Introduction to Text Mining with R

Dictionary Methods

Simon Munzert | IPSDS

Dictionary methods

Dictionary methods

Classifying documents when categories are known:

- Lists of words that correspond to each category:

Dictionary methods

Classifying documents when categories are known:

- Lists of words that correspond to each category:
    - Positive or negative, for sentiment

Dictionary methods

Classifying documents when categories are known:

- Lists of words that correspond to each category:
  - Positive or negative, for sentiment
  - Sad, happy, angry, anxious... for emotions

Dictionary methods

Classifying documents when categories are known:

- Lists of words that correspond to each category:
  - Positive or negative, for sentiment
  - Sad, happy, angry, anxious... for emotions
  - Insight, causation, discrepancy, tentative... for cognitive processes

Dictionary methods

Classifying documents when categories are known:

- Lists of words that correspond to each category:
  - Positive or negative, for sentiment
  - Sad, happy, angry, anxious... for emotions
  - Insight, causation, discrepancy, tentative... for cognitive processes
  - Sexism, homophobia, xenophobia, racism... for hate speech

Dictionary methods

Classifying documents when categories are known:

- Lists of words that correspond to each category:
  - Positive or negative, for sentiment
  - Sad, happy, angry, anxious... for emotions
  - Insight, causation, discrepancy, tentative... for cognitive processes
  - Sexism, homophobia, xenophobia, racism... for hate speech

    many others: see LIWC, VADER, SentiStrength, LexiCoder...

Dictionary methods

Classifying documents when categories are known:

- Lists of words that correspond to each category:
  - Positive or negative, for sentiment
  - Sad, happy, angry, anxious... for emotions
  - Insight, causation, discrepancy, tentative... for cognitive processes
  - Sexism, homophobia, xenophobia, racism... for hate speech

  many others: see LIWC, VADER, SentiStrength, LexiCoder...
- Count number of times they appear in each document

Dictionary methods

Classifying documents when categories are known:

- Lists of words that correspond to each category:
  - Positive or negative, for sentiment
  - Sad, happy, angry, anxious... for emotions
  - Insight, causation, discrepancy, tentative... for cognitive processes
  - Sexism, homophobia, xenophobia, racism... for hate speech

  many others: see LIWC, VADER, SentiStrength, LexiCoder...
- Count number of times they appear in each document
- Normalize by document length (optional)

Dictionary methods

Classifying documents when categories are known:

- Lists of words that correspond to each category:
  - Positive or negative, for sentiment
  - Sad, happy, angry, anxious... for emotions
  - Insight, causation, discrepancy, tentative... for cognitive processes
  - Sexism, homophobia, xenophobia, racism... for hate speech

    many others: see LIWC, VADER, SentiStrength, LexiCoder...
- Count number of times they appear in each document
- Normalize by document length (optional)
- **Validate, validate, validate**.

Dictionary methods

Classifying documents when categories are known:

- Lists of words that correspond to each category:
  - Positive or negative, for sentiment
  - Sad, happy, angry, anxious... for emotions
  - Insight, causation, discrepancy, tentative... for cognitive processes
  - Sexism, homophobia, xenophobia, racism... for hate speech

  many others: see LIWC, VADER, SentiStrength, LexiCoder...
- Count number of times they appear in each document
- Normalize by document length (optional)
- **Validate, validate, validate**.
  - Check sensitivity of results to exclusion of specific words

Dictionary methods

Classifying documents when categories are known:

- Lists of words that correspond to each category:
  - Positive or negative, for sentiment
  - Sad, happy, angry, anxious... for emotions
  - Insight, causation, discrepancy, tentative... for cognitive processes
  - Sexism, homophobia, xenophobia, racism... for hate speech

    many others: see LIWC, VADER, SentiStrength, LexiCoder...
- Count number of times they appear in each document
- Normalize by document length (optional)
- **Validate, validate, validate**.
  - Check sensitivity of results to exclusion of specific words
  - Code a few documents manually and see if dictionary prediction aligns with human coding of document

Example: Linguistic Inquiry and Word Count

Linguistic Inquiry and Word Count

- Created by Pennebaker et al — see http://www.liwc.net

Linguistic Inquiry and Word Count

- Created by Pennebaker et al — see http://www.liwc.net
- Uses a dictionary to calculate the percentage of words in the text that match each of up to 82 language dimensions

