Computational Linguistics, Corpora, Counting Words

Sprachverarbeitung (VL + Ü)

Nils Reiter

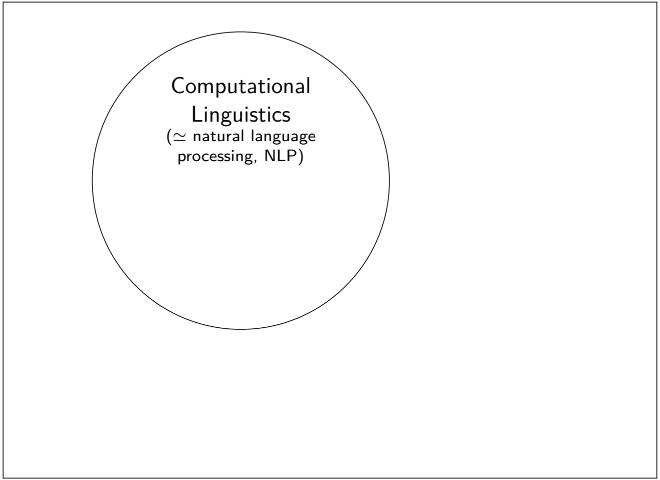
April 6, 2023

INSTITUT FÜR
DIGITAL HUMANITIES
UNIVERSITÄT ZU KÖLN

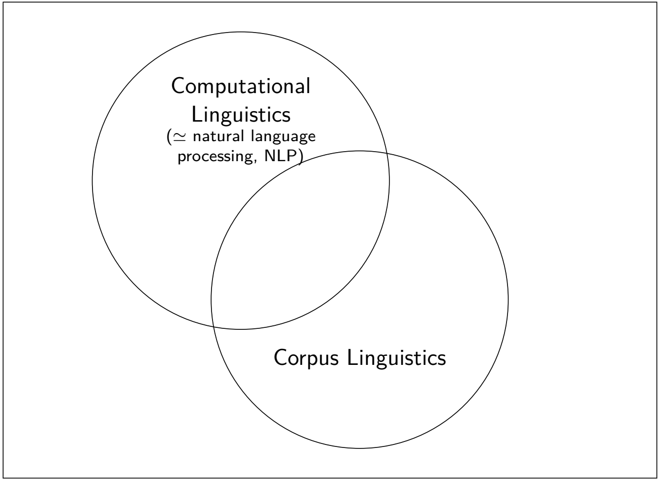
Section 1

Computational Linguistics

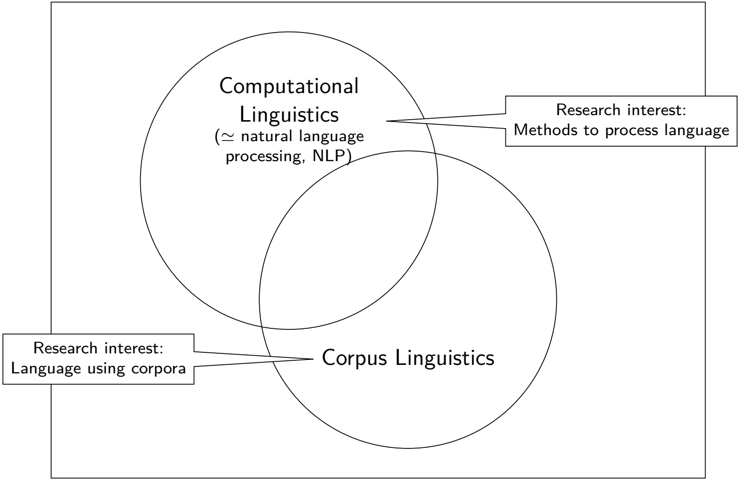
Disciplinary Placement



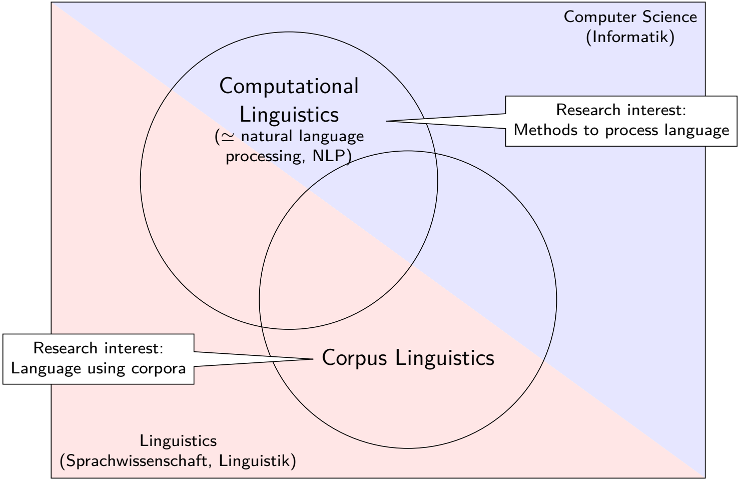
Disciplinary Placement



Disciplinary Placement



Disciplinary Placement



Brief history of Computational Linguistics I

- 1933: Russian engineer Troyanskii gets a patent on a mechanical translation device
  Hutchins/Lovtskii (2000)
- 1950s: DARPA Projects to automatically translate Russian into English
- 1957/65: Linguistics shifts focus from describing to generating Chomsky (1957, 1965)
- 1959: Theo Lutz for the first time generates a German poem with a computer
  Bernhart (2020); Lutz (1959)
- 1962: Foundation of the »Association for Machine Translation and Computational Linguistics«, 1968 renamed to »Association for Computational Linguistics (ACL)«
- 1966, ALPAC report: MT more expensive, less accurate and slower than human translation ALPAC (1966)
- 1968: Foundation of SYSTRAN, first MT company
- 1975: European commission uses SYSTRAN software (first use of MT on EU level)

