Lerneinheit: Named Entity Recognition mit dem Stanford Named Entity Recognizer

Mareike Schumacher [1]

1. Universität Regensburg

forTEXT

Thema:	Named Entity Recognition	DOI:	10.48694/fortext.3766
Jahrgang:	1	Ausgabe:	9
Erscheinungsdatum:	30-10-2024	Erstveröffentlichung:	2019-08-26 auf fortext.net
Lizenz:	CC BY SA		open access

Allgemeiner Hinweis: Rot dargestellte Begriffe werden im Glossar am Ende des Beitrags erläutert. Alle externen Links sind auch am Ende des Beitrags aufgeführt.

Eckdaten der Lerneinheit

- Anwendungsbezug: Figuren in Goethes *Wahlverwandtschaften* (1809)
- Methode: Named Entity Recognition (NER)
- Angewendetes Tool: Stanford Named Entity Recognizer
- Lernziele: Automatische Annotation von Figuren, Berechnung der Güte des Ergebnisses, Verbesserung der Erkennung durch Training eines eigenen NER-Modells
- Dauer der Lerneinheit: 120 Minuten
- Schwierigkeitsgrad des Tools: mittel bis schwierig

Bausteine

- Anwendungsbeispiel: Welcher Primärtext liegt der Analyse zugrunde? Annotieren Sie automatisch die Figuren in Goethes *Wahlverwandtschaften*.
- Vorarbeiten: Welche Arbeitsschritte sollten vor der Analyse ausgeführt werden? Das Tool wird installiert und eine digitale Reintextversion von Goethes *Wahlverwandtschaften* auf Ihren Rechner heruntergeladen.
- Funktionen: Welche Funktionen bietet Ihnen der Stanford Named Entity Recognizer? Lernen Sie, Figuren, Orte und Organisationen in Texten automatisch annotieren zu lassen und ein eigenes Modell zur automatischen Annotation von Eigennamen zu trainieren. Dazu lösen Sie Beispielaufgaben.
- Lösungen zu den Beispielaufgaben: Haben Sie die Beispielaufgaben richtig gelöst? Hier finden Sie Antworten.

1. Anwendungsbeispiel

In dieser Lerneinheit lassen Sie Figuren, Orte und Organisationen in Goethes *Wahlverwandtschaften* automatisch (vgl. Text Mining) annotieren (vgl. Annotation). Sie lernen, die Qualität Ihrer automatischen Annotation zu beurteilen und Sie verbessern das dahinter liegende Modell für Figuren. Die Methode Named Entity Recognition (NER) hat ihren Ursprung in der linguistischen Forschung (siehe Schumacher (2024b) zur Methode der NER). Mittels NER werden dort hauptsächlich Sachtexte unterschiedlicher Art (journalistische Artikel, Social-Media-Postings u. Ä.) untersucht. Eine Domäneadaption der Methode für die Literaturwissenschaften ist häufig mit der Anpassung der implementierten Modelle verbunden. Darum lernen Sie in dieser Lerneinheit nicht nur, den Stanford Named Entity Recognizer (Schumacher 2024c) zu nutzen, sondern auch, die dort angebotenen Funktionen für literaturwissenschaftliche Forschung zu optimieren.

2. Vorarbeiten

Für die automatische Annotation von Eigennamen benötigen Sie den Stanford Named Entity Recognizer, ein Named-Entity-Recognition-Modell für die deutsche Sprache, eine aktuelle Version von Java und Ihren Primärtext im TXT-Format (vgl. Reintext-Version). Den Stanford Named Entity Recognizer können Sie auf dieser Seite herunterladen. Das Modell für die deutsche Sprache bekommen Sie auf dieser Webseite. Sie müssen die Datei lediglich von der Dropbox-Adresse herunterladen, sie wird später mit dem Stanford Named Entity Recognizer geöffnet. In dieser Lerneinheit nutzen wir die Version 3.9.2 der deutschen NER-Modelle, die wir von der Webseite der Stanford Named Entity Recognition Group heruntergeladen und für Sie einzeln in eine ZIP-Datei verpackt haben. Wenn Sie das komplette Paket oder eine neue Version herunterladen wollen, kann es sein, dass die heruntergeladene Datei eine JAVA Datei ist („stanford-german-corenlp-2018-10-05-models.jar“), die als Archiv

