university of groningen

Quantitative Text Analysis – Essex Summer School

Dictionaries. Comparing documents

dr. Martijn Schoonvelde

University of Groningen

Today's class

- Dictionaries
- Document similarity
- Lab session
- Breakout groups

Analysis

- Three broad types of analysis (Boumans & Trilling 2016), from most deductive to most inductive:
  - **counting and dictionary methods**: the researcher can fully specify relevant features, and will categorise text accordingly
  - **supervised methods**: the researcher knows how to categorise documents, and uses machine learning methods to learn which features drive this categorisation
  - **unsupervised methods**: the researcher uses qta tools to learn about textual categorisation inductively

Dictionary methods

- "Dictionaries use the rate at which key words appear in a text to classify documents into categories or to measure the extent to which documents belong to particular categories" (Grimmer & Stewart, 2013)
  - That is, dictionaries count words associated with specific meanings
- Dictionaries consist of key-value pairs
  - Key: label for the concept or equivalence class
  - Values: (multiple) terms or patterns of terms that are declared equivalent occurrences of the key class

Dictionary methods

Under what conditions do dictionary methods excel? (Van Atteveldt *et al.* 2022)

- The categories that we want to code are manifest and concrete rather than latent and abstract
  - Multifaceted theoretical constructs such as frames or ideologies are difficult to capture using dictionaries alone
- All known synonyms are included in the dictionary
- Dictionary entries do not have multiple meanings

Dictionary methods

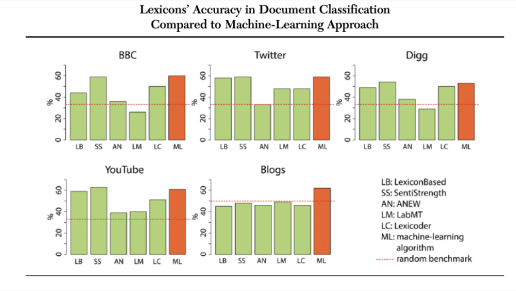
Key questions when applying dictionaries concern their validity, recall, and precision

- Validity – does the dictionary meaningfully operationalize our concept of interest?
- Recall – does the dictionary identify all relevant content?
- Precision – does the dictionary identify only relevant content?

Source: https://lse-my459.github.io/

Dictionary methods

Off-the-shelf dictionaries have a bit of a bad reputation. For example, lots of research that shows that sentiment dictionaries vary in their performance across types of text



Source: González Bailón & Paltoglu 2015

Issues with Off-the-Shelf Dictionary Methods

Domain-specificity problem

- Dictionaries are often created in one context and used in another, which can be problematic (Chan *et al.*, 2020; see also Rauh, 2018; Rice & Zorn, 2019).
  - "Honourable" in House of Commons speeches; "crude", "cost", "cancer" in earning reports (Loughran & McDonald, 2011)

Polysemy

- Words can have multiple meanings (polysemous words) depending on their use in a sentence. E.g., 'kind' as a noun, 'kind' as an adjective, 'kind' as an adverb

False negatives

- How can we be certain that a dictionary has captured all relevant synonyms in a corpus?

English bias

- Many dictionaries are constructed with an English bias. Do they translate effectively to other languages. But see Proksch *et al.* (2019) for a counterpoint.

Recent advancements

Recent studies have made progress on these issue by developing domain-specific dictionaries that rely on machine translation, word embeddings or extensive human coding (Müller, 2021; Proksch *et al.*, 2019; Rauh, 2018; Rheault *et al., 2016*; van Atteveldt *et al.*, 2008; Widmann, 2021).

Also: joint modeling of topics and sentiment – Joint Sentiment Topic model (JST) and the reversed Joint Sentiment Topic model (rJST) (Pipal *et al.* 2024; Lin *et al.*, 2009; Lin *et al.*, 2012).

