

Latent Semantic Mapping

Exposing the Meaning behind Words and Documents

Session 136

Matthias Neeracher, Dr. Sc. Techn.

Senior Software Engineer

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

Session Overview

- What is LSM?
- How does it work?
- Using LSM
- Case studies
- Your application here
- Q&A

What Is Latent Semantic Mapping?

A technology to analyze text documents
according to their meaning
and classify them by topic

...allow me to demonstrate

Demo

A simple LSM example

Some LSM Applications

- Junk mail filter
 - Assess whether mail message is legitimate or spam
- Parental controls
 - Assess whether web page contains explicit words or other objectionable material
- Kana to Kanji conversion
 - Use topic of a document to disambiguate between ambiguous characters
- Localization
 - Use underlying topic of discourse to aid in string translation

How Does It Work?

Jerome Bellegarda, Ph.D.
Apple Distinguished Scientist

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

It Is All in the Name!

- “Mapping”
 - Represent words and documents as points in multidimensional space
 - From discrete to continuous entities
- “Semantic”
 - Mapping aimed at uncovering global fabric of language
 - Based on overall content/meaning of documents
- “Latent”
 - Meaning not obtained from a dictionary, but inferred directly from data
 - Based on word co-occurrences, automatically handle synonyms and multiple senses

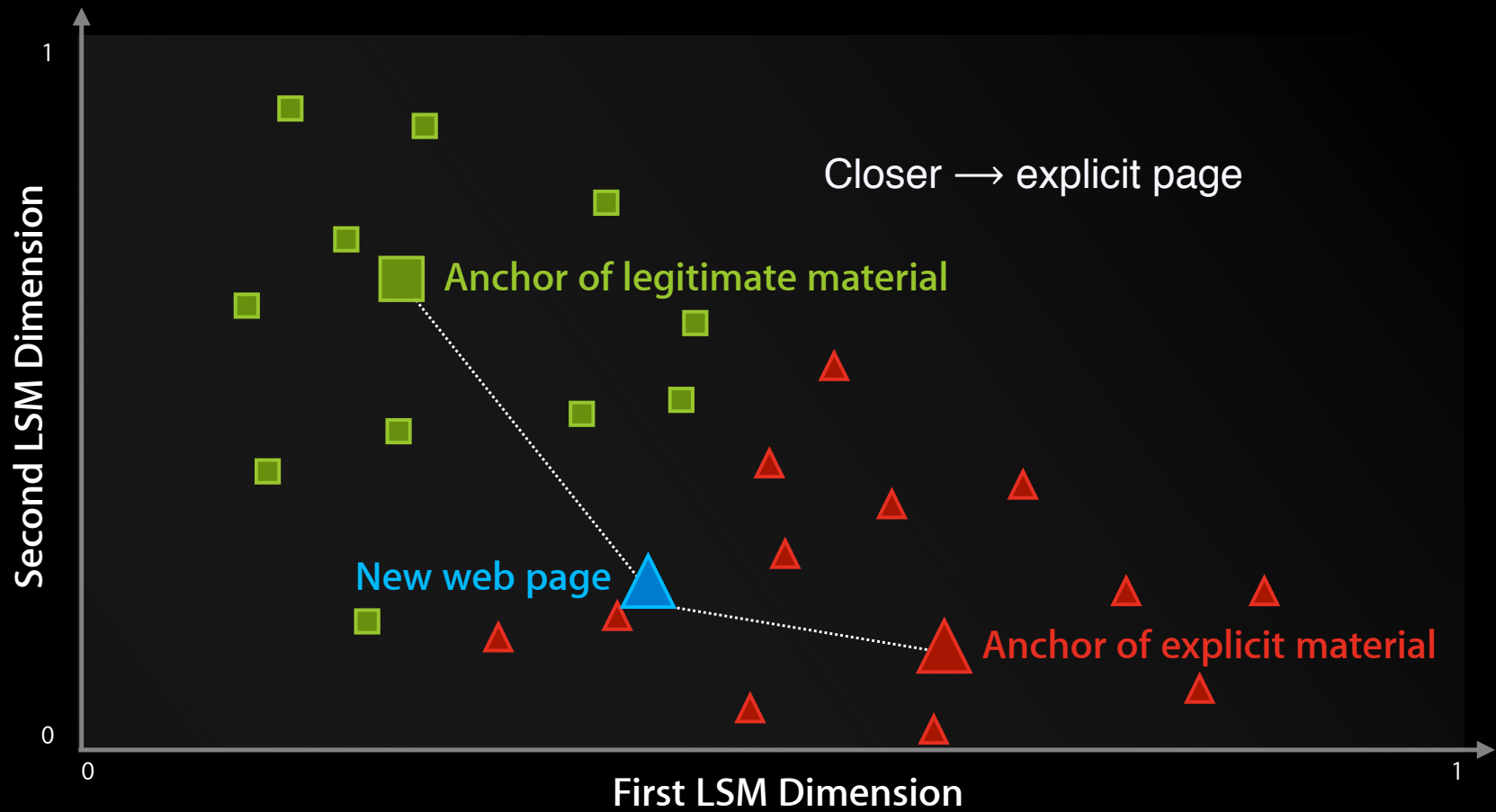
Latent

- Word co-occurrences
 - Words A and B present in the same document
 - Words A and C present in doc 1, words B and C present in doc 2
 - Words A and B “close” in LSM space
- Discover synonyms
 - “car” vs. “automobile”
 - “bank” vs. “financial institution”
- Discover multiple senses
 - “bank” + “rate” (→ finance)
 - “bank” + “river” (→ fishing)

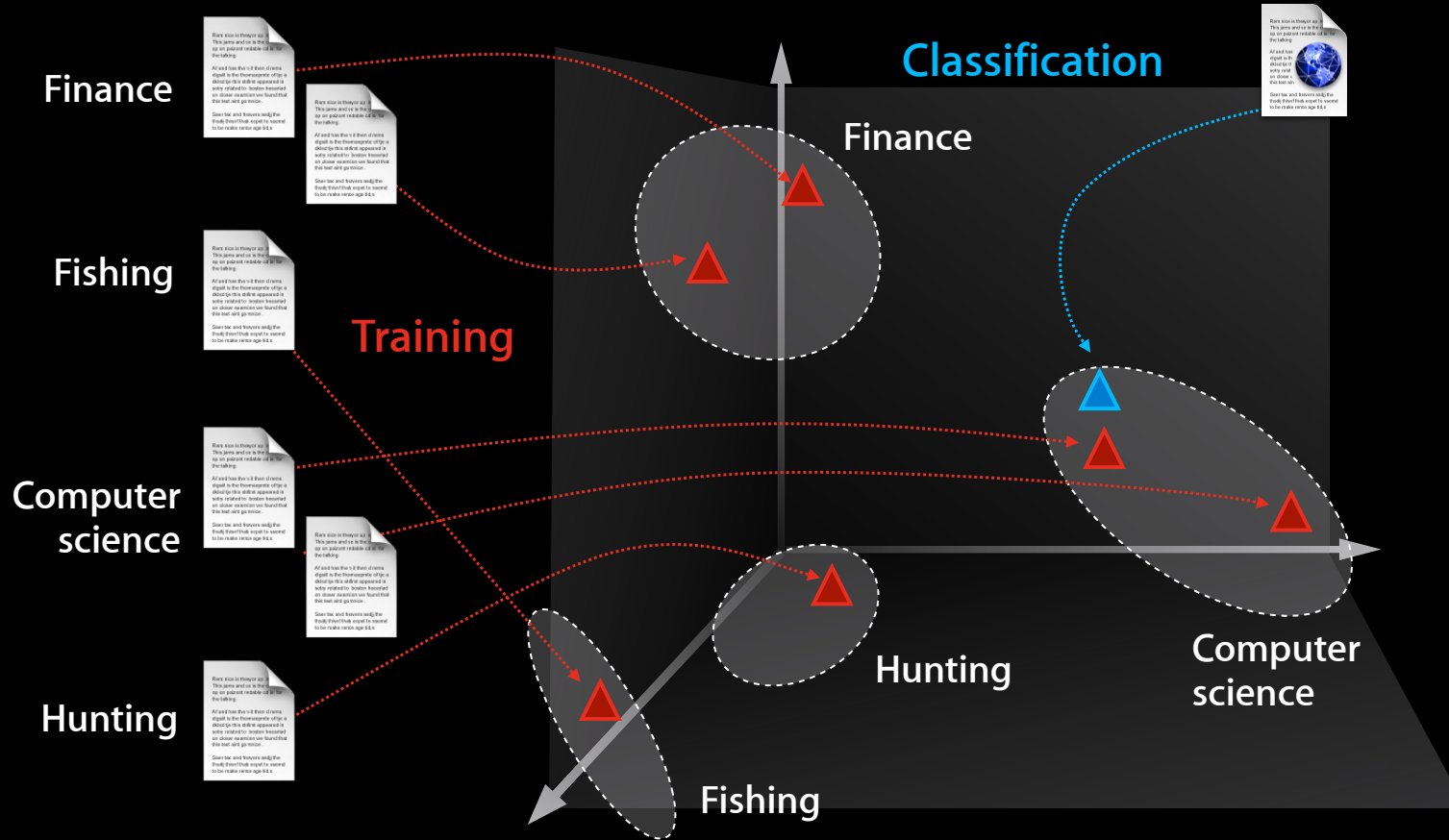
Semantic

- Example: parental controls
 - Assess whether web page is free of objectionable material
 - Separate “sex toys” from “sex education” (using underlying meaning)
 - Can leverage closeness in LSM space
 - “sex” + “toys” (→ probably objectionable)
 - “sex” + “education” (→ probably ok)
- LSM Implementation
 - Use two categories (one for explicit material, one for legitimate material)
 - Define two **semantic anchors** in LSM space
 - Evaluate each incoming web page against these two anchors