Brief history of Computational Linguistics II

- 1984: First corpus-based commercial MT system Nagao (1984)
- 1992: Study programs established in Germany (Saarbrücken/Stuttgart)
- 2011: IBM Watson beats two humans in Jeopardy YouTube / Apples Siri launched
- 2013: Word embeddings (e.g., word2vec) Mikolov et al. (2013)
- 2017: Launch of the DeepL Translator (a Cologne-based company)
- 2018: Transformer models: BERT Devlin et al. (2019)
- 2022: ChatGPT chat.openai.com
  - ⚠ Yes, we need to talk about ChatGPT 🡇

Computational Linguistics

Today

- It's an interesting time to do CL
- For a long time: Fundamental Research, and real applications are far in the future
- Huge changes in the past 10 years: CL methods are now used everyday by everyone
  - This changes how research should be done (e.g., ethical considerations)
- ChatGPT (and other applications) raise expectation, that language processing is a solved problem

Computational Linguistics

Today

- It's an interesting time to do CL
- For a long time: Fundamental Research, and real applications are far in the future
- Huge changes in the past 10 years: CL methods are now used everyday by everyone
  - This changes how research should be done (e.g., ethical considerations)
- ChatGPT (and other applications) raise expectation, that language processing is a solved problem

Discussion Panel: 29.06.2023, 17:45–19:15

Are we done? Computational Linguistics between linguistics, digital humanities and large language models

- Berenike Herrmann (Digital Humanities, Bielefeld University)
- Mark Finlayson (CL, Florida International University, Miami)
- Klaus von Heusinger (Linguistics, University of Cologne)

Digital Humanities and Computational Linguistics

- Digital Humanities, broadly: Working with ›digital methods‹ on humanities subjects
- Linguistics: Study of language
- Computational Linguistics: Pioneer DH area Reiter (2014, 4)
  - … but this is a minority position in CL, often also seen as part of AI

Digital Humanities and Computational Linguistics

- Digital Humanities, broadly: Working with ›digital methods‹ on humanities subjects
- Linguistics: Study of language
- Computational Linguistics: Pioneer DH area — Reiter (2014, 4)
  - … but this is a minority position in CL, often also seen as part of AI
  - Historically (and still today) split between engineering (natural language processing, NLP) and science/scholarship (computational linguistics, CL)
  - ⚠ Neurolinguistic programming and natural language processing are not the same (both use ›NLP‹ as abbreviation)

Digital Humanities and Computational Linguistics

- Digital Humanities, broadly: Working with ›digital methods‹ on humanities subjects
- Linguistics: Study of language
- Computational Linguistics: Pioneer DH area Reiter (2014, 4)
  - … but this is a minority position in CL, often also seen as part of AI
  - Historically (and still today) split between engineering (natural language processing, NLP) and science/scholarship (computational linguistics, CL)
  - ⚠ Neurolinguistic programming and natural language processing are not the same (both use ›NLP‹ as abbreviation)

University of Cologne

For historic reasons, CL and NLP are called »Sprachliche Informationsverarbeitung«

Experiments

- Cornerstone of the ›scientific method‹
- Used in many disciplines: Natural sciences, social sciences, medicine, …

Experiments

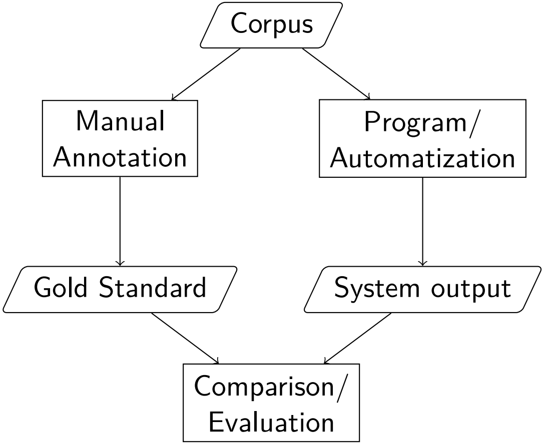
- Cornerstone of the ›scientific method‹
- Used in many disciplines: Natural sciences, social sciences, medicine, …
- Experiments are used to verify or falsify hypotheses
- Reproducibility: The outcome does not depend on the experimenter

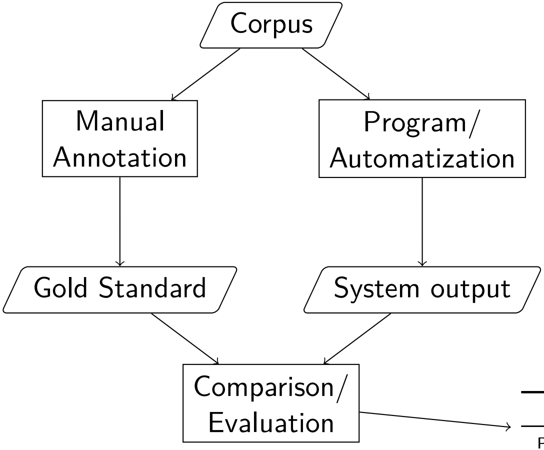
Experiments

- Cornerstone of the ›scientific method‹
- Used in many disciplines: Natural sciences, social sciences, medicine, ...
- Experiments are used to verify or falsify hypotheses
- Reproducibility: The outcome does not depend on the experimenter
- CL: Hypotheses about the operationalisation of language/text phenomena