Dictionary methods in quanteda

A dictionary in **quanteda** assigns possible features to categories, or "keys". It can have as many categories as you wish, and can be composed of any possible feature.

```
> txt_dfm <- corpus(c("this is excellent", "bad", "good, not horrible")) %>%
tokens() %>% dfm()
> sent_dict <- dictionary(list(positive=c("great","good","excellent"),
+                              negative=c("bad", "horrible", "badly")))
> dfm_dict <- dfm_lookup(txt_dfm, dictionary = sent_dict)
> dfm_dict
Document-feature matrix of: 3 documents, 2 features (33.33% sparse) and 0 docvars
       features
docs    positive negative
  text1        1        0
  text2        0        1
  text3        1        1
```

Sentiment dictionaries in quanteda

Through **quanteda.sentiment** you have access to a number of off-the-shelf sentiment dictionaries:

- Polarity dictionaries have two lists of words, each indicating one "pole" (by default "positive" and "negative")
- 'Valence dictionaries have continuous values/weights associated with each word in a given category, and may have more or fewer than two categories.

Name	Description	Polarity	Valence
data_dictionary_AFINN	Nielsen's (2011) 'new ANEW' valenced word list		✔
data_dictionary_ANEW	Affective Norms for English Words (ANEW)		✔
data_dictionary_geninqposneg	Augmented General Inquirer *Positiv* and *Negativ* dictionary	✔	
data_dictionary_HuLiu	Positive and negative words from Hu and Liu (2004)	✔	
data_dictionary_LoughranMcDonald	Loughran and McDonald Sentiment Word Lists	✔	
data_dictionary_LSD2015	Lexicoder Sentiment Dictionary (2015)	✔	
data_dictionary_NRC	NRC Word-Emotion Association Lexicon	✔	
data_dictionary_Rauh	Rauh's German Political Sentiment Dictionary	✔	
data_dictionary_sentiws	SentimentWortschatz (SentiWS)	✔	✔

Source: https://github.com/quanteda/quanteda.sentiment/

Do's When Using Off-the-Shelf Dictionaries

- **Read the dictionary**
  - Determine if it is domain-specific.
  - Modify the dictionary if necessary, but ensure transparency by reporting any changes.
- **Test on a subset of your corpus**
  - Check if the dictionary captures the specific construct you are studying.
  - Evaluate the dictionary's specificity (accuracy in identifying only the construct) and sensitivity (ability to detect all instances of the construct).
- **Document and report your methods**
  - Keep detailed records of how the dictionary was used and any modifications made.

Developing Dictionaries

- "When counting is automated, success or failure largely reflects a correct (excluding irrelevant data) and exhaustive (including all the relevant data)" (Boumans and Trilling 2016)
- As always, there is no silver bullet approach – much depends on what construct the dictionary is supposed to capture

Developing Dictionaries

Van Atteveldt *et al.* (2022) propose the following workflow for constructing a dictionary:

1. Construct a dictionary based on theoretical considerations and by closely reading a sample of example texts.
2. Code some articles manually and compare with the automated coding.
3. Improve your dictionary and check again.
4. Manually code a validation dataset of sufficient size. The required size depends a bit on how balanced your data is – if one code occurs very infrequently, you will need more data.
5. Calculate the agreement.
    - Precision and recall

Developing Dictionaries

1. Identify "extreme texts" with "known" categories / positions
   - Opposition leader and Prime Minister in a no-confidence debate
   - Five-star review of a product (excellent) and a one-star review (terrible)
2. Search for differentially occurring words using word frequencies
3. Examine these words in context to check their sensitivity and specificity
4. Examine inflected forms to see whether stemming or wildcarding is required
5. Use these words (or their lemmas) for categories

`Source: https://lse-my459.github.io/`

Assessing validity

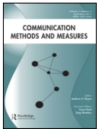
Content validation

- Check sensitivity of results to exclusion of specific words.
- Disambiguate meaning. For example, check grammatical function of ambiguous words using POS tagging.

