Lecture 2

1. Why translate text to numbers?

* Can process large volumes of data.
* Analysis can be systematic, based on explicit sampling procedure and enabling replication.
* Quantification enables use of statistical tools.
* The most important part is the research question.

1. Content analysis

* Collecting textual data and turning it into data so we can understand something about the world.
* It is in-between quantitative and qualitative.
* Other techniques exists (thematic analysis, discourse analysis, etc.) but often the focus is on other aspects – like in inter-textual relationships.
* Unitization
  + The units of texts that we are going to sample and analyse.
  + Coding units, i.e. a sentence or a paragraph.
  + Context units – how much we take into account when analysing the content of the text.
* Research interest as the driving force
  + Is our scheme theoretically derived (deductive), or are we exploring the material with no theoretical priors (inductive)?
  + Example -the position of states towards the causes of (and likely solutions to) war in Ukraine:
    - War results from Russia expansionism, it is an unprovoked aggression.
    - War results from Western expansionism and unilateralism, its threats to Russia.
* Coding scheme
  + List the coding categories.
  + Can be hierarchically structured.
  + Defines rules for assignment into categories or on scales.
  + Embodied in a formalized/written codebook.
* Document feature matrix – page 15 of the PowerPoint.

1. Validity and reliability of coding

* Validity
  + Face validity – being obviously true, sensible, plausible.
  + Empirical validity – compare how the analysis performs against some standards.
* Reliability
  + If I apply the same method repeatedly whether or not you will continue to get the same result when you retest.
  + Stability – the example he gave was you try the code again a couple of months later when you don’t have several coders, so you retest your initial code.
  + Table with the different types of reliability (PowerPoint p. 21).
  + Accuracy – test-standard – intraobserver inconsistencies, and interobserver disagreements and the deviations from a standard.
  + Reproducibility – test-test – intraobserver inconsistencies, interobserver disagreements.
  + Having reliability doesn’t necessarily mean you have validity.
    - Because you can increase reliability of the coders by changing the coding scheme, but that can lead to decrease the validity.
  + Unreliability limits the change of validity,
* Evaluating reliability: Krippendorf alpha
  + Measure of reliability between coders that takes into account random chance.
  + Even random number generators can agree by chance.

1. Automating text lability

* When it comes to computers the problem is not with reliability it is with validity.
* Need to judge what we are getting from the computer to see what we are getting – have a human look at what the computer produces.
* The authors in p. 32 argue that it may still be wrong considering that we are trying to use a simple thing to look at a very complex thing (human language).
* Confusion matrix – p. 35
  + Accuracy – the share of correct predictions, bad for when you gave unbalanced class membership (can have 90% accuracy in detecting spam but in reality you just didn’t get that much spam).
  + Precision – how many of the cases that we retrieved as having that property do actually have that property.
    - How many spam emails we collected are actually spam.
    - True positive / (true positive + false positive)
  + Recall – true positive / (true positive + false negative)
    - How many relevant items are retrieved.
* F-1 score
  + Harmonic mean of precision and recall.
  + For F1 to be high, both precision and recall must be high simultaneously.
  + F1 = 2 \* ((precision \* recall) / (precision + recall))
* Metrics for multilabel data (3+ categories)
  + Metric calculated for each category separately and averaged
    - Macro: simple, un-weighted average of category scores.
    - Micro: sum all true positives, false positives, true negatives together, … (can be dominant by some categories).

Lecture 3

1. Quanteda

* Integrated workflow for text-as-data analysis in R.
  + Corpus – the text files and the contextual variables (doc-level).
    - Document variables – date of publication, authors, if it was written, if it was a speech.
  + Quanteda holds in the corpus the text files and the contextual variables.
  + From the corpus we can create tokens, which are some specific parts of text that are usually defined as words, but it can be sentences/paragraphs sometimes.
  + The cleaning of the corpus is through the tokens.
  + Document feature metrics (DFM) – documents are stored in rows and features in column.
    - Then you can see how many times a specific feature is present in the documents.
  + Non-positional analysis (bag-of-words), this is that the position of words in the sentence doesn’t matter.
  + Can also do positional analysis which takes into account the position of the words in the sentence.
* Pre-processing
  + Stopwords – remove non-informative words (and, or, to, be the, numbers, symbols, etc.)
  + To lower – conversion of capital letters to lowercase
  + Stemming – this compress the vocabulary for the corpus analysis.
* Tokenization N-Grams
  + N-grams – sequence of tokens from already tokenized texts.
* DFM
  + In the code.

1. Descriptive analysis

* Now that you have all the things you did before with Quanteda you can actually start analysing the text.
* TF-IDF (term frequency-inverse document frequency)
  + It is used to find the things that are most important in the corpus – it rates most unique words in the corpus.
  + TF = (number of times word appears in document) / (total number of words in document).
  + IDF = log(Total number of documents / number of documents containing the word).
  + Falta.
  + It tris to reduce the impact of common words, like stop words.
  + Helps to increase the importance of rare words, the ones that are more relevant to the analysis.
* Falta.
* Lexical Diversity
  + Measured on a bunch of indicators.
  + The most simple one is type token ration (TTR) – calculates the ratio of unique word types to the total number of tokens (words) in a text.
  + Formula, page 30.
  + A higher TTR indicates greater lexical diversity, meaning that a text uses a wider range of words and phrases.
  + One of the flaws can be that the language in longer is not that diverse, because there are only so many words you can use each language so you end up repeating some of them.
* Document similarity
  + Measures the similarity between two vectors in a multi-dimensional space.
  + In machine learning the vectors are a lot more advanced and complicated than in bag-of-words.
  + Cosine similarity is widely used in many applications
    - Falta.

Lecture 4

1. Falta

* Falta.
* Mediated politics
  + Falta
* Falta
* Falta something about the different websites we can use for getting newspapers and stuff from the internet – different databases.
* Falta
* Event data: POLECAT
  + Is a project

1. API and RSS feeds

* APIs
  + Structured access to the data, that then you use to query the data and get the data.
  + Super easy if it exists – you have the loop that calls things and then that is it.
  + The code he gives for API is just making the df to format.
* RSS feed
  + Just spit the information into the world.
  + Tidyfeed() – very important function to get everything you need, easier way to deal with the getting and stuff.

1. Scrapping