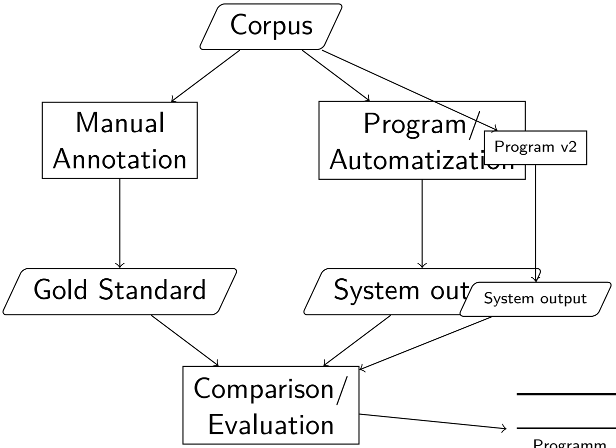
Example

Position within a sentence is indicative for the part of speech

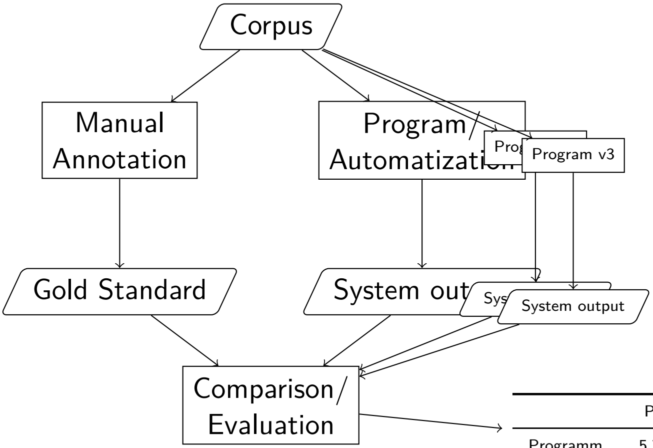

Corpus
Manual Annotation
Program/ Automatization
Gold Standard
System output
Comparison/ Evaluation



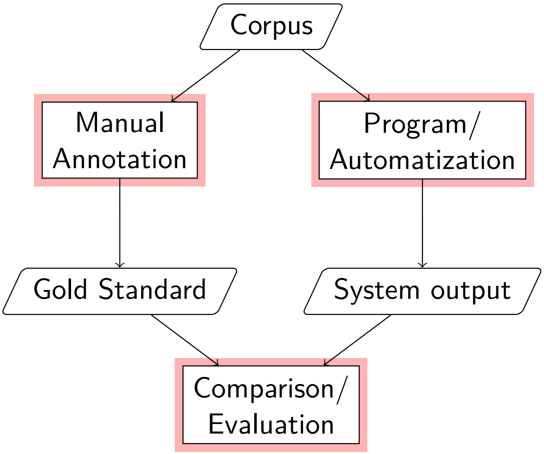
	P	R	F
Programm	5.7	9.2	...



	P	R	F
Programm	5.7	9.2	...
v2	9.9	16.7	...



	P	R	F
Programm	5.7	9.2	...
v2	9.9	16.7	...
v3	15.3	21.8	...


Corpus
Manual Annotation
Program/ Automatization
Gold Standard
System output
Comparison/ Evaluation

Checkliste zu NLP-Experimenten zur Klassifikation

Stand: 20. Dezember 2022

Hinweise

- Wenn Ihr Task kein Klassifikationstask ist, dann ist dieser Fragebogen nicht für Sie.
  - Keine Klassifikationstasks sind z.B. Generierung oder Übersetzung.
- Markieren Sie alle Punkte die Sie planen umzusetzen.
- Machen Sie einen Termin in der Sprechstunde, wenn Ihnen Punkte unklar sind.
- Der Fragebogen ist keine Prüfung, sondern dient als Hilfestellung, bei der Experimentplanung an alles zu denken, auf Ideen zu kommen und ggf. die richtigen Fragen zu stellen. Diese klären wir dann idealerweise im Gespräch.
- Der Fragebogen repräsentiert eine Planung. Gewisse Abweichungen von der Planung sind normal und zu erwarten.
- Bei Fragen schreiben Sie gerne eine E-Mail an `nils.reiter@uni-koeln.de` oder melden Sie sich gerne zu einer Sprechstunde an. Wenn Sie rein technische Fragen zur Infrastruktur haben, schreiben Sie bitte an `spinfo-admin@uni-koeln.de`.

Der Task

1. Die Aufgabe heißt: ____________________.
2. Es handelt sich um ☐ Textklassifikation, ☐ Sequence Labeling, oder ☐ Sonstiges: __________.
3. Die zu klassifizierenden Instanzen sind: ____________________.
4. Es gibt ______ Kategorien/Klassen.
5. Einer Instanz kann ☐ genau eine oder ☐ mehrere Klassen zugewiesen werden.

Die Daten

1. Annotierte Daten ☐ liegen bereits vor oder ☐ müssen noch erstellt werden.
2. In den Daten sind ______ Instanzen (von o.g. Typ) annotiert.
3. Die Klassen sind
   - ☐ gleichverteilt (d.h. jede Klasse ist ungefähr gleich häufig)
   - ☐ unterschiedlich verteilt, und zwar: __________

Die Annotationen

Nur relevant, wenn neue Daten annotiert werden sollen. (Frage Task.1)

1. Annotationsrichtlinien
   - ☐ Ich verwende die folgenden, bereits existierenden Annotationsrichtlinien: __________.
     - Mit diesen wurde ein Inter-Annotator-Agreement von ___ erzielt (Metrik: ________).
   - ☐ Ich schreibe neue Annotationsrichtlinien.