2-D Illustration



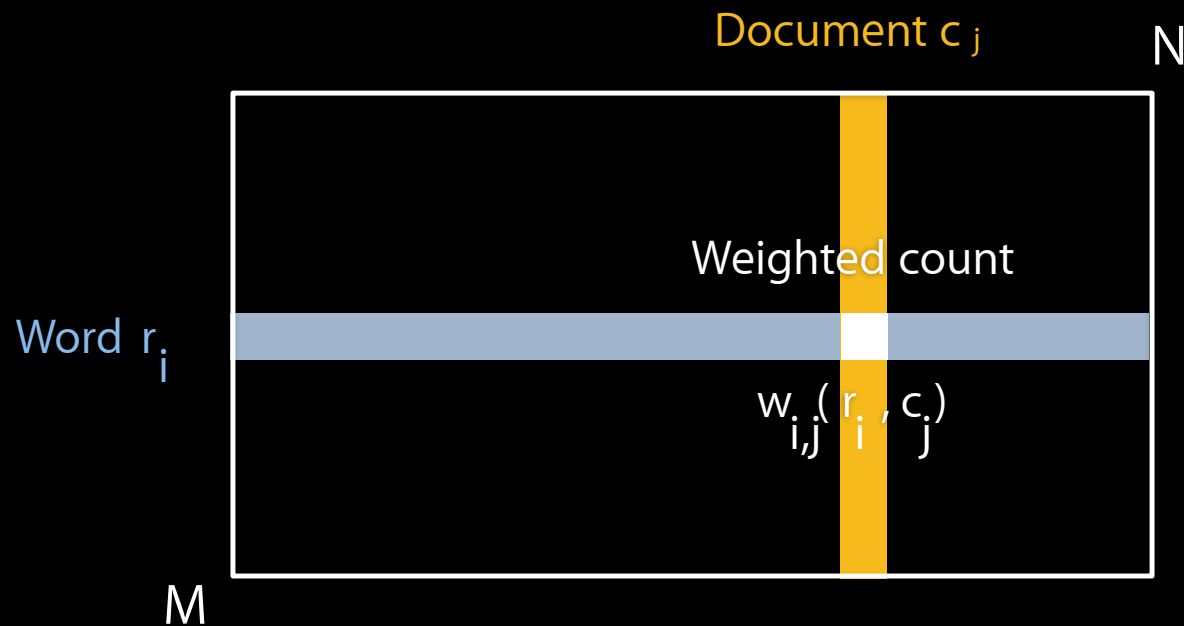
Mapping



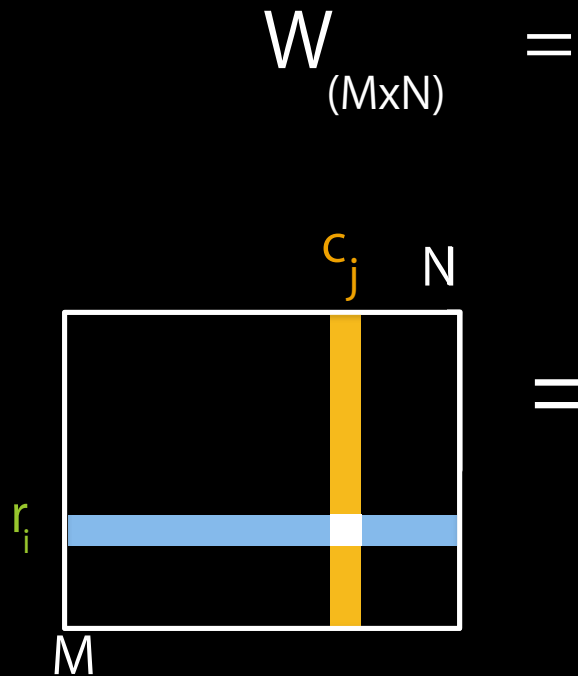
It Is Not (All) Magic!

Basic info

- How often does each word appear in each document?
- How often does each word appear in entire training data?



Singular Value Decomposition (SVD)



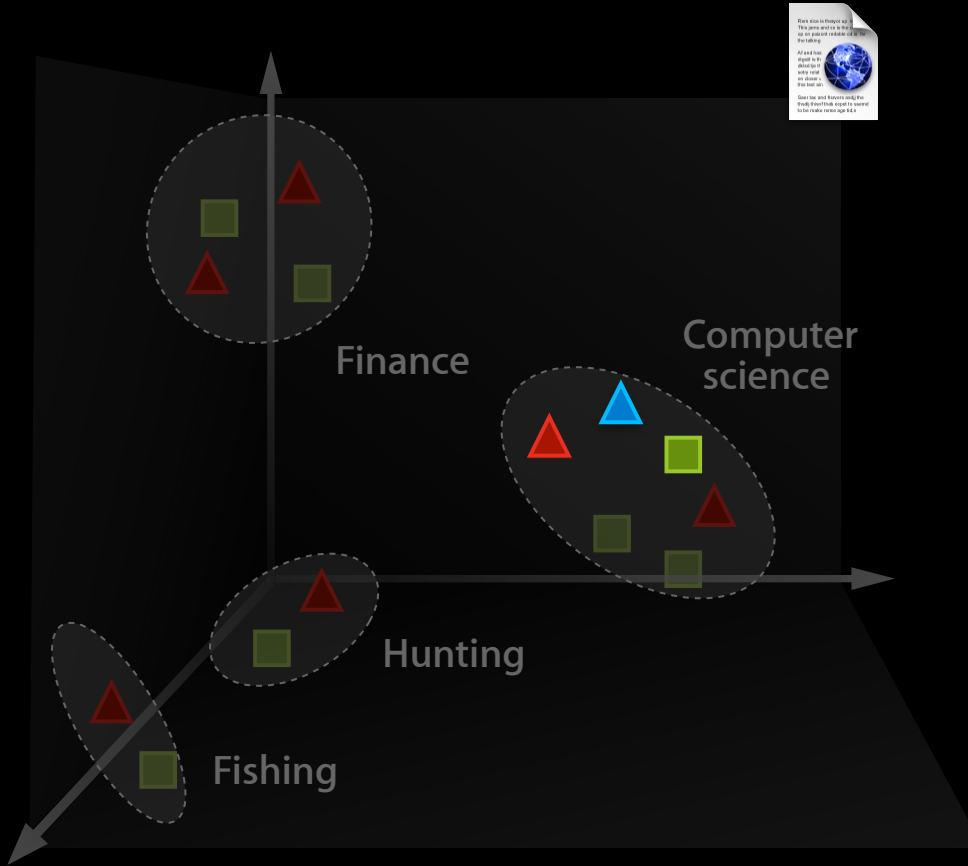
R: Number of dimensions retained

LSM Space

Documents



Words appearing
in documents



Caveats

- Intrinsic descriptive power
 - Shallow sense of “semantic” (tied to co-occurrences)
 - No actual “natural language understanding”
 - Word order is ignored (“bag of words” modeling)
 - Local constraints need to be added explicitly
- Critical importance of training data
 - Ambiguity: “river bank” and “Bank of Cupertino” in same doc?
 - Writing style: Wall Street Journal vs. Associated Press
- Offline training cost (in some apps)
 - SVD can take a long time with large matrices

Clustering

- Problem of ill-defined categories
 - In Kana to Kanji conversion, topic information is used to disambiguate between ambiguous characters
 - Analogous to “the **tale** of a princess” vs. “the **tail** of a peacock”
 - But Japanese corpus contains over 300,000 documents
 - How to best extract and leverage topic information?

Clustering (Cont.)

- Two solutions
 - Manual assignment of documents into topics (categories)
 - LSM clustering
 - Initial LSM space where each document is a separate category
 - Data-driven clustering to reduce number of categories
 - (Optionally) new LSM space using clustered data

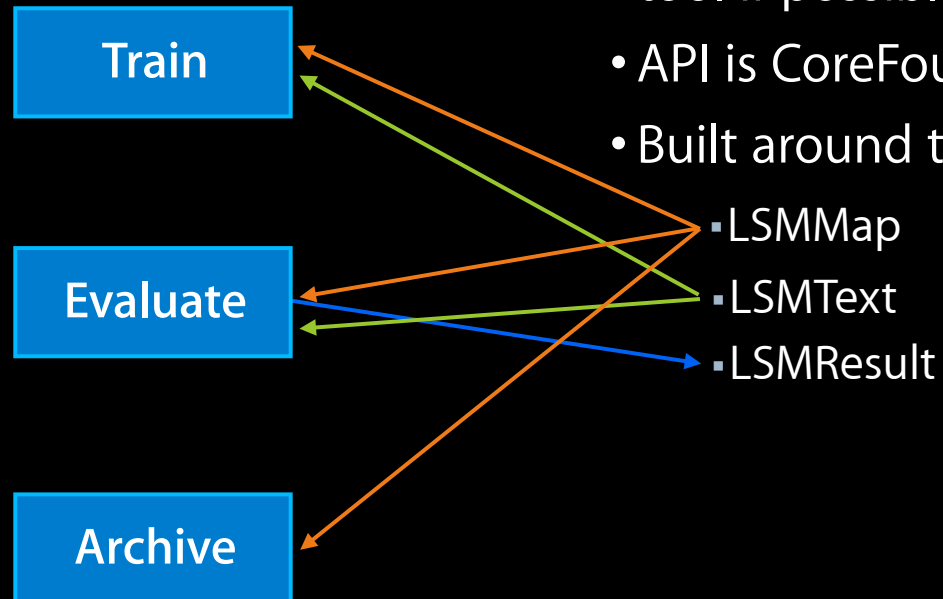
Two Implementations

- K-means clustering
 - Pick initial cluster centroids (“seeds”)
 - Compute distances to these centroids
 - Adjust centroids accordingly and iterate
 - Caveat: sensitive to initial cluster assignment
- Agglomerative clustering
 - Compute all pair-wise distances between points
 - Merge closest pair, replace by its centroid
 - Adjust affected distances and iterate
 - Caveat: prohibitive for large data sets

