Enhancing the TAG Project

Drew Schmidt

October 21, 2015

The TAG Project

What is TAG?

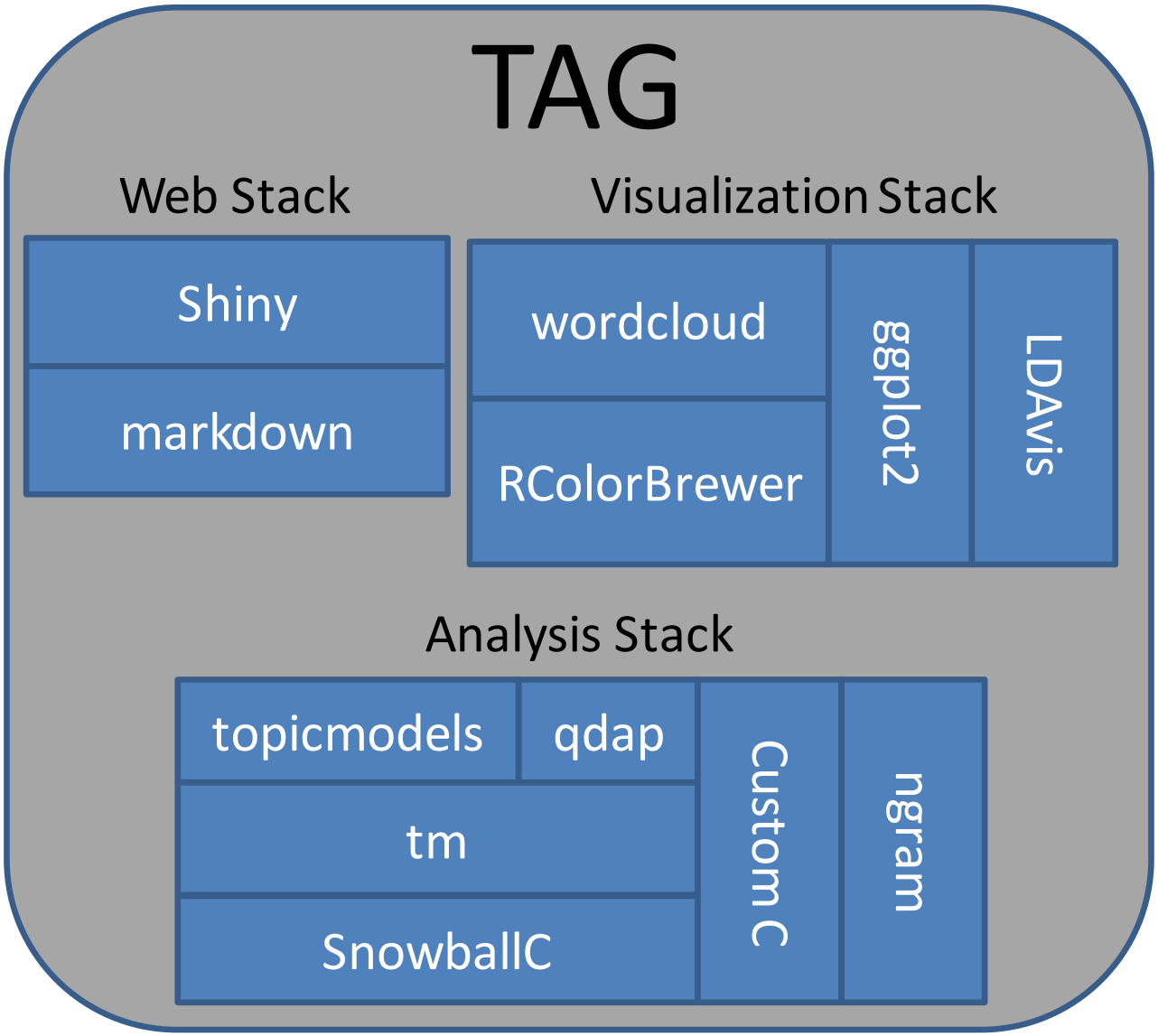
- Web service/GUI.
- Basic text processing and mining.
- Serving HASS comunities.
- Designed for portability: laptop, supercomputers, and the cloud.