genutzt wird. Um Dateien dieser Art zu entpacken, ist auf den meisten Betriebssystemen die Installation eines weiteren Programmes notwendig. Ein MacOS-Programm, das solche Dateien in einfache Ordner umwandeln kann, ist z. B. „Open All Files“, eines für Windows ist z. B. „WinZip“. Die aktuelle Java-Version (falls Sie noch keine auf Ihrem PC haben) finden Sie über diesen Link. Den vom Deutschen Textarchiv (DTA) stammenden Primärtext in einer TXT-Version erhalten Sie als Direktdownload auf Zenodo (forTEXT 2019c). Der Stanford Named Entity Recognizer benötigt keine Installation, sondern kann sofort genutzt werden.

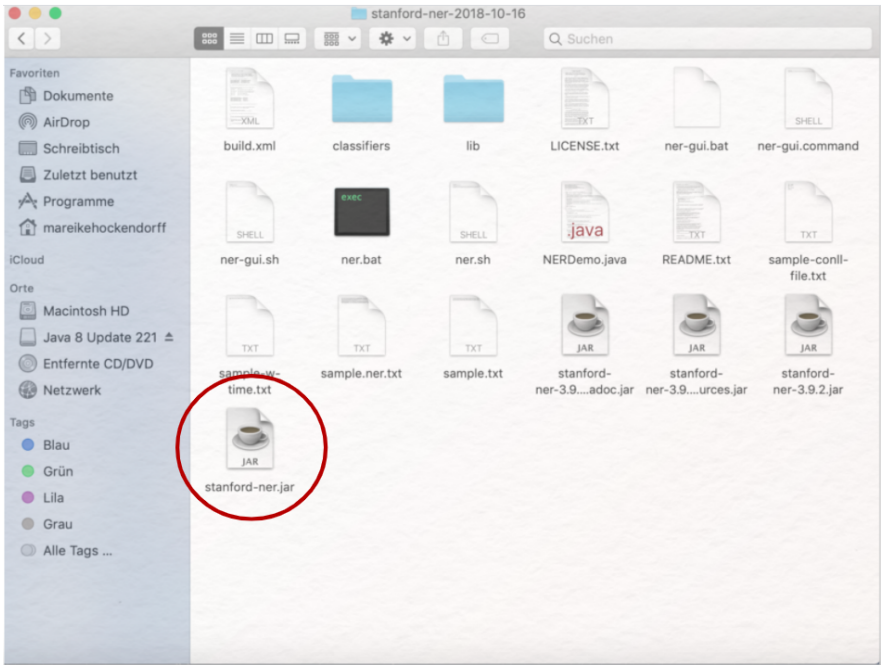


Abb. 1: Stanford Named Entity Recognizer: Entpackter Ordner

Entpacken Sie die ZIP-Datei, die Sie von der Homepage der Stanford Natural Language Group heruntergeladen haben, indem Sie doppelt auf den Archivordner klicken. Öffnen Sie den Ordner, der nun neu angelegt wurde, so sehen Sie darin mehrere Dateien und Unterordner. Je nach Version des Stanford Named Entity Recognizers und Ihres eigenen Betriebssystems, müssen Sie nun eine der ausführbaren Dateien öffnen. Eventuell müssen Sie dafür Ihre Sicherheitseinstellungen anpassen (forTEXT 2019b; forTEXT 2019a).
Für MacOS ist die in Abb. 1 eingekreiste JAR-Datei die richtige. Unter Windows probieren Sie am besten zuerst eine Datei, die GUI heißt (oben nicht im Bild). Öffnen Sie die Datei per Doppelklick, sollte sich eine einfache grafische Benutzeroberfläche zeigen:

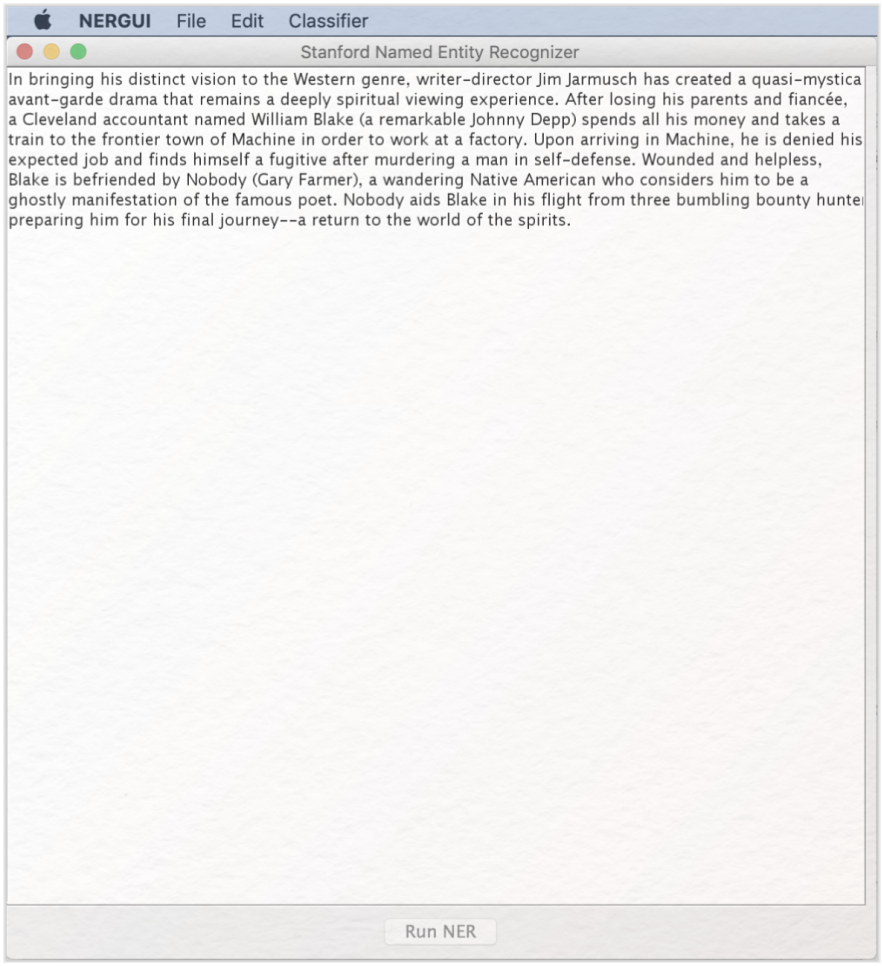


Abb. 2: Grafische Benutzeroberfläche des Stanford Named Entity Recognizers

Die Kategorien, die mit Hilfe des Tools automatisch annotiert werden können (Personen, Orte, Organisationen, Vermischtes), werden als „Classifier“ bezeichnet und können über die obere Menüleiste geladen werden. Klicken Sie auf „Classifier“ und dann auf „Load CRF from file“. CRF steht hier für *Conditional Random Fields*, eine statistische Konkretisierung der sequenziellen Modelle, die in den Classifiern angelegt sind (Sutton und McCallum 2010). Wählen Sie dann die aus der Dropbox heruntergeladene Datei „german.conll.germeval2014.hgc_175m_600.crf.ser.gz“. Nach einem kurzen Moment sollten sich an der rechten Seite der Benutzeroberfläche die Kategorien zeigen.