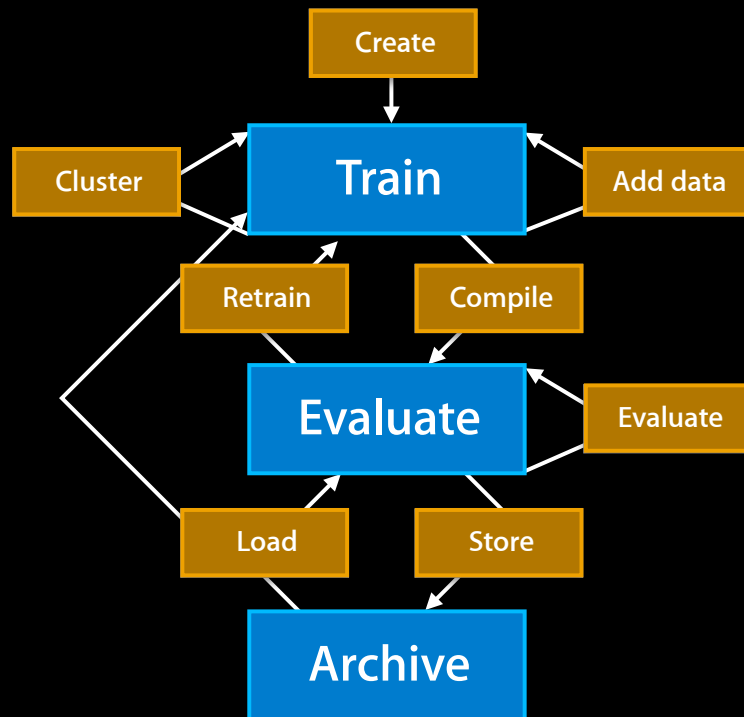
Using the LSM API

Basics of LSM Programming

- Prototype with the command-line tool if possible
- API is CoreFoundation-based
- Built around three types



Using LSM Maps



LSMMapCreate

LSMMapAddCategory
LSMMapAddText

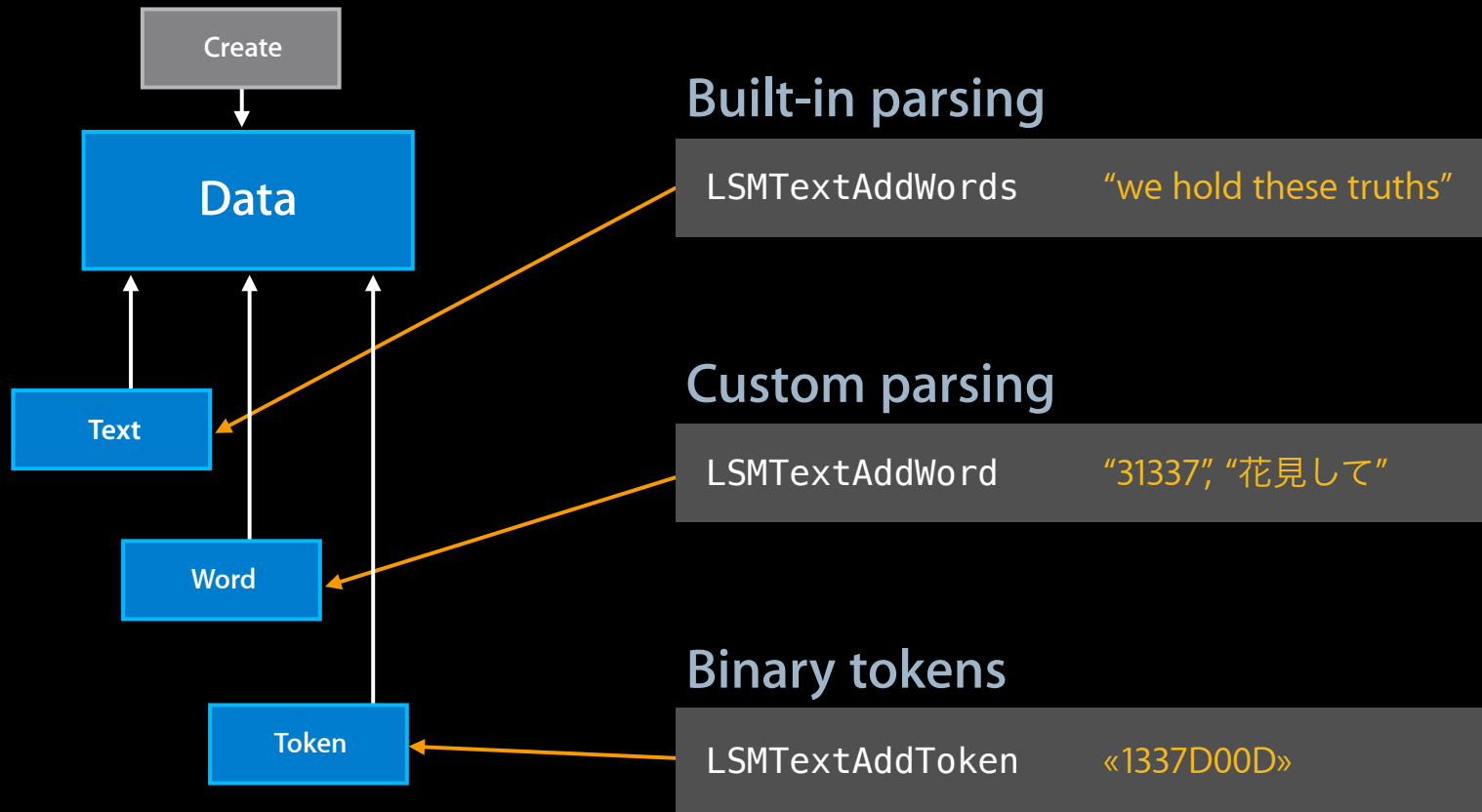
LSMMapCompile
LSMMapStartTraining

LSMResultCreate

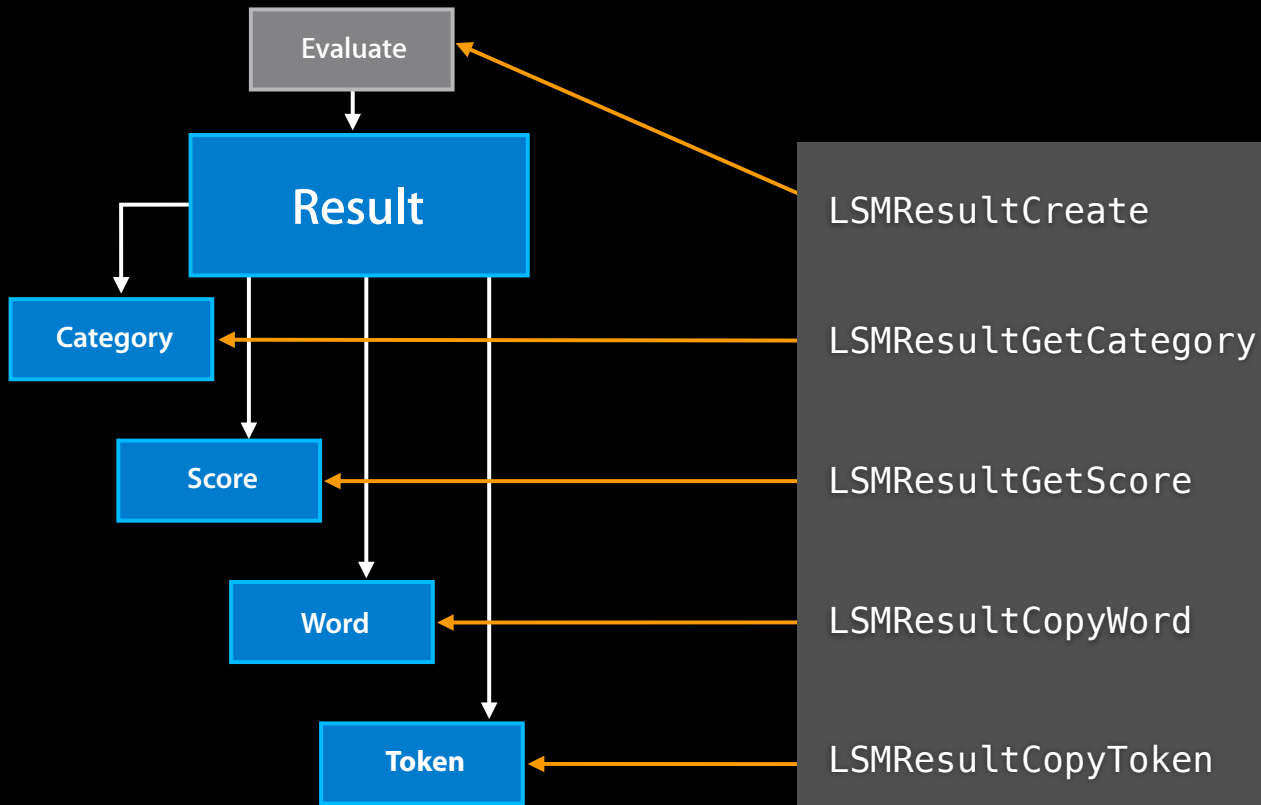
LSMMapWriteToURL
LSMMapCreateFromURL

LSMMapCreateClusters
LSMMapApplyClusters

More Than Words



Evaluating a Text



Case Studies

Case Study: Junk Mail Filtering

- Two categories: legitimate/junk
- Biased toward legitimate

```
LSMResultGetCategory(res, 0) == kJunk  
&& LSMResultGetScore(res, 0) > kJunkThreshold
```