Concept validation

- Code a few documents manually and see if dictionary prediction aligns with human coding of document (check precision and recall)
- Check if categorisation behaves in predictable ways. For example, do newspaper articles about the economy peak during periods of economic uncertainty

Further reading



Communication Methods and Measures

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/hcms20

The Validity of Sentiment Analysis:Comparing Manual Annotation, Crowd-Coding, Dictionary Approaches, and Machine Learning Algorithms

Wouter van Atteveldt , Mariken A. C. G. van der Velden & Mark Boukes

To cite this article: Wouter van Atteveldt , Mariken A. C. G. van der Velden & Mark Boukes (2021): The Validity of Sentiment Analysis:Comparing Manual Annotation, Crowd-Coding, Dictionary Approaches, and Machine Learning Algorithms, Communication Methods and Measures, DOI: 10.1080/19312458.2020.1869198

To link to this article: https://doi.org/10.1080/19312458.2020.1869198

PA

Automated Text Classification of News Articles: A Practical Guide

Pablo Barberá[1], Amber E. Boydstun[2], Suzanna Linn[3], Ryan McMahon[4] and Jonathan Nagler[5]

[1] Associate Professor of Political Science and International Relations, University of Southern California, Los Angeles, CA 90089, USA. Email: pbarbera@usc.edu
[2] Associate Professor of Political Science, University of California, Davis, CA 95616, USA. Email: aboydstun@ucdavis.edu
[3] Liberal Arts Professor of Political Science, Department of Political Science, Penn State University, University Park, PA 16802, USA. Email: sll8@psu.edu
[4] PhD Graduate, Department of Political Science, Penn State University, University Park, PA 16802, USA (now at Google). Email: mcmahon.rb@gmail.com
[5] Professor of Politics and co-Director of the Center for Social Media and Politics, New York University, New York, NY 10012, USA. Email: jonathan.nagler@nyu.edu

Abstract

Automated text analysis methods have made possible the classification of large corpora of text by measures such as topic and tone. Here, we provide a guide to help researchers navigate the consequential decisions they need to make before any measure can be produced from the text. We consider, both theoretically and empirically, the effects of such choices using as a running example efforts to measure the tone of *New York Times* coverage of the economy. We show that two reasonable approaches to corpus selection yield radically different corpora and we advocate for the use of keyword searches rather than predefined subject categories provided by news archives. We demonstrate the benefits of coding using article segments instead of sentences as units of analysis. We show that, given a fixed number of codings, it is better to increase the number of unique documents coded rather than the number of coders for each document. Finally, we find that supervised machine learning algorithms outperform dictionaries on a number of criteria. Overall, we intend this guide to serve as a reminder to analysts that thoughtfulness and human validation are key to text-as-data methods, particularly in an age when it is all too easy to computationally classify texts without attending to the methodological choices therein.

Keywords: statistical analysis of texts, automated content analysis, content analysis

Document similarity and document distance

A frequent challenge in QTA is comparing pairs of documents and assessing how close or similar they are to one another.

- Constitutional scholars may want to know which constitutions are most alike.
- Communication scholars may want to know how news travels through outlets.
- Political scientists may want to compare bills to laws

To this end it helps to think of documents as a vector of features as it allows us to use similarity metrics from linear algebra

- **Vector space model** – a document's vector is its numerical representation in a document feature matrix

```
                features
docs             so now , on this hallowed ground where just days
  2021-Biden.12  1   1 5  1    2        1      1     1    1    1
```

Biden inaugural address

"So now, on this hallowed ground where just days ago violence sought to shake this Capitol's very foundation, we come together as one nation, under God, indivisible, to carry out the peaceful transfer of power as we have for more than two centuries." (Biden, 2021)

Boris Johnson resignation speech

"Being prime minister is an education in itself. I have traveled to every part of the United Kingdom and, in addition to the beauty of our natural world, I have found so many people possessed of such boundless British originality.." (Johnson, 2022)

Leo Varadkar resignation speech

"I have learned so much about so many things, met people who I would never have got to meet, been to places I would never have seen, both home and abroad." (Varadkar, 2024)

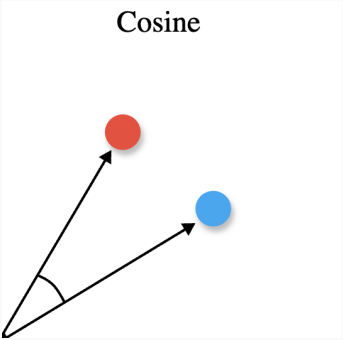
Cosine Similarity

Cosine Similarity measures the cosine of the angle between two vectors in a multi-dimensional space.

