Contents

1 Computer-Assisted Text Analysis for Comparative Politics

2 Introduction

- Fokus auf Tools für Komparatisten, um *textual* data zu nutzen
- Hervorhebung des *unsupervised topic modeling*
- Verwendung des Structural Topic Model um das Potential von Topic Modeling für vergleichende Politik aufzuzeigen
  - STM erlaubt Rückschlüsse auf Beziehung zw Metadaten und Textkorpus
- wie unterscheidet sich Textanalyse und Text Processing in versch Sprachen
  - R package ‘translateR‘

3 Text and Language Basics

3.1 Research Questions and Data Analysis

- automatische Inhaltsanalyse und vergleichende Politik sind eine gute Kombination
- Länder produzieren Texte in noch nie dagewesenem Umfang
- traditionelle Regierungsstatistiken sind häufig nicht vorhanden, unvollständig, manipuliert oder falsch gemessen
  - Regierungen produzieren allerdings große Mengen an Textdaten, welche für deskriptive und kausale Inferenzen genutzt werden können
  - Anreiz für Gelehrte andere Formen von Data zu verwenden
- Gelehrte der vergleichenden Regierungslehre/Politik verwenden bereits automatische Methoden für Textanalysen
  - weitverbreiteste Form von Text zu Politiker sind wahrscheinlich Aufzeichnungen von Reden oder anderen Statements
- Auflistung einiger interessanten Studien die automatische Textanalyse ultilisiert haben

3.2 Text Processing Basics: A Multilanguage View

- Analytiker muss zuerst sicherstellen das zu analysierender Text maschinell lesbar ist
  - statistische Methoden der Textanalyse sind meist unabhängig von der Sprache
    - aber Tools des Prepocessings *nicht*

3 Herausforderungen bei der Arbeit mit verschiedenen Sprachen:

1. Umgang mit Zeichenkodierung (dealing with encodings)
2. Präprozessing zur Reduktion der Dimensionalität
3. Umgang mit großen Korpora

3.3 Umgang mit Encodings

- Sprachen können unterschiedliche Zeichenkodierung haben und unterschiedliche Computer händeln dies auf underschiedliche Art & Weise
  - unterschiedliche default encodings
- wenn der Analyst Daten aus versch Quellen bezieht ist es von nöten, dass das Encoding angepasst wird, sodass es in allen Dokumenten gleich ist
  - anschließend muss sichergestellt werden, dass die Software die Zeichenkodierung korrekt liest

3.4 Prepocessing to extract the most information

3.4.1 Stopword removal

- Entfernung von Worten die extrem häufig auftreten aber nicht relevant im Bezug auf das Erkenntnisinteresse sind (zb "and", "the", "und", "zum")
  - viele Sprachen haben eine Liste üblicher stop words
- eine Liste von stop words die entfernt werden, sollte sorgfältig gewählt werden, da unterschiedliche stop words zu unterschiedlichen Ergebnissen führen können und manchmal im Kontext entscheidend sein können

3.4.2 Stemming & lemmatization

Stemming:

- Entfernung der Enden von konjugierten Verben oder Nomen in der Pluralform, so dass nur der "Stamm" überbleibt
- nützlich in jeder Sprache in der das Ende von Worten geändert wird für eine Veränderung der Zeit oder Anzahl (Englisch, Spanisch, Französisch etc)
- nicht in jeder Sprache nötig/nützlich
  - chinesische Verben werden zB nicht konjugiert und Nomen in chinesisch werden nicht durch eine Endung pluralisiert
- Nützlichkeit ist anwendungs- und sprachabhängig

- Stemming ist ein Vefahren/Näherung an ein allgemeineres Ziel was als Lemmatization (Lemmatisierung) bezeichnet wird

Lemmatization:

- Identifikation der Grundform eines Wortes und Gruppierung dieser Worte
- komplexer Algorithmus, der nicht einfach das Ende eines Wortes abschneidet, sondern die Herkunft des Wortes identifiziert und nur das Lemma (Grundform) des Wortes zurück gibt
- kann außerdem Kontext schlussfolgern:
  - zB "saw" als Nomen = "Säge" bleibt so, als Verb = "sah" wird zu → "sehen/see"
- für Englisch funktioniert Stemming fast so gut wie Lemmatization in anderen Sprachen wie zB Koreanisch oder Türkisch ist Lemmatization hilfreicher

3.4.3 Compound words

- einige (compound) Sprachen setzen oft Worte zusammen (compounding) um ein neues Wort zu bilden zB Kirche + Rat = Kirchenrat
  - decompounding macht in diesem Fall keinen Sinn da die Worte zusammengehören
- in decompoung Sprachen widerrum können *mehrere* getrennte Worte zu *einem* Konzept gehören:
  - "social security" und "national security"
    - beide enthalten "security" aber haben trotzdem unterschiedliche Bedeutung, daher möchte der Analyst die Worte evtl. compounden (zusammenführen), zu "nationalsecurity" und "socialsecurity", um die Bedeutung an *ein* Wort zu koppeln

3.4.4 Segmentation

- einige Sprachen wie zB Chinesisch werden nicht durch Leerzeichen segmentiert und erfordern deshalb automatische Segmentierung bevor sie von einem Statistikprogramm weiterverarbeitet werden können

3.5 Building the document-term matrix

- nach dem das Prepocessing abgeschlossen ist, werden die übrig gebliebenen Worte genutzt, um eine document-term matrix (DTM) zu konstruieren
- in einer document-term matrix repräsentiert jede Reihe ein Dokument und jede Spalte ein ein einzigartiges Wort
  - jede Zelle enthält die Anzahl des Auftreten des jeweiligen Wortes (Spalte) im jeweiligen Dokument (Reihe)
    - üblicherweise enthalten viele Zellen eine 0

Beispiel:

Berlin	Brüssel	Merkel	Schulz
0	1	0	1
1	0	1	0

- Reihenfolge der Worte beachtet die DTM nicht
- da diese DTM schon bei Korpora moderater Größe sehr groß werden kann, ist es ratsam nur Einträge zu speichern die nicht 0 sind (sparse representation)
- die DTM oder ihre sparse representation sind der primäre Input für automatische Textanalysemethoden

3.6 Multilanguage Preprocessing Tools

Language-specific processing

- das R Package ‘tm‘ kann Stemming für 11 Sprachen betreiben

4 Computer-Assisted Text Analysis