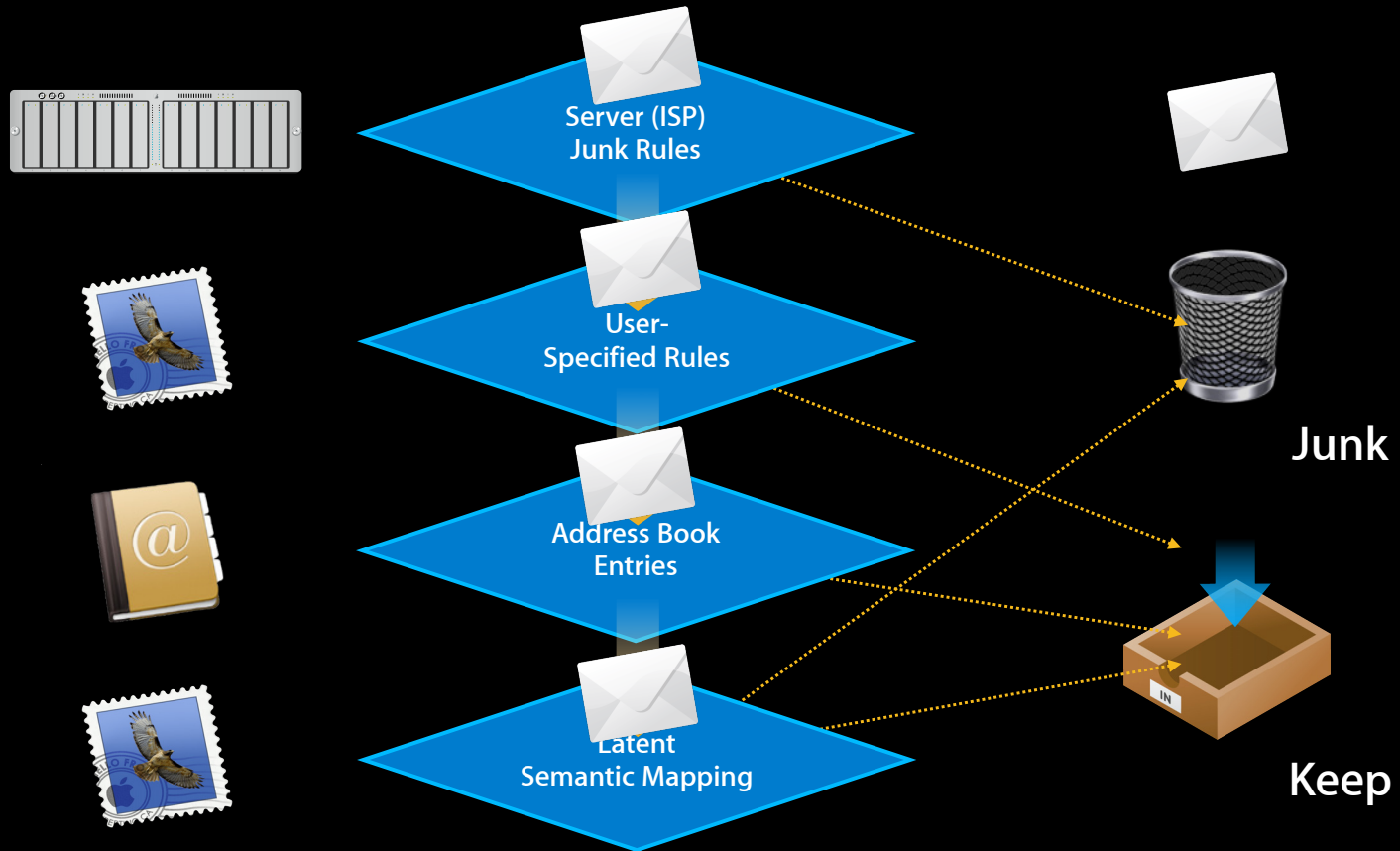
- Parsing can be difficult: m.o.n.e.y, víåg®ä

```
LSMTextAddWords(text, words, NULL,  
kLSMTextApplySpamHeuristics);
```

- Map contains all sorts of offensive words

```
LSMMapCreate(kCFAllocatorDefault, kLSMMapHashText);
```

