp
public List<shipLocationDate> GetShipListByLocation(string locationName)
{
    var connection = new SQLite.Net.SQLiteConnection(
        TinyIoC.TinyIoCContainer.Current.Resolve<ISQLitePlatform>(),
        TinyIoC.TinyIoCContainer.Current.Resolve<IFolderProvider>().MapDatabasePath);

    var query = connection.Table<shipLocationDate>().Where(r => r.locationname == locationName);

    var result = query.ToList();

    return result.OrderBy(r => r.OrderedDate).ToList();
}
```

# Don't underestimate the power of NSAttributedString and UILabel

```
private NSAttributedString GetShipEventDetailText(shipdate shipEvent)
{
    NSMutableAttributedString detailString = new NSMutableAttributedString();

    detailString.Append(new NSAttributedString(shipEvent.year, new UIStringAttributes
    {
        Font = UIFont.BoldSystemFontOfSize(12)
    }));

    detailString.Append(new NSAttributedString("\t\t\t"));

    detailString.Append(new NSAttributedString(shipEvent.title, new UIStringAttributes
    {
        Font = UIFont.ItalicSystemFontOfSize(12)
    }));

    return detailString;
}
```

A city in, and the seat of government for, Westchester County, N.Y. After
the Battle of Long Island on 27 August 1776, during the Revolutionary Wa…

1776                          White Plains II (AFS-4)

\* Be Lazy! Don't make custom UITableViewCell(s) if you don't have to!

# Call To Action

- Find your data set

- Find your own motivation to do something with that dataset

- The tools are amazing and free!

- You can build on the shoulders of giants.

# Future Stuff

- People - Name + Rank + Ship Mentions + Location associations.

- Ship to Ship Encounters at given locations on certain dates

- On-map routes, begin / end date routing, multiple ships per map, ….

- Ship Data cracking and correlation + lookup and association by class, type, width, beam, …..

- Auto generate Wikipedia Links to content.
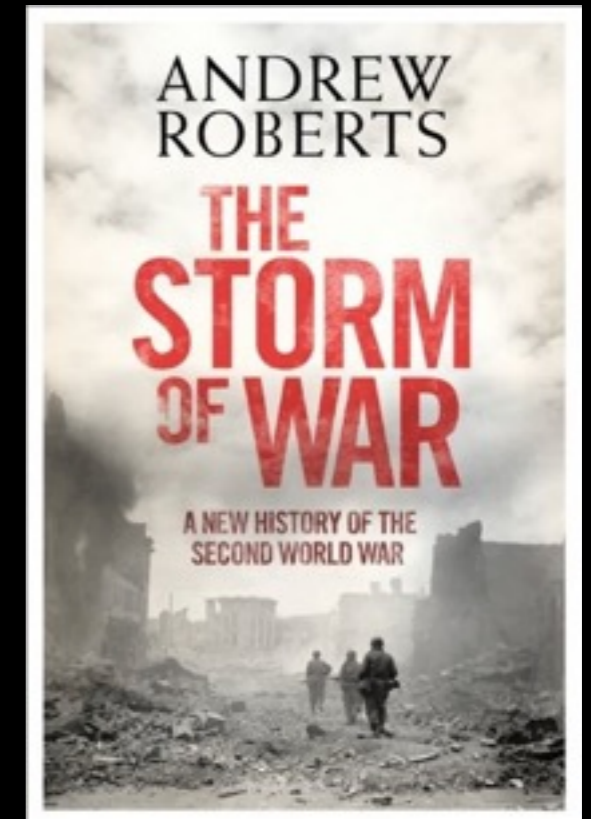
- …. and even more!

# Other World War II Perspectives
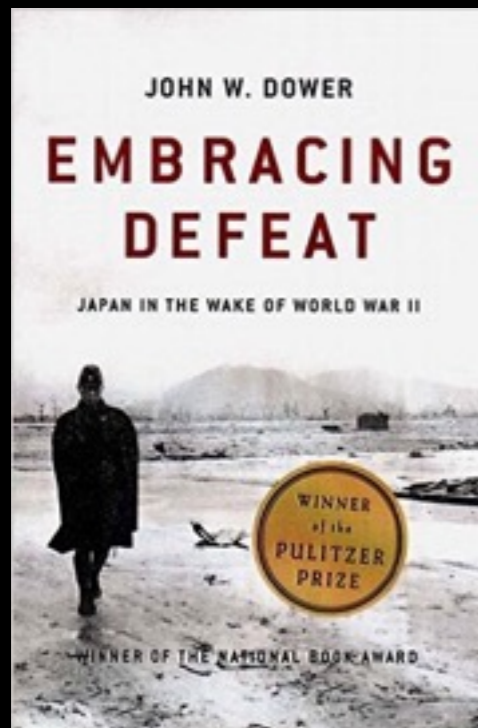


Post WWII Japan
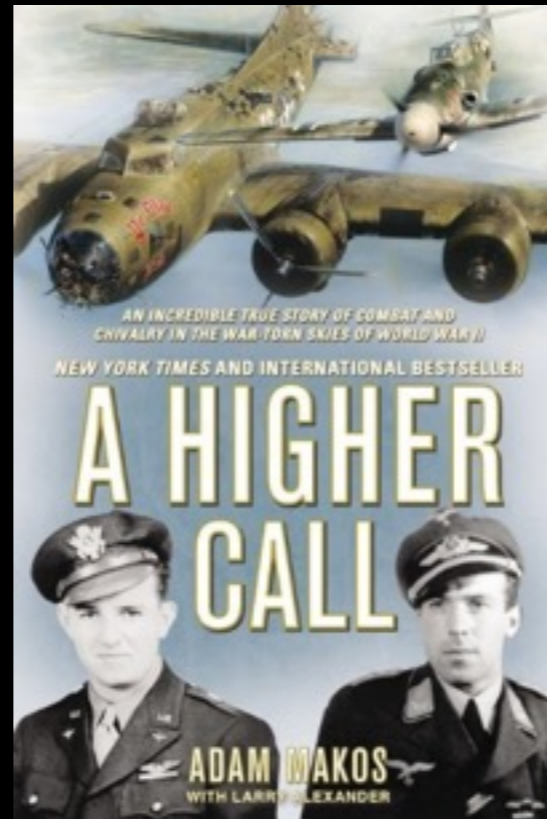


German Pilot Perspective of North Africa and Europe



Eastern Front Focused

# Tools Used In This Presentation

- MindNode (iOS and Mac App Stores) - For all diagramming

- SnagIt - For all screen shots

- Highlight - For all in-text XML highlighting - *maroloccio* theme — See also this great guide on how to use it.