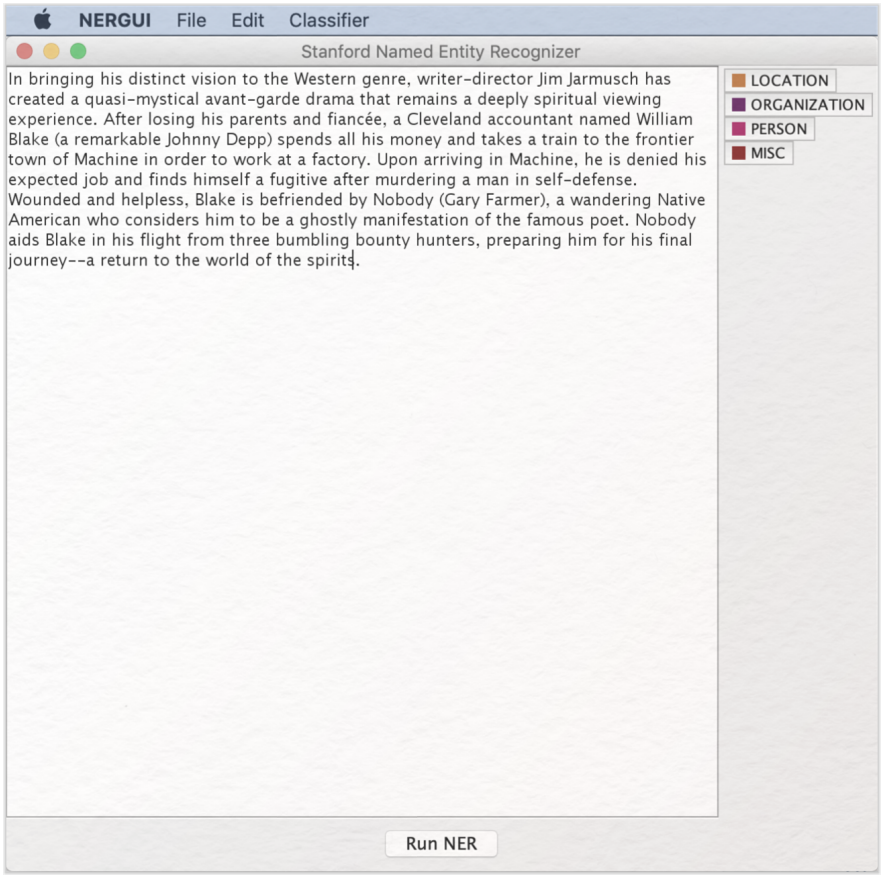


Abb. 3: Stanford Named Entity Recognizer mit geöffneten Classifiern

Jetzt gehen Sie in der oberen Menüleiste auf „File“ und dann im Drop-Down-Menü auf „Open File“. Suchen Sie aus Ihrer Ordnerstruktur den Primärtext heraus und gehen auf „Öffnen“. Der Text zeigt sich in der Benutzeroberfläche des Tools:

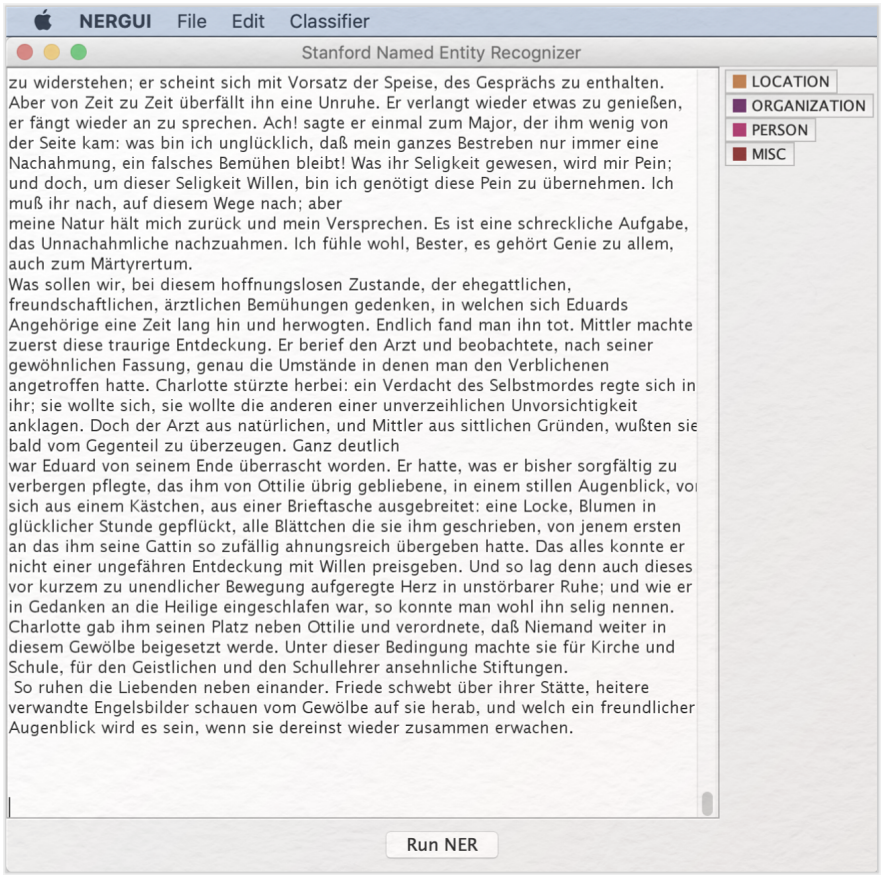


Abb. 4: Stanford Named Entity Recognizer mit geladenen Classifiern und Goethes Wahlverwandtschaften *als Primärtext*

3. Funktionen

Die Kernfunktion des Stanford Named Entity Recognizers – die **automatische Annotation** im Classifier festgelegter Kategorien – ist sehr einfach zu nutzen. Klicken Sie auf „Run NER“ und warten Sie, bis die Schaltfläche nicht mehr blau unterlegt ist.

Aufgabe 1: Scrollen Sie durch Ihren annotierten Text und schauen Sie sich die in unterschiedlichen Farben markierten Eigennamen an. Was fällt Ihnen auf?

Die Qualität der Annotationen können Sie mit Hilfe des digitalen Annotationstools CATMA (Schumacher 2024a) messen. **Speichern** Sie dafür Ihr annotiertes Dokument. Gehen Sie dafür auf „File“ und dann im Drop-Down-Menü auf „Save tagged file as“. Geben Sie der Datei einen Namen, der zeigt, dass es sich hierbei um eine annotierte Version der *Wahlverwandtschaften* handelt. Öffnen Sie dann die gespeicherte Datei mit einem Texteditor, der XML-Dateien (vgl. XML) erstellen kann. Für Windows ist z. B. Notepad ein solches Programm, für MacOS eignet sich BBEdit (ehemals TextWrangler). Öffnen Sie Ihre gerade gespeicherte Datei mit diesem Programm, so sehen Sie, dass der Stanford Named Entity Recognizer als Output eine TXT-Datei generiert hat, die HTML-Tags (vgl. HTML) enthält. Damit ein anderes Programm wie z. B. CATMA erkennen kann, dass es sich um solche handelt (und nicht um Text), muss die Datei in eine vollständige XML-Datei umgewandelt werden. Dazu fügen Sie ganz oben einen sog. Opening-Tag ein. Dieser hat die Struktur `<Ihr Opening Tag\>`. Fügen Sie jetzt den Opening-Tag `<NER\>` ganz oben in Ihr Dokument ein (selbstverständlich können Sie auch eine andere Bezeichnung als „NER“ wählen). Fügen Sie ganz unten im Dokument dann einen sog. Closing-Tag ein, der dazu passt. Für unser Beispiel ist `</NER>` der richtige Closing-Tag.

Abb. 5: Ende des Textes mit Opening-Tag

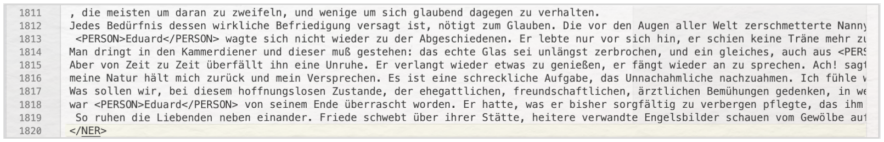


Abb. 6: Ende des Textes mit Closing-Tag

Gehen Sie dann auf „Speichern unter“, geben eine XML-Endung ein und setzen die Codierung auf UTF-8 (vgl. Unicode/UTF-8) (wenn diese Einstellung nicht automatisch vorgenommen wurde):