LSM as Last Line of Defense



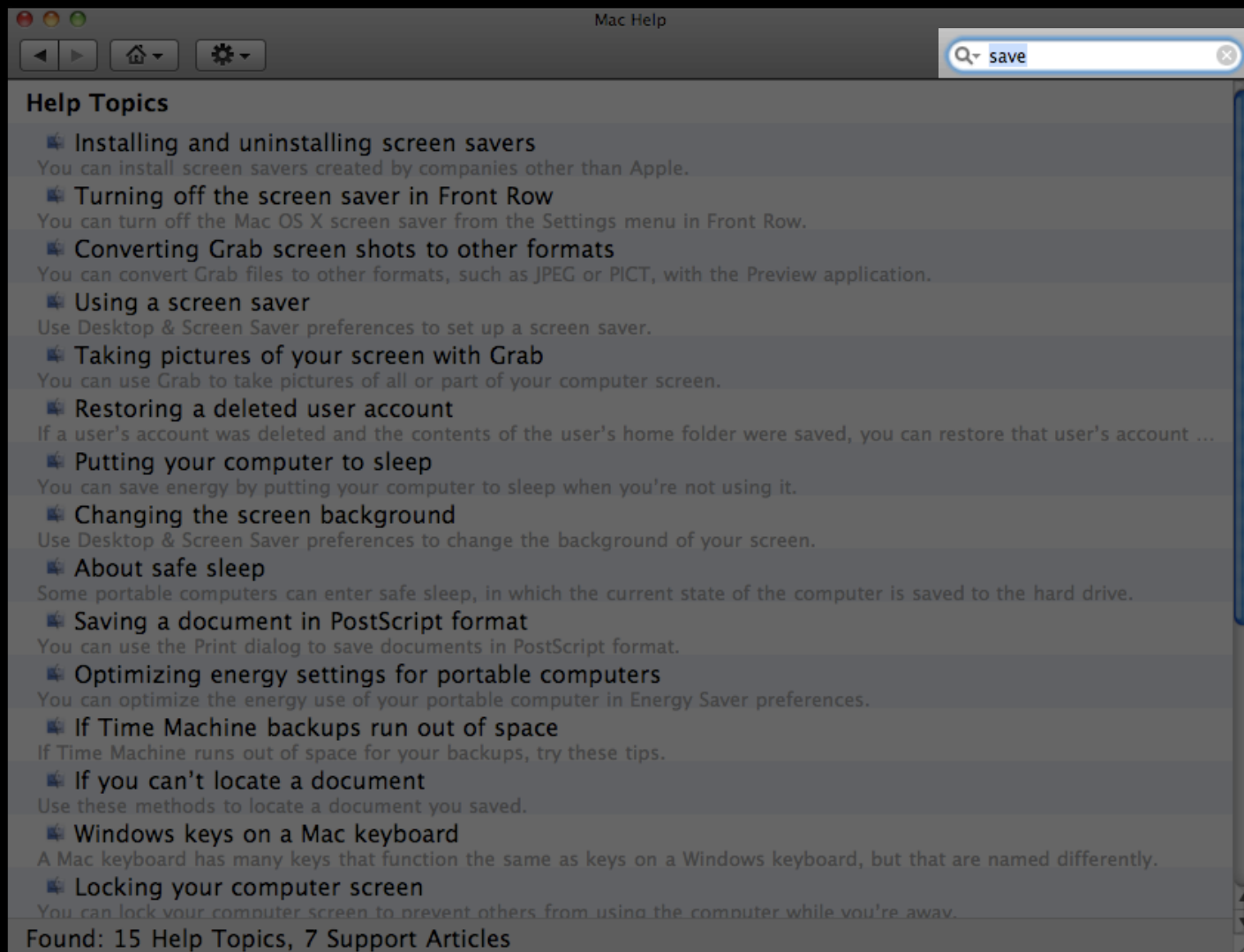
Case Study: Help

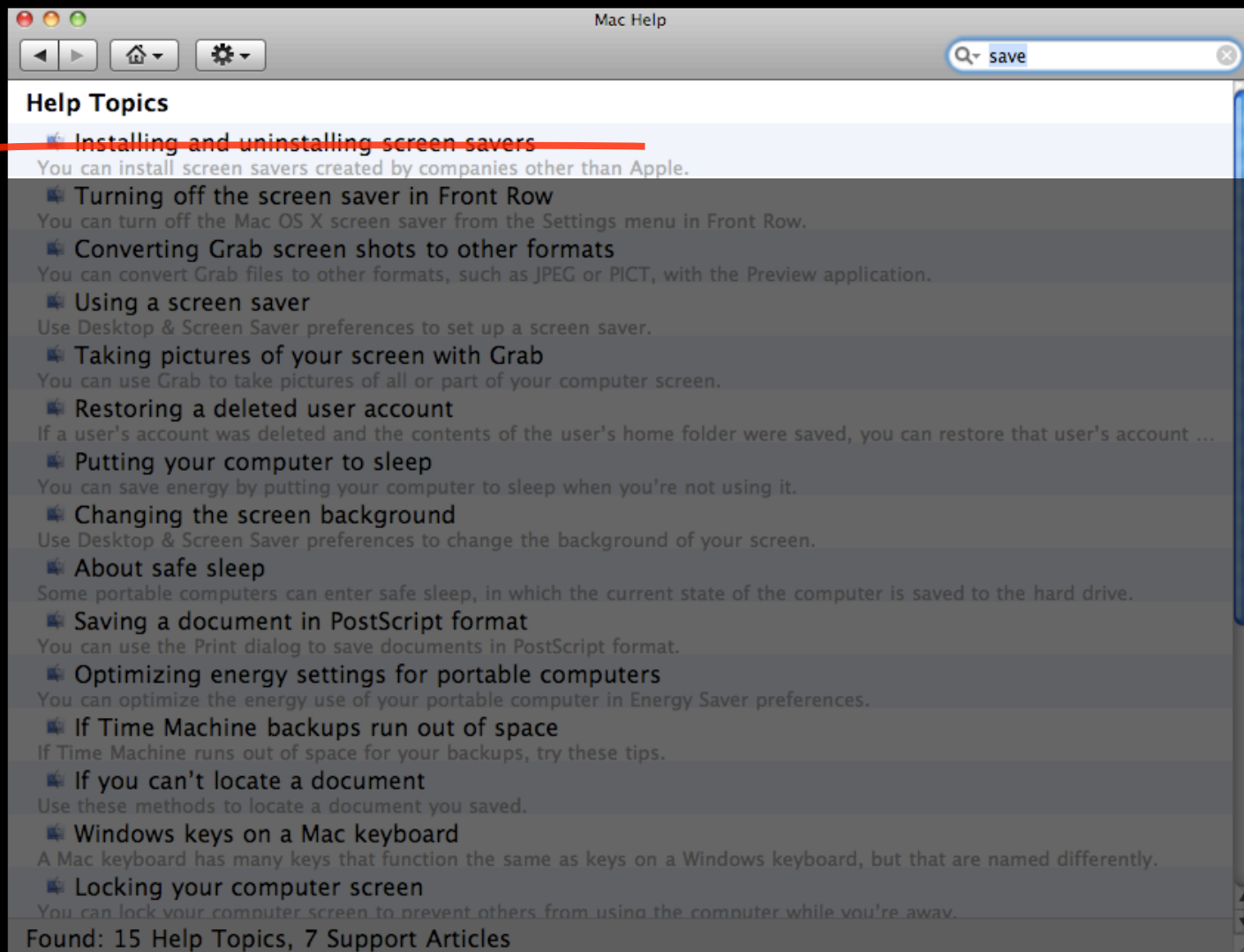
Kim Silverman, Ph.D.
Principal Research Scientist

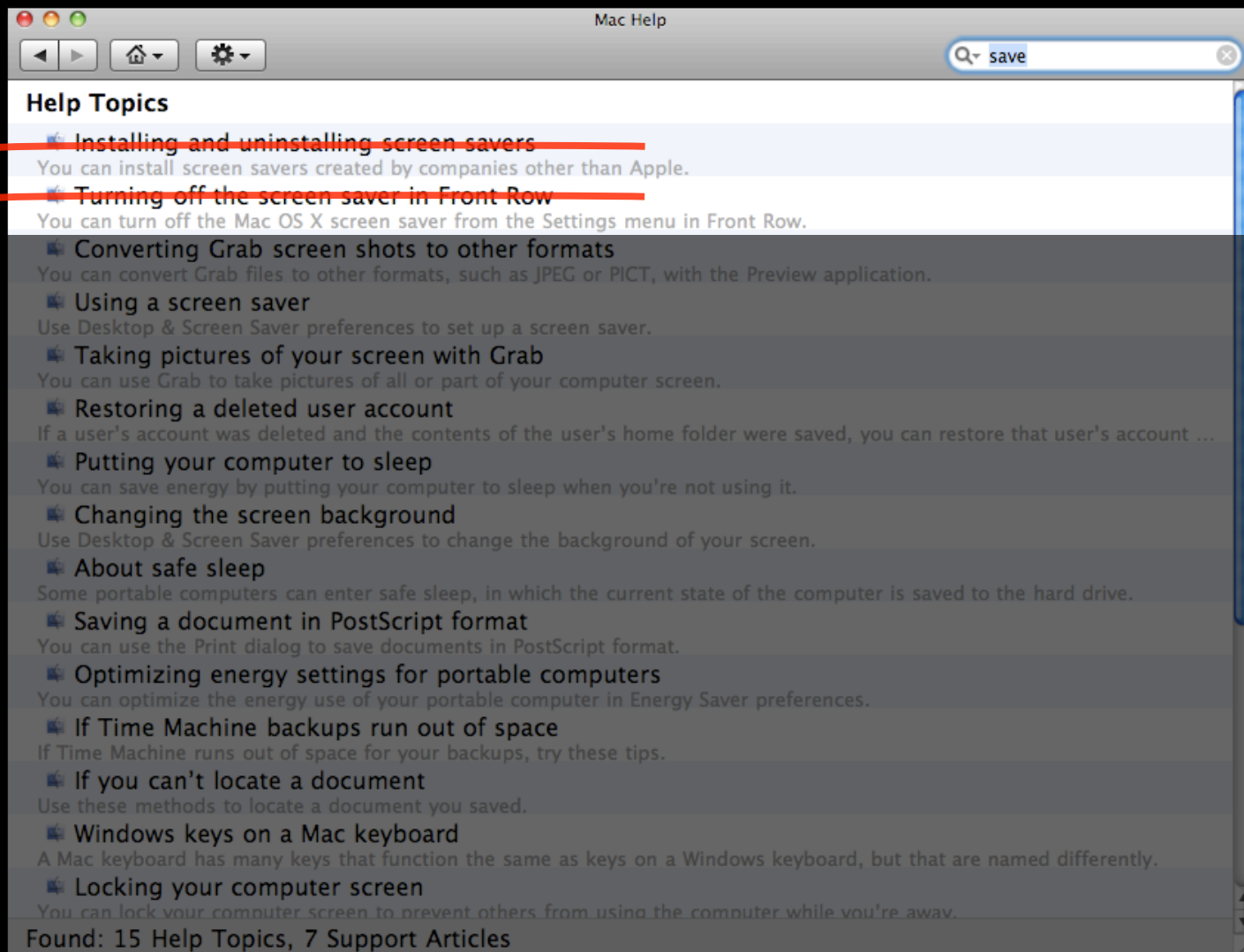
These are confidential sessions—please refrain from streaming, blogging, or taking pictures