- Each document is represented as a vector.
- The cosine of the angle between these vectors indicates their similarity.

Pros and cons:

- Captures the orientation (direction) of vectors, not their magnitude.
- Effective for high-dimensional data like text.



Source: https://www.maartengrootendorst.com

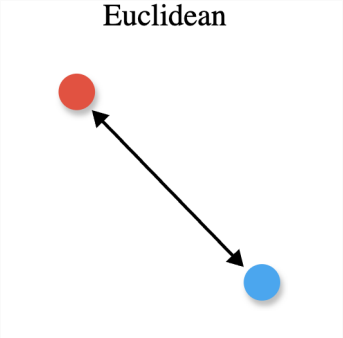
Euclidean Distance

Euclidean Distance measures the straight-line distance between two points (vectors) in a multi-dimensional space.

- Distance is calculated as the square root of the sum of the squared differences between corresponding vector elements.

Pros and cons:

- Simple and intuitive.
- Useful when document vectors are of the same length; but sensitive to scaling



Source: https://www.maartengrootendorst.com

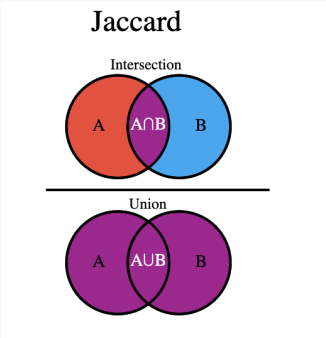
Jaccard Similarity

Jaccard similarity is another similarity measure. It's fairly easy to calculate:

- Count the number of types that appear in both documents (intersection)
- Count the number of types that appear in either document (union)
- Divide the first by the second (intersection / union)

Pros and cons:

- Effective for comparing the similarity of text with distinct term sets.
- Ignores the frequency of terms, focusing on presence/absence.



Source: https://www.maartengrootendorst.com

Cosine similarity in quanteda

```
> similarity_cosine <- textstat_simil(dfm,
+                                     method = "cosine",
+                                     margin = "documents")
> similarity_cosine
textstat_simil object; method = "cosine"
        Joe Boris   Leo
Joe   1.000 0.455 0.432
Boris 0.455 1.000 0.533
Leo   0.432 0.533 1.000
```

Euclidian distance in quanteda

```
>  similarity_euclidian <- textstat_dist(dfm,
+                                       method = "euclidean",
+                                       margin = "documents")
> similarity_euclidian
textstat_dist object; method = "euclidean"
       Joe Boris  Leo
Joe      0  9.33 9.38
Boris 9.33     0 8.06
Leo   9.38  8.06    0
```

Jaccard similarity in quanteda

```
> similarity_jaccard <- textstat_simil(dfm,
+                                      method = "jaccard",
+                                      margin = "documents")
> similarity_jaccard
textstat_simil object; method = "jaccard"
         Joe  Boris    Leo
Joe    1.0000 0.0882 0.0667
Boris 0.0882 1.0000 0.1600
Leo    0.0667 0.1600 1.0000
```

Similarity and distance measures in quanteda

textstat_dist options are: "euclidean" (default), "canberra", "Chisquared", "Chisquared2", "hamming", "kullback". "manhattan", "maximum", "canberra", and "minkowski".
textstat_simil options are: "correlation" (default), "cosine", "jaccard", "eJaccard", "dice", "eDice", "simple matching", "hamann", and " "faith".

Similarity in social science

Similarity and distance measures are blind to the semantic content of a text.

As social scientists we often have a specific idea in mind when we are interested in similarity between documents (e.g., the extent to which they share certain topics, or the extent to which they share a particular sentiment)

- If we want to try and measure context-specific similarity, other tools and methods are probably better
- But similarity and distance scores may help us in the conceptualization stage