2. Annotator:innen
   - ☐ Ich annotiere selbst.
   - ☐ Ich rekrutiere Annotator:innen aus meinem Freundes-/Bekanntenkreis.
   - ☐ Ich sammle Annotationen über eine Umfrage, z.B. mittels LimeSurvey.
   - ☐ Ich sammle Annotationen über crowd sourcing.
3. Annotationsworkflow
   - ☐ Annotator:innen treffen eine Annotationsentscheidung auf der Basis eines Kontextes von ___ ☐ Wörtern, ☐ Sätzen, ☐ Zeilen, ☐ Absätzen, ☐ ____________________; oder ☐ sie verwenden den gesamten Text als Kontext.
   - ☐ Sie können dabei außerdem die folgenden Wissensquellen verwenden: ☐ Wikipedia, ☐ Lexika, ☐ Wörterbücher
4. Anforderungen an Annotationssoftware
   - ☐ Annotator:innen müssen Spannen selbst markieren können.
   - ☐ Annotator:innen müssen neue Kategorien oder Labels ergänzen können.

Die Baseline

- ☐ Weil die Klassen ungleich verteilt sind, bietet sich eine majority baseline an.
  Diese erzielt eine Accuracy von ______ %.
- ☐ Weil die Klassen gleich verteilt sind, bietet sich eine random baseline an.
  Diese erzielt eine Accuracy von ______ %.
- ☐ Eine weitere mögliche Baseline ist: ____________________.
  Diese erzielt eine Accuracy von ______%.
- ☐ Eine weitere mögliche Baseline ist: ____________________.
  Diese erzielt eine Accuracy von ______%.

Das Experiment

1. Ich möchte das folgende oder die folgenden Verfahren verwenden:
   - ☐ Entscheidungsbaum / Decision Tree (DT)
   - ☐ Naive Bayes
   - ☐ Support Vector Machines (SVM)
   - ☐ Logistic Regression
   - ☐ Neural Networks (NN)
     - ☐ Feed-Forward Neural Networks
     - ☐ Convolutional Neural Networks
     - ☐ Recurrent Neural Networks
     - ☐ Transformer-Architektur (BERT & co.)
   - ☐ Sonstige: __________
2. Ich möchte die folgenden Features verwenden
   - ☐ Metadaten: ____________________
   - ☐ Inhaltsdaten, z.B. aus Texten:
     - ☐ Worthäufigkeiten (von allen Wörtern), auch bekannt als bag of words

     - ☐ Häufigkeiten von Wörtern aus folgenden Wortlisten: __________
     - ☐ Embeddings (z.B. Word Embeddings)
     - ☐ Sequenzielle Information (d.h. Klassifikationsergebnisse für Elemente davor oder danach)
     - ☐ N-Gram-Häufigkeiten, mit $N \leq$ ______
     - ☐ Thematische Informationen aus einem Topic Model (z.B. Latent Dirichlet Allocation, LDA)
3. Meine Features haben die folgenden Datentypen:
   - ☐ Numerisch: ______ (Anzahl Features) ☐ Kategorial: ______ (Anzahl Features)
4. Testdaten
   - ☐ Ich teile meinen o.g. Datensatz selbst in Trainigs- und Testdaten auf. ______% der Instanzen werden als Trainingsdaten verwendet.
   - ☐ Ich verwende N-fold cross validation, mit $N =$ ______.
   - ☐ Trainings- und Testdaten sind bereits aufgeteilt, z.B. weil es Daten aus einem shared task sind.
5. Ich variiere und vergleiche
   - ☐ die Größe des Trainingsdatensatzes (z.B. 100, 1000, 10 000 Instanzen für den Trainingsdatensatz)
   - ☐ die Menge an oder Art von Features die verwendet werden (z.B. inhaltliche vs. sprachliche Features)
   - ☐ das Verfahren als solches oder Parameter davon (z.B. NN vs. SVM)
   - ☐ die Vorverarbeitung (z.B. Groß- und Kleinschreibung)
6. Meine Hypothese ist: ____________________

Die Auswertung und Evaluation

1. Ich verwende die Evaluationsmetrik(en)
   ☐ Accuracy ☐ Precision ☐ Recall ☐ F-Measure ☐ Area under curve (AUC)
   ☐ Sonstige: ____________________

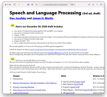
- ☐ Meine Testdaten sind stark unbalanciert (Frage Daten.3), daher verwende ich die Metriken in der Mikro- und Makro-Average-Variante.

2. Für meine Fehleranalyse inspiziere ich ______ Instanzen manuell.

Die praktische Umsetzung

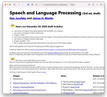
1. Ich verwende die Programmiersprache
   - ☐ Python
   - ☐ Java
   - ☐ R
   - ☐ __________
2. Hardware-Ausstattung und Vorkenntnisse
   - ☐ Ich verfüge über einen Computer
     - ☐ der auch mal über Nacht durchlaufen kann, wenn eine Berechnung etwas länger dauert.
     - ☐ der eine GPU mit CUDA-Unterstützung hat oder ein Mac mit M1/M2-Prozessor ist.
     - ☐ der ausreichend freien Plattenspeicher hat.
   - ☐ Ich möchte Berechnungen auf einem Server der Universität laufen lassen.
     - ☐ Ich kann mich per SSH auf einem Server einloggen.
     - ☐ Ich weiß wie ich auf einer Kommandozeile ein Programm laufen lasse.