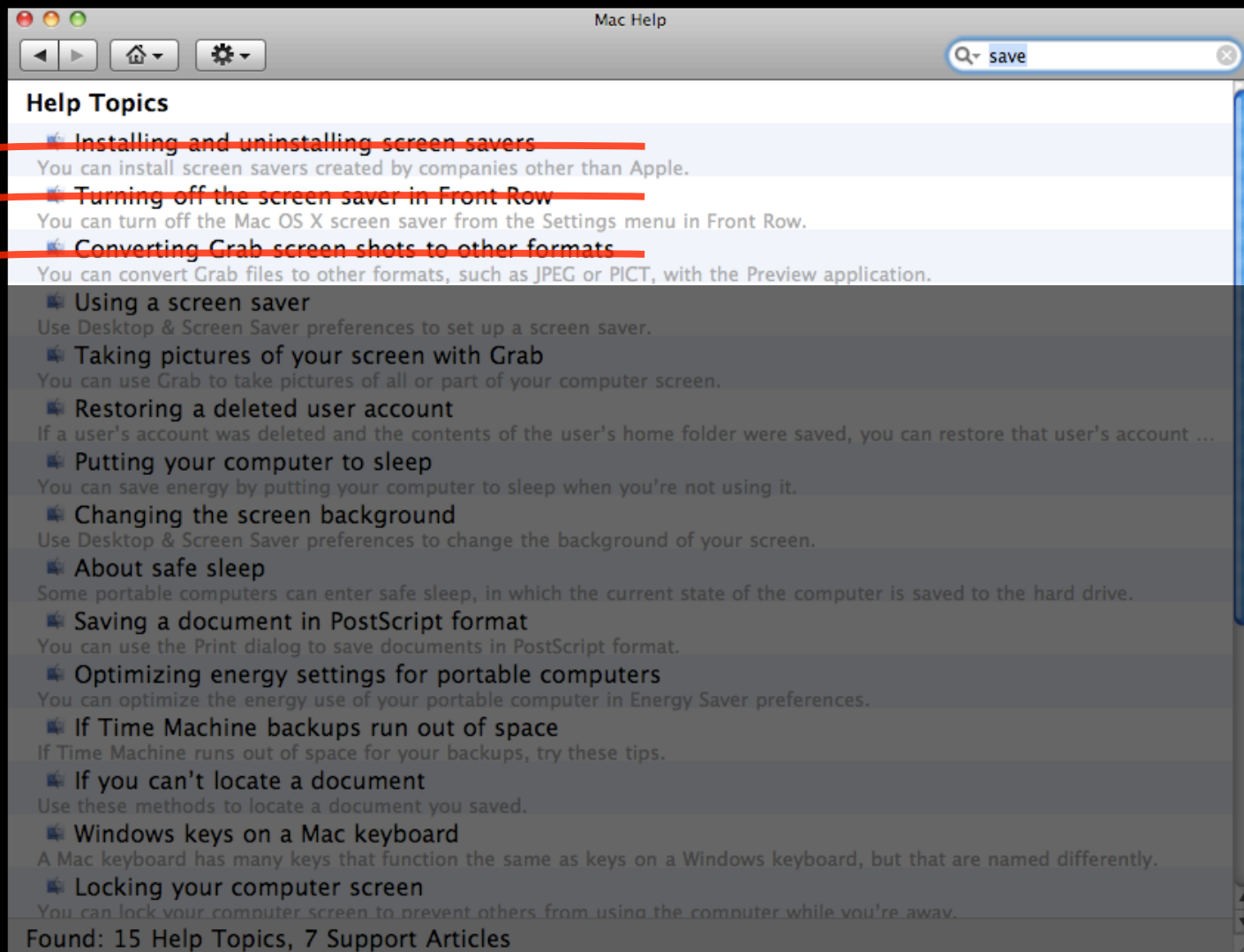
Problem

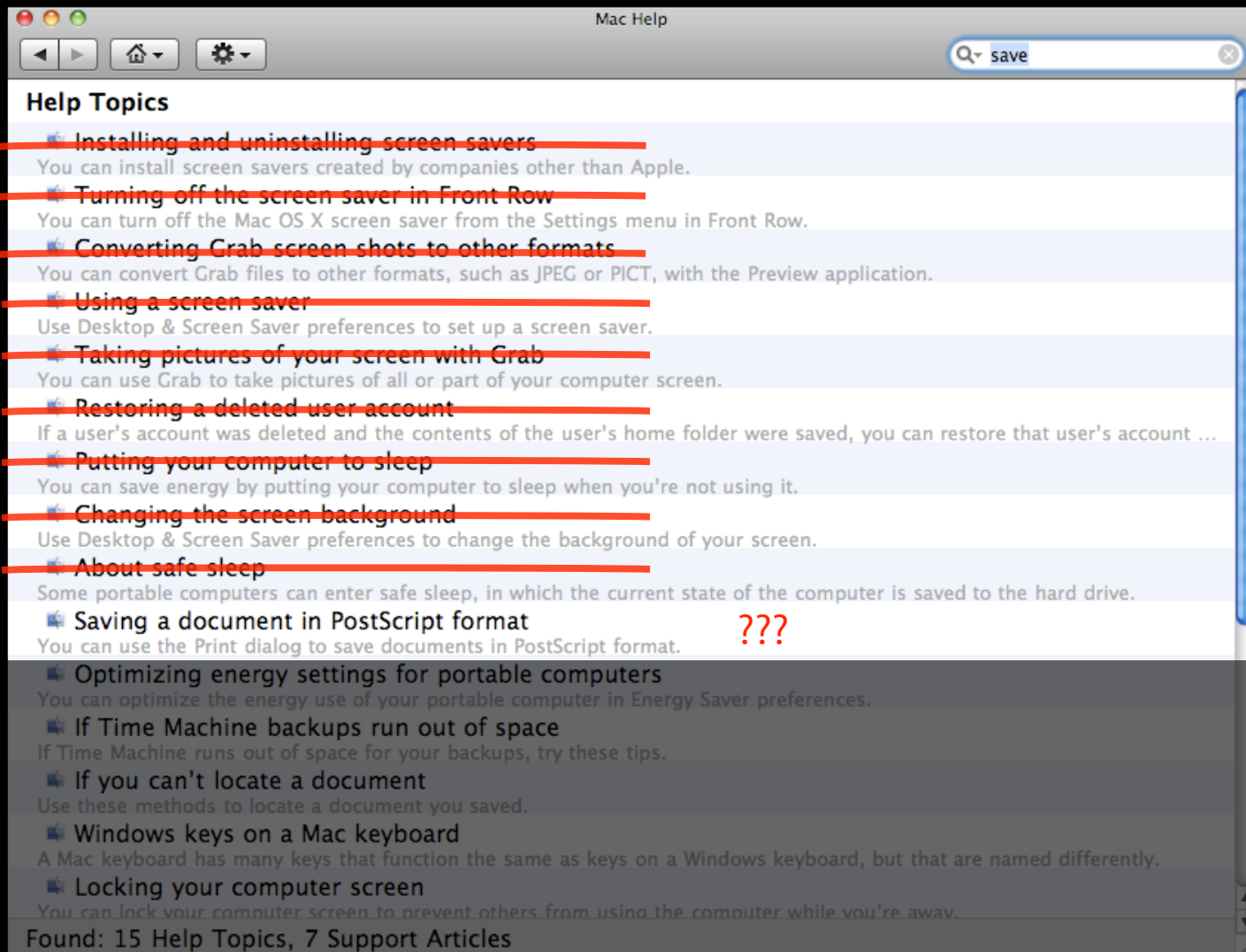
- Unsatisfying results when people type queries to help














Current Approach


- Look for documents that contain words in the query
- Use hand-inserted synonyms for common typos, different forms of words, etc.


Latent Semantic Mapping


 **Installing and uninstalling screen savers**


 **Turning off the screen saver in Front Row**

 **Converting Grab screen shots to other formats**

 **Using a screen saver**


 **Taking pictures of your screen with Grab**


 **Restoring a deleted user account**

 **Putting your computer to sleep**


When you're not using your computer, you can save energy by putting it to sleep. When your computer is in sleep, it's turned on but consumes much less power. It takes the computer less time to wake from sleep than it does to start up after being turned off.


Latent Semantic Mapping

 **Installing and uninstalling screen savers**
You can install screen savers created by companies other than Apple and use them with Screen Saver preferences on your computer.


 **Turning off the screen saver in Front Row**
If you have Front Row open, the Mac OS X screen saver appears if there is no activity after a period of time. You can turn off the screen saver from the Settings menu in Front Row.

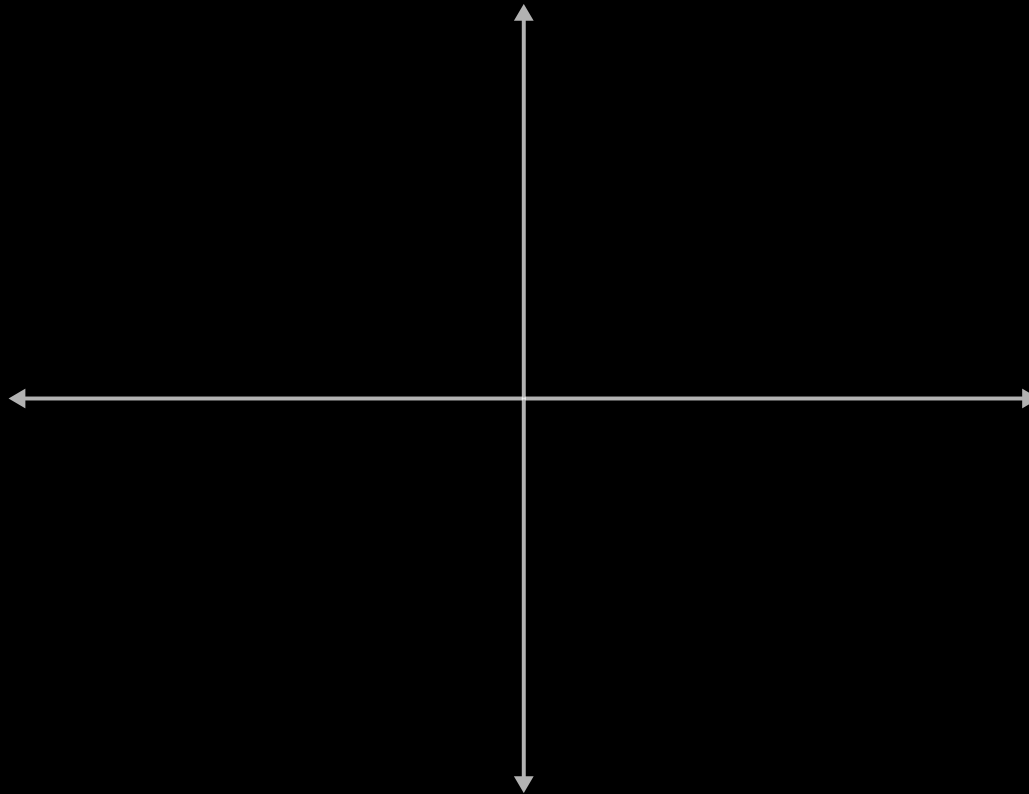
 **Converting Grab screen shots to other formats**
Grab saves screen shots as files in TIFF format. If you want to use your screen shots on the web, in email, or in a word processor, you can use the Preview application to convert the TIFF files to other formats, such as JPEG or PNG.

 **Using a screen saver**
You can have images appear and move around on your screen (called a "screen saver") when you aren't using your computer. You might want to use a screen saver to hide the items on your desktop while you're away.