Computing in HASS

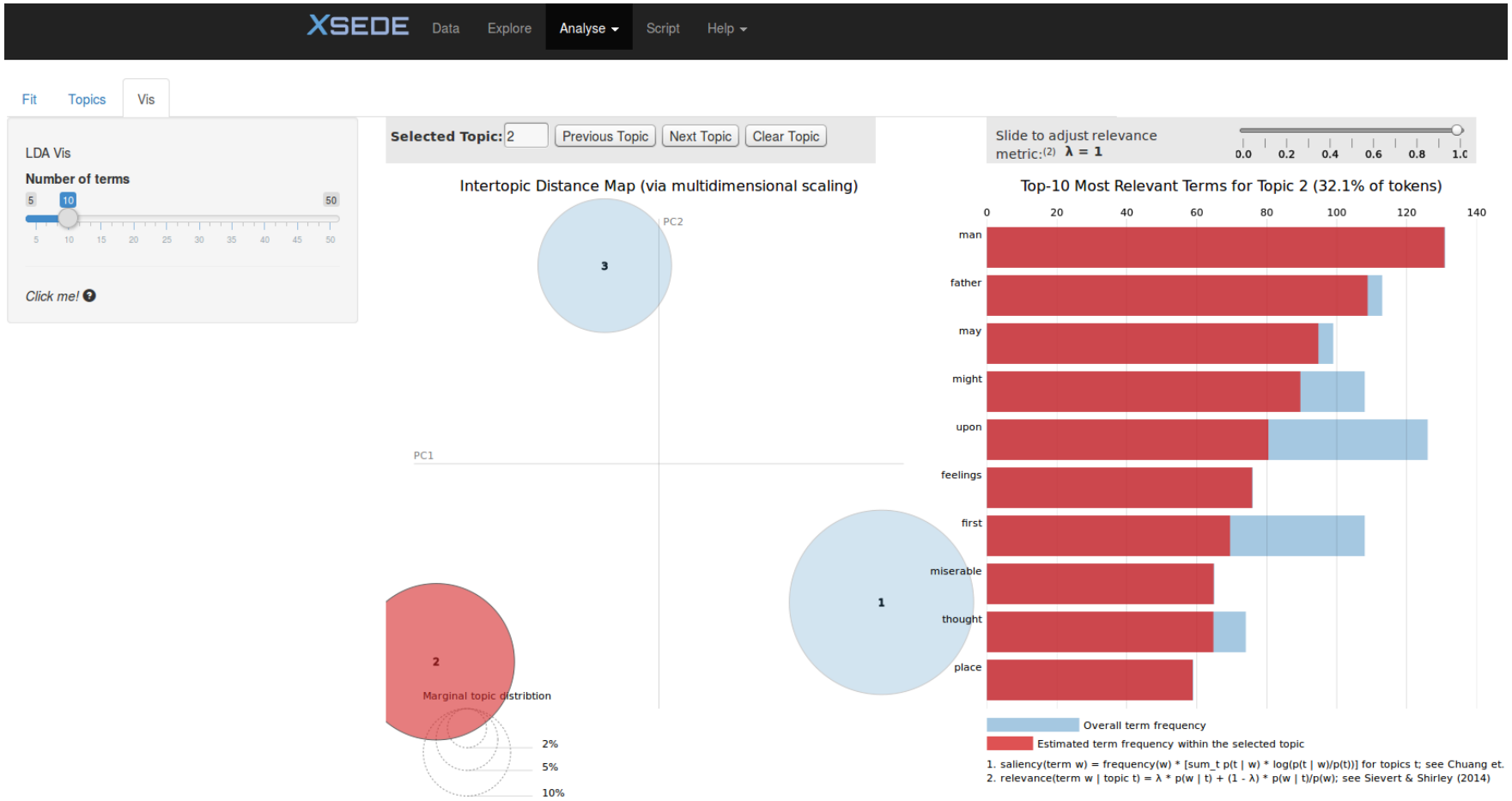
- Like GUI's, tolerate web services.
- Typically not programmers.
- Don't want this:

```
corpus <- tm::tm_map(corpus, tm::content_transformer(base::to
corpus <- tm::tm_map(corpus, tm::removePunctuation)
corpus <- tm::tm_map(corpus, tm::removeNumbers)
corpus <- tm::tm_map(corpus, tm::stripWhitespace)
corpus <- tm::tm_map(corpus, tm::removeWords, tm::stopwords("e
tdm <- tm::TermDocumentMatrix(corpus)
wordcount_table <- sort(rowSums(as.matrix(tdm)), decreasing=TR
```

TAG


TAG
Web Stack
Shiny
markdown
Visualization Stack
wordcloud
RColorBrewer
ggplot2
LDAvis
Analysis Stack
topicmodels
qdap
tm
SnowballC
Custom C
ngram

Analysis and Visualizations



Proposed Project

Two Elements

1. Fix the script capturer.
2. Demonstrate features in a complete workflow.

Script Capturer

TAG Script

This script is roughly what is run underneath TAG as you perform your analysis. The goal is to be able to enhance reproducibility, help you learn R, and speed up making minor changes to an analysis.

HOWEVER, this feature is currently very experimental and should not be considered fully implemented.

```
### WARNING: very experimental
library(TAG)


### Transform text

# Set lowercase
corpus <- tm::tm_map(corpus, tm::content_transformer(base::to
lower))
# Remove punctuation
corpus <- tm::tm_map(corpus, tm::removePunctuation)
# Remove numbers
corpus <- tm::tm_map(corpus, tm::removeNumbers)
# Remove extra whitespace
corpus <- tm::tm_map(corpus, tm::stripWhitespace)
```

Script Capturer

Who cares?

- Very important feature!
- Adds reliability, ability to make quick changes, work in batch...

Have:

- Uses regular expressions.
- Highly non-regular language...

Want:

- A proper parser.

Script Capturer: Challenges and Novelty

Challenge:

- Need C for reasonable performance.
- Basically has to be entirely custom.

That said...

- Inputs are well-defined and well-understood.
- I have lots of experience with C.

Complete Workflow Demonstration(s)

- Demonstrate capabilities of TAG.
- Written for and of interest to target audience (HASS).
- Relevant to a forthcoming paper...

Demonstration Possibilities

Challenge:

- Keeping it interesting, but also basic enough for intended audience.
- In HASS, a wordcloud is a paper!

Plan:

- Recreate examples from *Text Analysis with R for Students of Literature*.
- Go beyond...
- Stretch goal: stylometric analysis, *Wizard of Oz* books. (hash table)

Current Status

Status

- Script capturer: haven't started.
- Workflow: mostly done.
- Stretch goal: thought hard about it...

Thanks!

Questions?