Literature

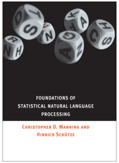


Dan Jurafsky/James H. Martin (2023). *Speech and Language Processing*. 3rd ed. Draft of Janaury 7, 2023. Prentice Hall. URL: https://web.stanford.edu/~jurafsky/slp3/ JM23

Literature

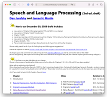


Dan Jurafsky/James H. Martin (2023). *Speech and Language Processing*. 3rd ed. Draft of Janaury 7, 2023. Prentice Hall. URL: https://web.stanford.edu/~jurafsky/slp3/ JM23

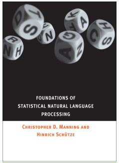


Christopher D. Manning/Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts and London, England: MIT Press MS99

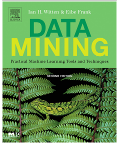
Literature



Dan Jurafsky/James H. Martin (2023). *Speech and Language Processing*. 3rd ed. Draft of Janaury 7, 2023. Prentice Hall. URL: https://web.stanford.edu/~jurafsky/slp3/ JM23



Christopher D. Manning/Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts and London, England: MIT Press MS99



Ian H. Witten/Eibe Frank (2005). *Data Mining*. 2nd ed. Practical Machine Learning Tools and Techniques. Elsevier WF05

Section 2

Corpora

Corpora

- (Large) collections of linguistic expressions
- Speech corpora: Spoken language
  - File formats: wav, mp3, ...
- Text corpora: Written language
  - File formats: txt, xml, json, ...

Corpora

- (Large) collections of linguistic expressions
- Speech corpora: Spoken language
  - File formats: wav, mp3, ...
- Text corpora: Written language
  - File formats: txt, xml, json, ...
- Why do we look at corpora?

Corpora

- (Large) collections of linguistic expressions
- Speech corpora: Spoken language
  - File formats: wav, mp3, ...
- Text corpora: Written language
  - File formats: txt, xml, json, ...
- Why do we look at corpora?
  - Making statements about language needs to take into account many language expressions
  - We under-estimate creativity, flexibility and productivity of language use

  → Empiricism

Meta data and annotations

Meta data: Data about the data

- Information about the corpus
- Language, date of creation, author(s), publication source, ...
- Machine-readable: XML, JSON, CSV, ...

Meta data and annotations

Meta data: Data about the data

- Information about the corpus
- Language, date of creation, author(s), publication source, ...
- Machine-readable: XML, JSON, CSV, ...

Annotations: Data about parts of the corpus

- Examples
  - Linguistic annotation: Parts of speech, named entities, syntactic relations, ...
  - Non-linguistic annotation: Sentiment expressions, rhetoric devices, arguments, ...

Meta data and annotations

Meta data: Data about the data

- Information about the corpus
- Language, date of creation, author(s), publication source, ...
- Machine-readable: XML, JSON, CSV, ...

Annotations: Data about parts of the corpus

- Examples
  - Linguistic annotation: Parts of speech, named entities, syntactic relations, ...
  - Non-linguistic annotation: Sentiment expressions, rhetoric devices, arguments, ...
- Explicit location in the corpus: Document/word/character numbers in text, milliseconds in speech

Preparations (for text corpora)

- OCR: Optical Character Recognition (MS99, 123)
  - Convert images (e.g., from a scan) into text
  - Huge improvements in last five years

Preparations (for text corpora)

- OCR: Optical Character Recognition (MS99, 123)
  - Convert images (e.g., from a scan) into text
  - Huge improvements in last five years
- Encoding: How to specify characters in a computer
  - Simple: ASCII (7 bit per character, $2^7 = 128$ different characters)
  - Outdated: Latin-1 / ISO-8859 (8 bit, $\Rightarrow 256$ diff. characters)
  - Modern: Unicode (e.g., UTF-8)
    - 1 B/char to 4 B/char
    - $1\,112\,064$ characters can be represented

Tools and Techniques

- Plain text editors
  - We often want to inspect the corpus as it is on disk (i.e., without an editor interfering too much)
  - Mac: Textmate/emacs/vi; Windows: Notepad++/emacs/vi

Tools and Techniques

- Plain text editors
  - We often want to inspect the corpus as it is on disk (i.e., without an editor interfering too much)
  - Mac: Textmate/emacs/vi; Windows: Notepad++/emacs/vi
- Regular expressions
  - The most important tool for corpus analysis
    - Cleanup (e.g., after scraping a corpus from the web)
    - Analysis (e.g., to find all variants of a word or deal with slang)
  - Usable in *all** programming languages and find tools

Tools and Techniques

- Plain text editors
  - We often want to inspect the corpus as it is on disk (i.e., without an editor interfering too much)
  - Mac: Textmate/emacs/vi; Windows: Notepad++/emacs/vi
- Regular expressions
  - The most important tool for corpus analysis
    - Cleanup (e.g., after scraping a corpus from the web)
    - Analysis (e.g., to find all variants of a word or deal with slang)
  - Usable in *all** programming languages and find tools
- Command line
  - Large corpora often cannot be displayed with GUI tools
  - Command line tools faster and more memory efficient

Tokenization

- Segmenting a corpus into individual units
- Tokens: Words, punctuation, numbers, symbols, ...

Tokenization

- Segmenting a corpus into individual units
- Tokens: Words, punctuation, numbers, symbols, ...
- Naive: Splitting at white space (space, newline, ...)
  - Why naive?