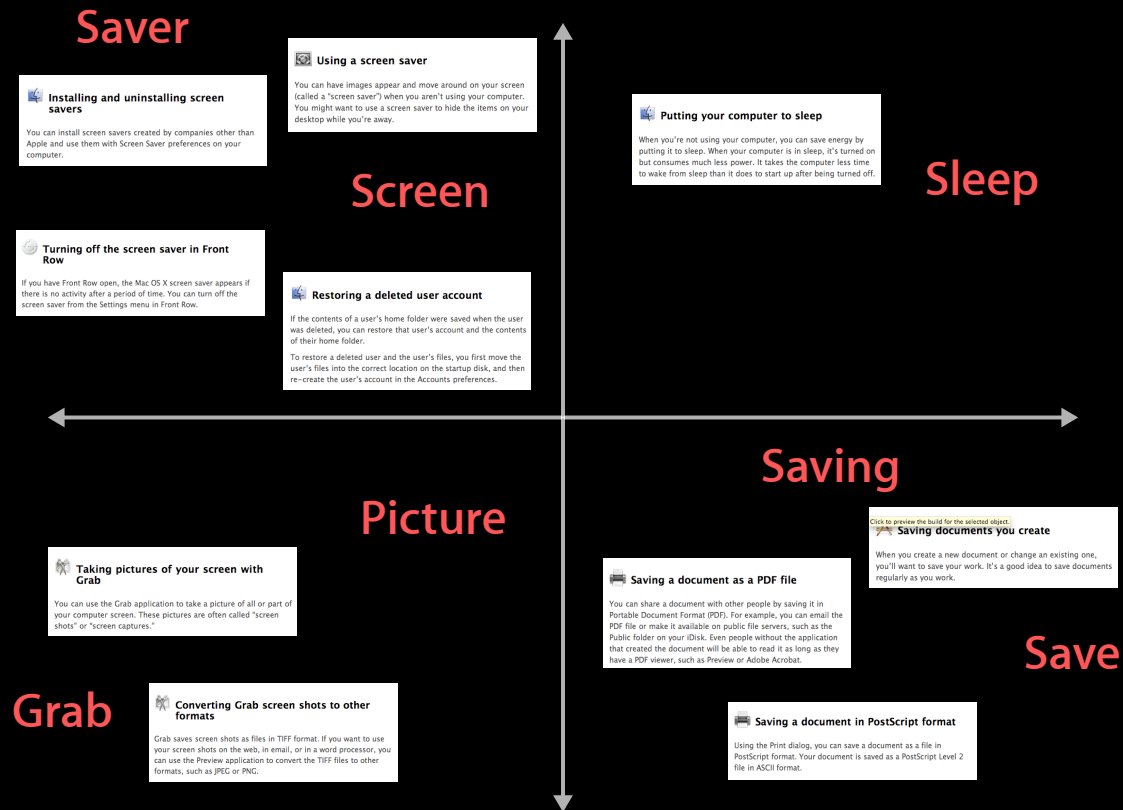
 **Taking pictures of your screen with Grab**
You can use the Grab application to take a picture of all or part of your computer screen. These pictures are often called "screen shots" or "screen captures."

 **Restoring a deleted user account**
If the contents of a user's home folder were saved when the user was deleted, you can restore that user's account and the contents of their home folder.
To restore a deleted user and the user's files, you first move the user's files into the correct location on the startup disk, and then re-create the user's account in the Accounts preferences.

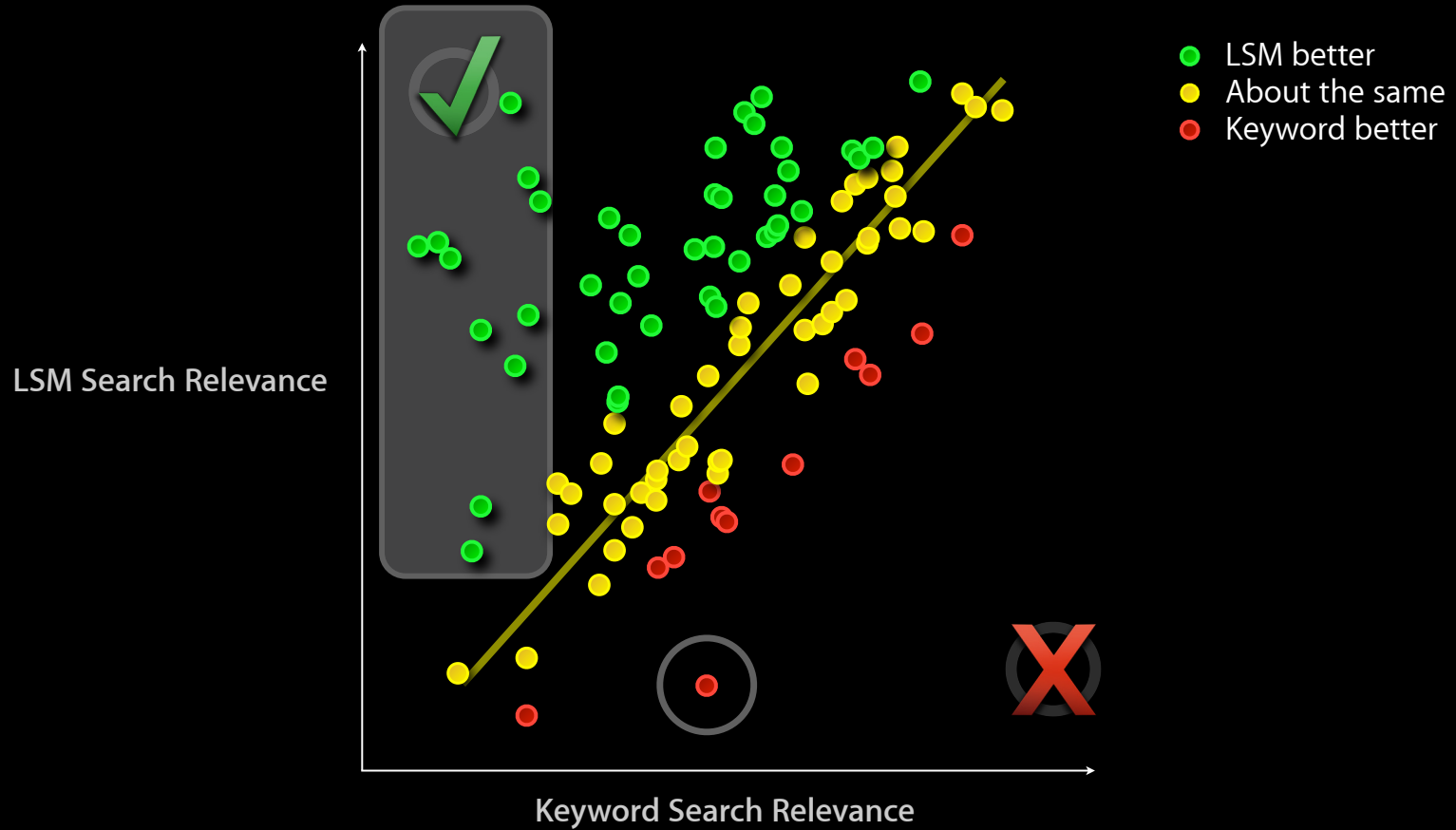
 **Putting your computer to sleep**
When you're not using your computer, you can save energy by putting it to sleep. When your computer is in sleep, it's turned on but consumes much less power. It takes the computer less time to wake from sleep than it does to start up after being turned off.



Latent Semantic Mapping



100 Queries



How to Improve LSM Results

Documents



Preprocessor

LSM
Framework

Postprocessor

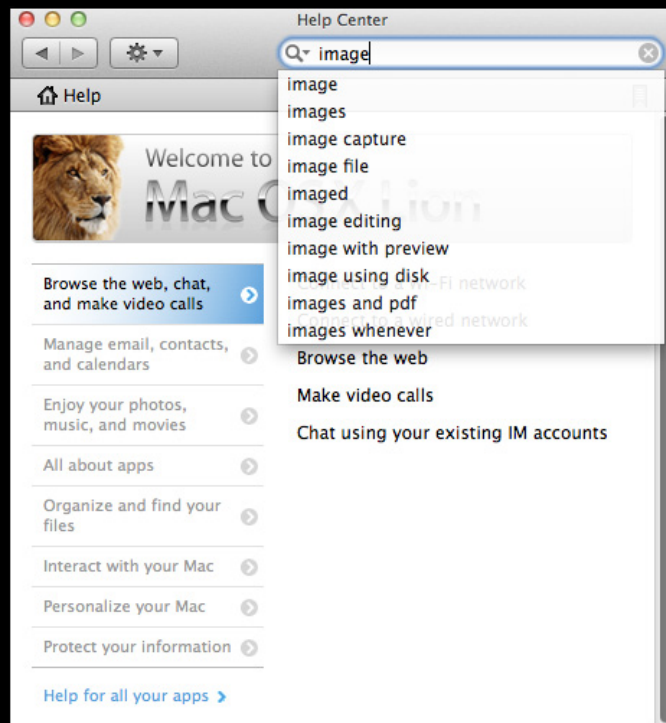
Results

Preprocessing Text

- Use n-grams
 - Word pairs, word triplets
 - e.g., “double click” vs. “key click”
- Remove unwanted/irrelevant text
 - HTML tags
 - “click here”, “return to contents”
- Stemming
 - “save” -> saves, saved, saving
 - But not “saver”