Linguistic Inquiry and Word Count

- Created by Pennebaker et al — see http://www.liwc.net
- Uses a dictionary to calculate the percentage of words in the text that match each of up to 82 language dimensions
- Consists of about 4,500 words and word stems, each defining one or more word categories or subdictionaries

Linguistic Inquiry and Word Count

- Created by Pennebaker et al — see http://www.liwc.net
- Uses a dictionary to calculate the percentage of words in the text that match each of up to 82 language dimensions
- Consists of about 4,500 words and word stems, each defining one or more word categories or subdictionaries
- For example, the word *cried* is part of five word categories: sadness, negative emotion, overall affect, verb, and past tense verb. So observing the token *cried* causes each of these five subdictionary scale scores to be incremented

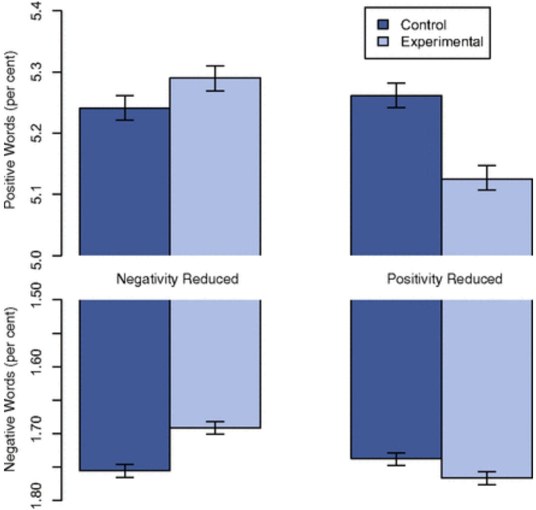
Linguistic Inquiry and Word Count

- Created by Pennebaker et al — see http://www.liwc.net
- Uses a dictionary to calculate the percentage of words in the text that match each of up to 82 language dimensions
- Consists of about 4,500 words and word stems, each defining one or more word categories or subdictionaries
- For example, the word *cried* is part of five word categories: sadness, negative emotion, overall affect, verb, and past tense verb. So observing the token *cried* causes each of these five subdictionary scale scores to be incremented
- Hierarchical: so "anger" are part of an *emotion* category and a *negative emotion* subcategory

Linguistic Inquiry and Word Count

- Created by Pennebaker et al — see http://www.liwc.net
- Uses a dictionary to calculate the percentage of words in the text that match each of up to 82 language dimensions
- Consists of about 4,500 words and word stems, each defining one or more word categories or subdictionaries
- For example, the word *cried* is part of five word categories: sadness, negative emotion, overall affect, verb, and past tense verb. So observing the token *cried* causes each of these five subdictionary scale scores to be incremented
- Hierarchical: so "anger" are part of an *emotion* category and a *negative emotion* subcategory
- You can **buy** it here: http://www.liwc.net/descriptiontable1.php