Tokenization

- Segmenting a corpus into individual units
- Tokens: Words, punctuation, numbers, symbols, ...
- Naive: Splitting at white space (space, newline, ...)
  - Why naive?
- Solved, but complex
  - E.g., syntactic points vs. morphological points
- Sometimes, shortcuts are ok – depends on the use case

Word Counts

Count	Word
585	die
584	und
407	er
404	der
348	zu
311	sich
259	nicht
250	sie
243	in
243	den
233	war
218	Gregor
189	mit
178	das
176	auf
171	es
162	dem
155	hatte
137	ein
136	aber
133	daß
123	als
110	auch
107	Schwester
	...

Word Counts

Count	Word
585	die
584	und
407	er
404	der
348	zu
311	sich
259	nicht
250	sie
243	in
243	den
233	war
218	Gregor
189	mit
178	das
176	auf
171	es
162	dem
155	hatte
137	ein
136	aber
133	daß
123	als
110	auch
107	Schwester
	...

- Number of words in a text
- Most frequent words (MFW) are function words
- ›Content words‹ that appear often indicate text content

Zipf's Law

MS99, 23 ff.

- George Kingsley Zipf (1902-1950): American Linguist
- Basic property of human language
    - Frequency distribution of words (in a corpus) is stable
    - Word frequency is inversely proportional to its position in the ranking

$$f \propto \frac{1}{r}$$

(there is a constant k, such that $f \times r = k$)

Zipf's Law

MS99, 23 ff.

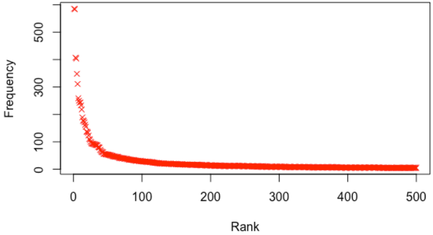


Figure: Words sorted after their frequency (red). Text: Kafka's »Die Verwandlung«.

Zipf's Law

MS99, 23 ff.

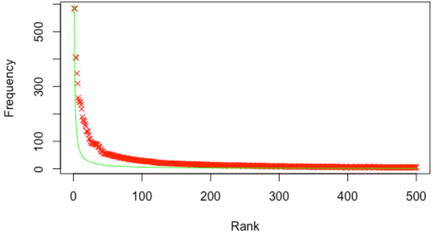


Figure: Words sorted after their frequency (red). Zipf distribution: $y = 600\frac{1}{x}$ (green). Text: Kafka's »Die Verwandlung«.

Zipf's Law

MS99, 23 ff.

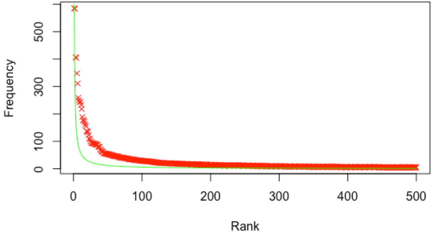


Figure: Words sorted after their frequency (red). Zipf distribution: $y = 600\frac{1}{x}$ (green). Text: Kafka's »Die Verwandlung«.

Consequences

- Very few words appear with very high frequency
- The vast majority of words appear only once
  - It's difficult to learn something about these words!

Counting Words

- Absolute numbers are not that interesting
- Insights are only generated through comparison

Abs. number	Word form
20	women
67	woman
31	men
79	family
82	sister
83	friend
99	bath
117	father
133	man
144	sir

Table: Jane Austens's *Persuasion* (nouns)

Abs. number	Word form
0	friend
2	bath
11	women
23	men
30	father
68	woman
83	family
113	sir
121	man
282	sister

Table: Jane Austens's *Sense and Sensibility* (nouns)

Absolute Numbers

Word	Persuasion	Sense
woman	67	68
women	20	11
man	133	121
men	31	23
sister	82	282

...does it make sense to compare absolute numbers? No.

Absolute Numbers

Word	Persuasion	Sense
woman	67	68
women	20	11
man	133	121
men	31	23
sister	82	282

...does it make sense to compare absolute numbers? No.

- The texts/corpora do not have the same size
- Scaling using their length: Division by the total number of words

Absolute Numbers

Word	Persuasion		Sense	
woman	67	$0.000\,79\,\%$	68	$0.000\,55\,\%$
women	20	$0.000\,24\,\%$	11	$0.000\,09\,\%$
man	133	$0.001\,58\,\%$	121	$0.001\,00\,\%$
men	31	$0.000\,37\,\%$	23	$0.000\,19\,\%$
sister	82	$0.000\,97\,\%$	282	$0.002\,33\,\%$

...does it make sense to compare absolute numbers? No.

- The texts/corpora do not have the same size
- Scaling using their length: Division by the total number of words
- Visible changes: Proportion of »sister«: $3.4 \rightarrow 2.4$

Scaling

- Number of words: Result of a measurement
- If measuring in different scenarios, it's important to scale the results
    - »In a text that is much shorter, there are much less chances for a certain word to be used.«

Scaling

- Number of words: Result of a measurement
- If measuring in different scenarios, it's important to scale the results
  - »In a text that is much shorter, there are much less chances for a certain word to be used.«

Recipe

- Divide the result of the measurement by the theoretical maximum
- How many chances are there for »sister« to be used?
  - As many as there are words in the text
- Thus, we divide by the total number of words

Scaling

- Number of words: Result of a measurement
- If measuring in different scenarios, it's important to scale the results
    - »In a text that is much shorter, there are much less chances for a certain word to be used.«

Recipe

- Divide the result of the measurement by the theoretical maximum
- How many chances are there for »sister« to be used?
    - As many as there are words in the text
- Thus, we divide by the total number of words

- It's not always obvious how to scaled
- When reading research: Was it scaled, and how?