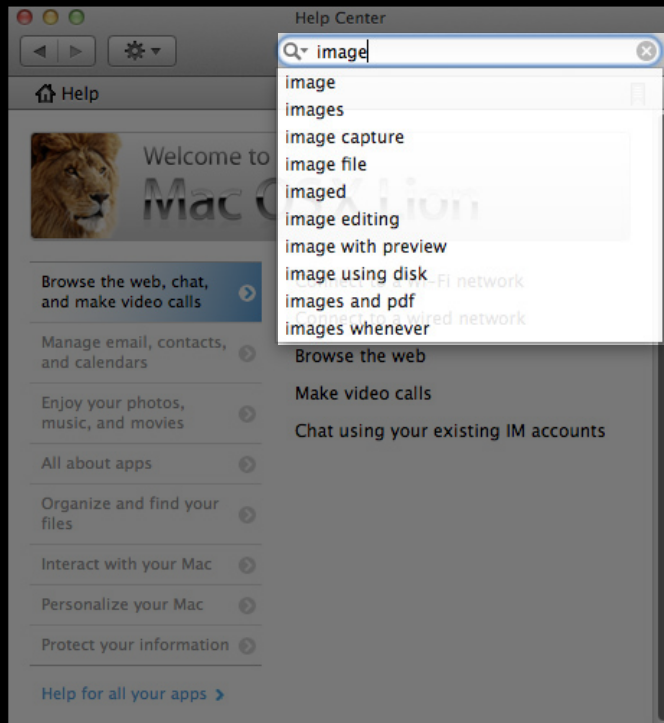
Postprocessing Text

Autocompletion and suggestions for Help



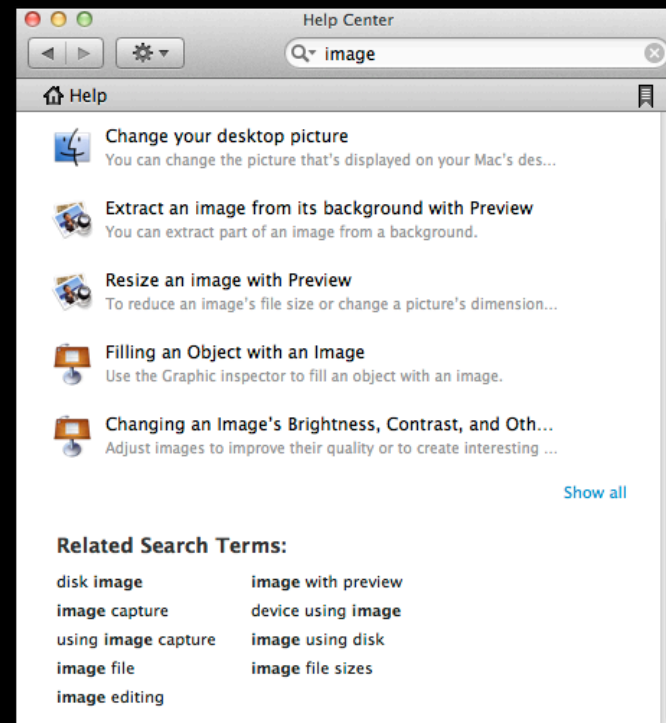
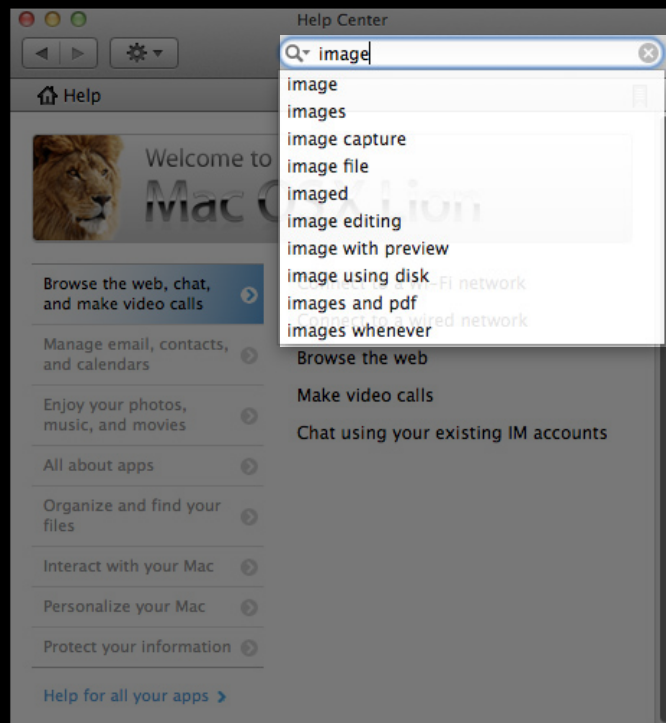
Postprocessing Text

Autocompletion and suggestions for Help



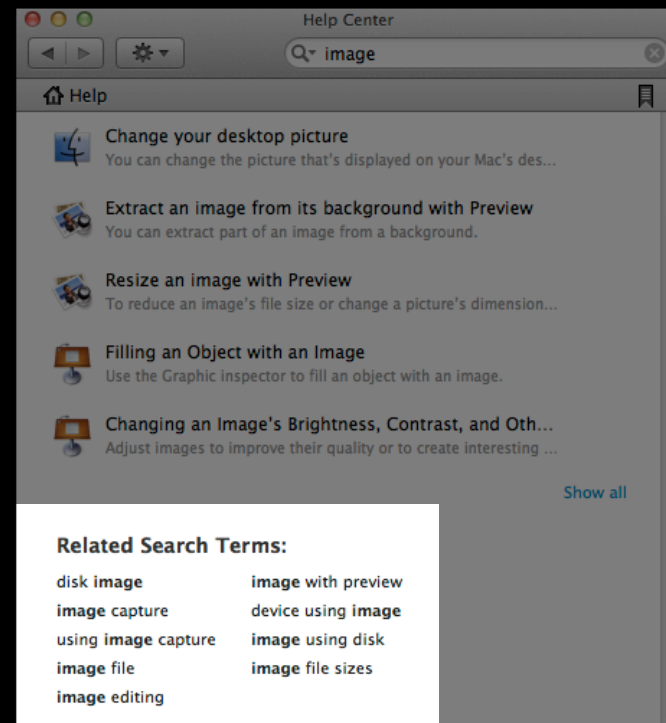
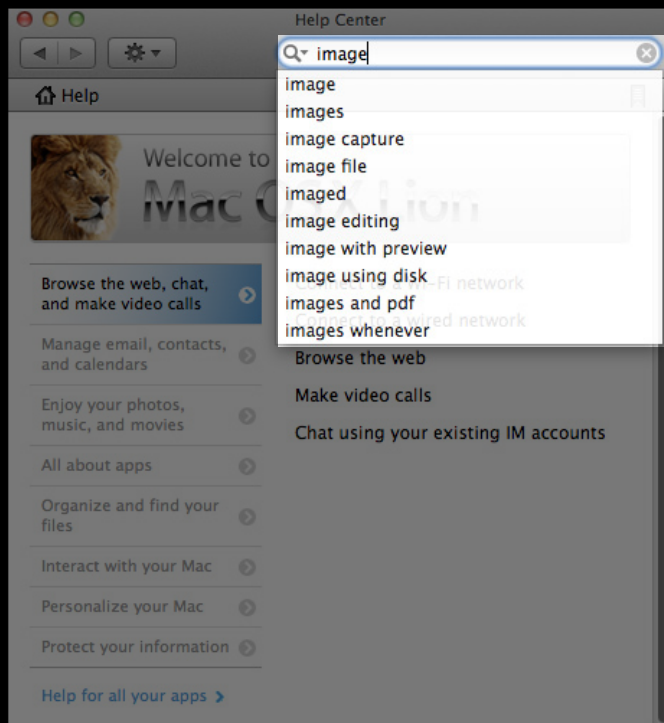
Postprocessing Text

Autocompletion and suggestions for Help



Postprocessing Text

Autocompletion and suggestions for Help



Postprocessing Text

Filter nearest n-grams

Clean up your workspace

Set up your workspace

Postprocessing Text

Filter nearest n-grams

Clean **up your workspace**

Set **up your workspace**



up your workspace

Making LSM Work for You

What Can LSM Categorize for Me?

- Bookmarks
- RSS feeds
- Books/CDs/DVDs (by fetching reviews/abstracts)
- Wines and cheeses
- DNA sequences
- ...

Guidelines—General

- Can LSM handle the task?
 - Problem is syntactic in nature
 - “Find dates, times, email addresses, etc.”
 - Problem is semantic in nature
 - “Sort documents by topic”

- Are the categories distinct enough?
 - Economy vs. business
 - Economy vs. entertainment



Guidelines—Testing

- Validation data
 - Partition training data into 10 random chunks
 - Train on first nine chunks, test on last (held out)
 - Repeat sequentially (round-robin) and average results

Guidelines—Testing (Cont.)

- What if outcome looks strange?
 - Try again with (short) stopword list
 - Words appearing roughly equally in all categories (“the,” “in”)
 - Try experimenting with number of dimensions
 - Default is number of categories, but for natural language problems use between 100 and 300

Guidelines—Training

- Quality of training data
 - Representative of full breadth of domain
 - As balanced as possible in each category

Guidelines—Training (Cont.)

- Quantity of training data
 - Large enough to cover variability
 - Rule of thumb for large vocabulary applications
 - Preferably $> 30,000$ unique words
 - Larger as more categories are added
 - Larger still if data changes over time (for example, news)

Final Recommendation

Integrate LSM with other source(s) of knowledge

- LSM tends to complement other techniques
 - It often can improve the robustness of the overall system
- Example 1: Junk Mail filter
 - Complements (instead of replacing) white lists, black lists, and handwritten rules
- Example 2: Kana to Kanji
 - Conversion uses LSM as an additional source of information to be exploited in final decision

Go Forth and Map Some Text!

For More Information...

More Information

Bill Dudney

Application Frameworks Evangelist
dudney@apple.com

Mailing List

Latent Semantic Mapping Mailing List
<http://lists.apple.com/mailman/listinfo/latentsemanticmapping>

Documentation

Latent Semantic Mapping Reference
<http://developer.apple.com/documentation/TextFonts/Reference/LatentSemanticMapping/index.html>

Apple Developer Forums

<http://devforums.apple.com>

Q&A

