SOTU-db

State of the Union

database

SOTU-db is a digital tool that provides a simple web interface for searching and conducting sentiment analysis on the texts of annual US Presidential Addresses (“State of the Unions). It is due to be released publicly on December 6 at:

www.sotu-db.com

Created by Tyler Monaghan

*Tyler is a second-year student in LUC’s Digital Humanities MA program.*

portfolio.tylermonaghan.com

“SOTUs:” A Text Corpus

(or three)

* **Three sources for text of addresses:**
  + Project Gutenberg (1790-2006, with gaps)
  + UCSB “American Presidency Project”
  + C-Span (1945-2006)
* **Transcriptions are not neutral:**
  + Many past annual messages delivered as written report, not spoken address
  + Should interruptions (such as applause) be included and noted?
* **Texts must be “cleaned” and prepared for textual analysis**
  + for example, converting all text to lowercase, removing special characters ($)
  + often, “stopwords” (common words like articles, pronouns) are ignored in textual analysis, but can be important here (e.g. how presidents use “us” and “them”)

“www.sotu-db.com”

The User Experience

* **Inspired by** other digital tools like voyant-tools
* **Designed for** a range of users and use-cases:
  + students conducting general research
  + scholars conducting specific analyses
  + journalists/bloggers looking for a take or visualization
* As **simple to use** as a Google search, with more powerful features for advanced users
* Encourages playful **exploration and experimentation**
* Results must be exactly **reproducible and sharable**

The “LAMP” Stack:

Digital Infrastructure

* **L**inux virtual machine to run everything
* **A**pache web server to accept user requests and serve pages
* **M**ySQL database to hold the text corpus
* **P**HP to dynamically generate content in response to user searches
* Together, these tools enable complete control of the whole user experience

*If you think you’re “not a tech person,” take heart: before last September I had never written a single line of code!*

Text Mining and Analytics

* Two primary tools:
  + R statistical computing & graphics language
  + Natural Language Toolkit (NLTK) Python library
* Take user input (search term)
* Process text (transform, filter, sort, etc.)
* Provide output and serve to user as web page