Types and Tokens

MS99, 21 f.

- If a text has been tokenized, we can access individual units: Tokens
- Not all tokens are words: Punctuation, detached prefixes, ...

Types and Tokens

MS99, 21 f.

- If a text has been tokenized, we can access individual units: Tokens
- Not all tokens are words: Punctuation, detached prefixes, ...
- We are often also interested in different tokens: Types

Types and Tokens

MS99, 21 f.

- If a text has been tokenized, we can access individual units: Tokens
- Not all tokens are words: Punctuation, detached prefixes, ...
- We are often also interested in different tokens: Types

Example

the cat chases the mouse

Types and Tokens

MS99, 21 f.

- If a text has been tokenized, we can access individual units: Tokens
- Not all tokens are words: Punctuation, detached prefixes, ...
- We are often also interested in different tokens: Types

Example

the cat chases the mouse

- Tokens: the, cat, chases, the, mouse
- Types: the, cat, chases, mouse

Type-Token-Ratio (TTR)

- What is the relation between number of tokens and number of types?

Type-Token-Ratio (TTR)

- What is the relation between number of tokens and number of types?
- Construct a sentence with 5 tokens and 5 types!

Type-Token-Ratio (TTR)

- What is the relation between number of tokens and number of types?
- Construct a sentence with 5 tokens and 5 types!
  - »the dog barks loudly .«

Type-Token-Ratio (TTR)

- What is the relation between number of tokens and number of types?
- Construct a sentence with 5 tokens and 5 types!
  - »the dog barks loudly .«
- Construct a sentence with 5 tokens and 4 types!

Type-Token-Ratio (TTR)

- What is the relation between number of tokens and number of types?
- Construct a sentence with 5 tokens and 5 types!
  - »the dog barks loudly .«
- Construct a sentence with 5 tokens and 4 types!
  - »the cat loves the mouse«

Type-Token-Ratio (TTR)

- What is the relation between number of tokens and number of types?
- Construct a sentence with 5 tokens and 5 types!
  - »the dog barks loudly .«
- Construct a sentence with 5 tokens and 4 types!
  - »the cat loves the mouse«
- Construct a sentence with 5 tokens and 1 type!

Type-Token-Ratio (TTR)

- What is the relation between number of tokens and number of types?
- Construct a sentence with 5 tokens and 5 types!
  - »the dog barks loudly .«
- Construct a sentence with 5 tokens and 4 types!
  - »the cat loves the mouse«
- Construct a sentence with 5 tokens and 1 type!
  - »dog dog dog dog dog« (not really a sentence …)
  - It's not possible to create a ›proper‹ sentence with 1 type

Type-Token-Ratio (TTR)

- Measure for ›lexical variability‹

$$TTR = \frac{\text{number of types}}{\text{number of tokens}}$$

- Max value: 1

Type-Token-Ratio (TTR)

- Measure for ›lexical variability‹

$$TTR = \frac{\text{number of types}}{\text{number of tokens}}$$

- Max value: 1 (there cannot be more types than tokens)
- Min value: $\epsilon = \frac{1}{\text{very large number}}$

Type-Token-Ratio (TTR)

- Measure for ›lexical variability‹

$$TTR = \frac{\text{number of types}}{\text{number of tokens}}$$

- Max value: 1 (there cannot be more types than tokens)
- Min value: $\epsilon = \frac{1}{\text{very large number}}$
- Real (German) texts
  - 1000 words (Wikipedia): $\frac{4021}{10\,000} = 0.4021$

TTR and Text Length

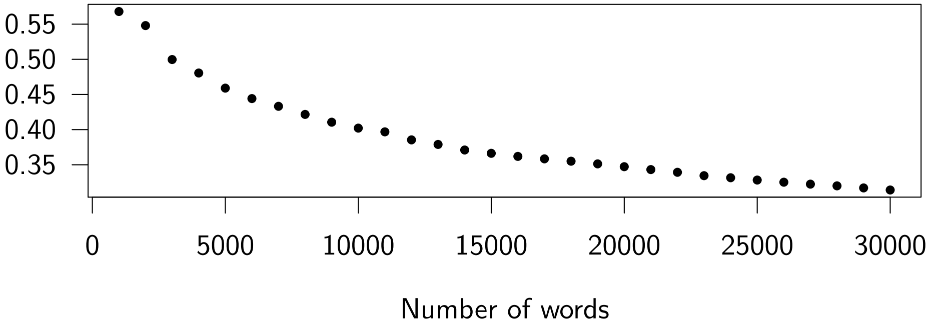


Figure: Type-Token-Ratio for increasing text lengths

TTR and Text Length

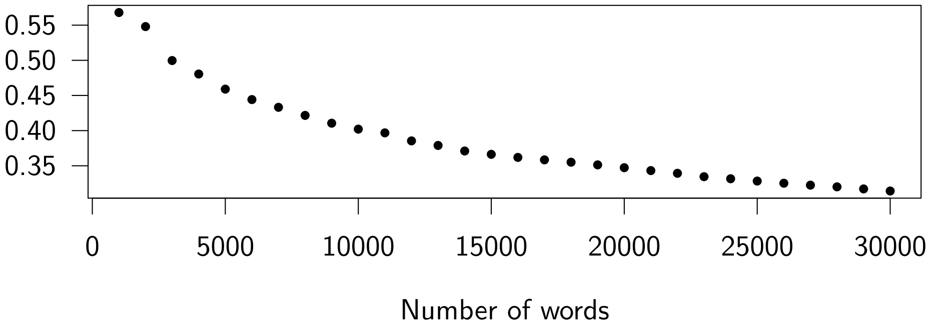


Figure: Type-Token-Ratio for increasing text lengths

- Increasing length → lower TTR!
- Why?