Example: Emotional Contagion on Facebook



Source: Kramer et al. 2014

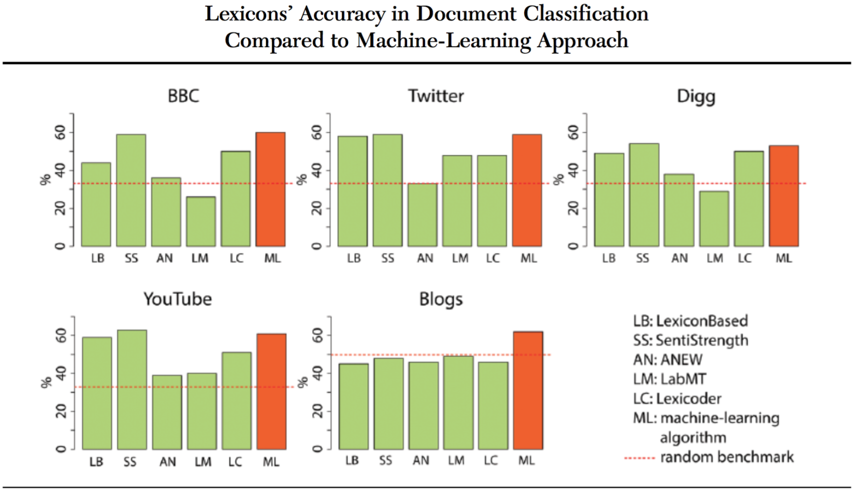
Potential advantage: Multi-lingual

APPENDIX B
DICTIONARY OF THE COMPUTER-BASED CONTENT ANALYSIS

	NL	UK	GE	IT
Core	elit*	elit*	elit*	elit*
	consensus*	consensus*	konsens*	consens*
	ondemocratisch*	undemocratic*	undemokratisch*	antidemocratic*
	ondemokratisch*			
	referend*	referend*	referend*	referend*
	corrupt*	corrupt*	korrupt*	corrot*
	propagand*	propagand*	propagand*	propagand*
	politici*	politici*	politiker*	politici*
	bedrog	*deceit*	täusch*	ingann*
	bedrieg	*deceiv*	betrüg*	
			betrug*	
	verraa	*betray*	*verrat*	tradi*
	verrad			
	schaam*	shame*	scham*	vergogn*
			schäm*	
	schand*	scandal*	skandal*	scandal*
	waarheid*	truth*	wahrheit*	verità
	oneerlijk*	dishonest*	unfair*	disonest*
			unehrlich*	
Context	establishm*	establishm*	establishm*	partitocrazia
	heersend*	ruling*	*herrsch*	
	capitul*			
	kapitul*			
	kaste*			
	leugen*		lüge*	menzogn*
	lieg*			mentir*

Source: Rooduijn and Pauwels 2011

Potential disadvantage: Context specific



Source: González-Bailón and Paltoglou (2015)

How to build a dictionary

How to build a dictionary

- The ideal content analysis dictionary associates all and only the relevant words to each category in a perfectly valid scheme
- Three key issues:

Validity	Is the dictionary's category scheme valid?
Recall	Does this dictionary identify *all* my content?
Precision	Does it identify *only* my content?

- Imagine two logical extremes of including all words (too sensitive), or just one word (too specific)

How to build a dictionary

1. Identify "extreme texts" with "known" positions. Examples:

How to build a dictionary

1. Identify “extreme texts” with “known” positions. Examples:
    - Tweets by populist vs mainstream parties (for populism dictionary)

How to build a dictionary

1. Identify “extreme texts” with “known” positions. Examples:
   - Tweets by populist vs mainstream parties (for populism dictionary)
   - Facebook comments to news about natural catastrophes vs football victories (for sentiment dictionary)

How to build a dictionary

1. Identify “extreme texts” with “known” positions. Examples:
   - Tweets by populist vs mainstream parties (for populism dictionary)
   - Facebook comments to news about natural catastrophes vs football victories (for sentiment dictionary)
   - Subreddits for white nationalist groups vs regular politics (for racist rhetoric)

How to build a dictionary

1. Identify “extreme texts” with “known” positions. Examples:
    - Tweets by populist vs mainstream parties (for populism dictionary)
    - Facebook comments to news about natural catastrophes vs football victories (for sentiment dictionary)
    - Subreddits for white nationalist groups vs regular politics (for racist rhetoric)
2. Search for differentially occurring words using word frequencies

How to build a dictionary

1. Identify "extreme texts" with "known" positions. Examples:
    - Tweets by populist vs mainstream parties (for populism dictionary)
    - Facebook comments to news about natural catastrophes vs football victories (for sentiment dictionary)
    - Subreddits for white nationalist groups vs regular politics (for racist rhetoric)
2. Search for differentially occurring words using word frequencies
3. Examine these words in context to check their precision and recall

How to build a dictionary

1. Identify “extreme texts” with “known” positions. Examples:
    - Tweets by populist vs mainstream parties (for populism dictionary)
    - Facebook comments to news about natural catastrophes vs football victories (for sentiment dictionary)
    - Subreddits for white nationalist groups vs regular politics (for racist rhetoric)
2. Search for differentially occurring words using word frequencies
3. Examine these words in context to check their precision and recall
4. Use regular expressions to see whether stemming or wildcarding is required