TTR and Text Length

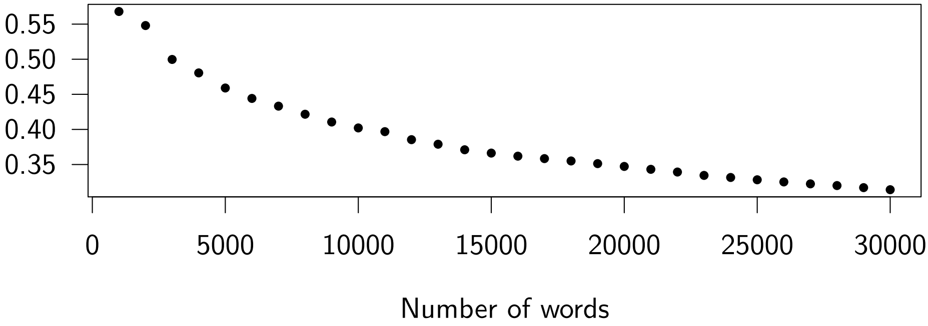


Figure: Type-Token-Ratio for increasing text lengths

- Increasing length → lower TTR!
- Why?– Zipf!

Standardized TTR (STTR)

- Calculate TTR over windows of fixed size (e.g., 1000 words)
- Calculate arithmetic mean over TTR values

Standardized TTR (STTR)

- Calculate TTR over windows of fixed size (e.g., 1000 words)
- Calculate arithmetic mean over TTR values

$$TTR_n = \frac{\text{number of types in } n\text{th window}}{\text{number of tokens in } n\text{th window}}$$

Standardized TTR (STTR)

- Calculate TTR over windows of fixed size (e.g., 1000 words)
- Calculate arithmetic mean over TTR values

$$TTR_n = \frac{\text{number of types in } n\text{th window}}{\text{number of tokens in } n\text{th window}}$$

$$STTR = \frac{1}{w}\sum_{i=0}^{w} TTR_i$$

n-grams

- So far: Individual tokens
- But: Context is important for linguistic expressions

n-grams

- So far: Individual tokens
- But: Context is important for linguistic expressions
- n-gram: A list of n directly adjacent tokens
    - Popular choices for n: 2 to 4

n-grams

- So far: Individual tokens
- But: Context is important for linguistic expressions
- n-gram: A list of n directly adjacent tokens
  - Popular choices for n: 2 to 4

Example

The dog barks.

- 1-grams: »the«, »dog«, »barks«, ».«
- 2-grams (bigrams): »the dog«, »dog barks«, »barks .«
- 3-grams (trigrams): »the dog barks«, »dog barks .«

Section 3

Summary

Summary

- Computational Linguistics as a discipline between computer science and linguistics
  - also known as »natural language processing«, (NLP)
  - Experiments are important way of making progress in CL
- Corpora
- Types and tokens
- Zipf distribution
- Type-Token-Ratio

References I

- ALPAC (1966). *Language and Machines. Computers in Translation and Linguistics*. Tech. rep. National Research Council.
- Bernhart, Toni (2020). »Beiwerk als Werk. Stochastische Texte von Theo Lutz«. In: *editio* 34. DOI: 10.1515/editio-2020-0010.
- Chomsky, Noam (1957). *Syntactic Structures*. Mouton De Gruyter.
- — (1965). *Aspects of the theory of syntax*. MIT Press.
- Devlin, Jacob/Ming-Wei Chang/Kenton Lee/Kristina Toutanova (2019). »BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding«. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

References II

Hutchins, John/Evgenii Lovtskii (2000). »Petr Petrovich Troyanskii (1894-1950): A Forgotten Pioneer of Mechanical Translation«. In: *Machine Translation* 15.3, pp. 187–221. ISSN: 09226567, 15730573. URL: http://www.jstor.org/stable/40009018.

Jurafsky, Dan/James H. Martin (2023). *Speech and Language Processing*. 3rd ed. Draft of Janaury 7, 2023. Prentice Hall. URL: https://web.stanford.edu/~jurafsky/slp3/.

Lutz, Theo (1959). »Stochastische Texte«. In: *augenblick* 4, pp. 3–9. URL: https://www.netzliteratur.net/lutz%5C_schule.htm.

Manning, Christopher D./Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts and London, England: MIT Press.

Mikolov, T./K. Chen/G. Corrado/J. Dean (2013). »Efficient Estimation of Word Representations in Vector Space«. In: *ArXiv e-prints*.

References III

- Nagao, Makoto (1984). »A Framework of a Mechanical Translation between Japanese and English by Analogy Principle«. In: *Proc. of the International NATO Symposium on Artificial and Human Intelligence*. Lyon, France: Elsevier North-Holland, Inc., pp. 173–180. ISBN: 0444865454.
- Reiter, Nils (2014). »Discovering Structural Similarities in Narrative Texts using Event Alignment Algorithms«. PhD thesis. Heidelberg University, Germany.
- Witten, Ian H./Eibe Frank (2005). *Data Mining*. 2nd ed. Practical Machine Learning Tools and Techniques. Elsevier.