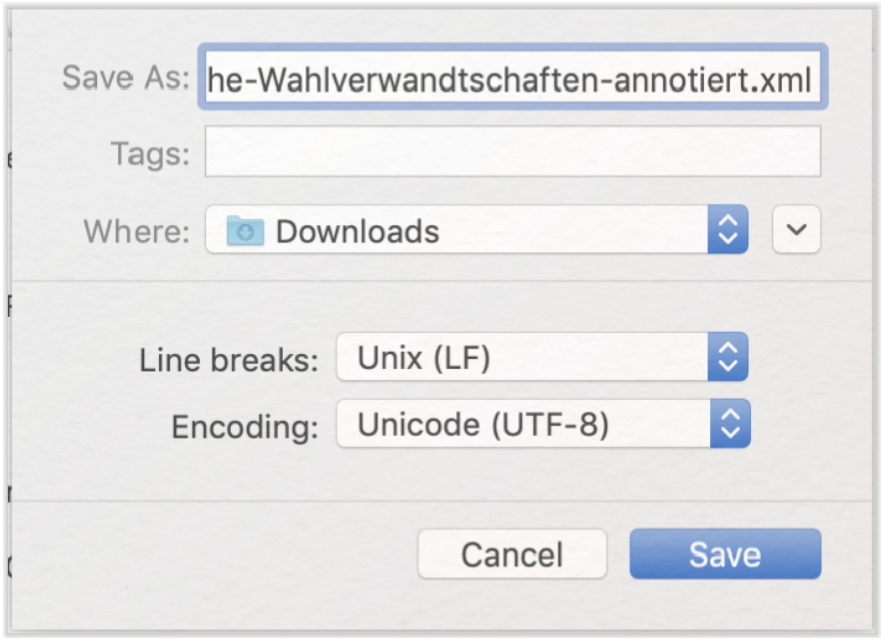


Abb. 7: Menüfeld zum Speichern des Textes

Sobald Sie die Datei als XML gespeichert haben, erscheinen die HTML-Tags nicht mehr als Text in schwarz, sondern werden in blau hervorgehoben:

Abb 8: Text mit blau hervorgehobenen HTML-Tags

Gehen Sie nun zu CATMA und loggen Sie sich ein oder erstellen einen Account. Laden Sie über die Schaltfläche „Add Document“ (es öffnet sich ein Upload-Assistent) Ihre XML-Datei hoch. Klicken Sie dann auf den kleinen Pfeil vor Ihrem Upload, sodass sich die Schaltfläche „Annotations“ öffnet, dann auf den kleinen Pfeil vor „Annotations“, sodass „Intrinsic Markup“ sichtbar wird. Wählen Sie „Intrinsic Markup“ aus und klicken Sie auf „Open Annotations“.

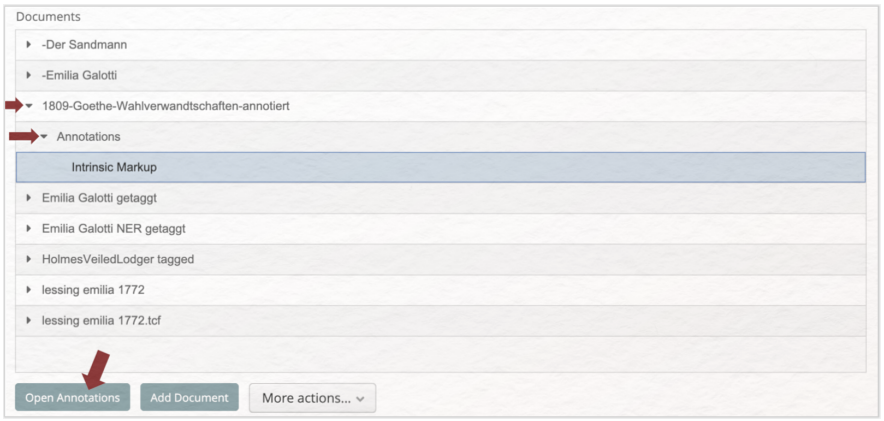


Abb. 9: Ihr XML-Dokument im Modul Manage Resources *von CATMA*

Wenn das Dokument geöffnet ist, klicken Sie erst auf den kleinen Pfeil vor „Intrinsic Markup“ und dann auf den Pfeil vor „NER“. Setzen Sie nun einen Haken hinter „PERSON“, sodass alle gefundenen Personennamen unterstrichen dargestellt werden.

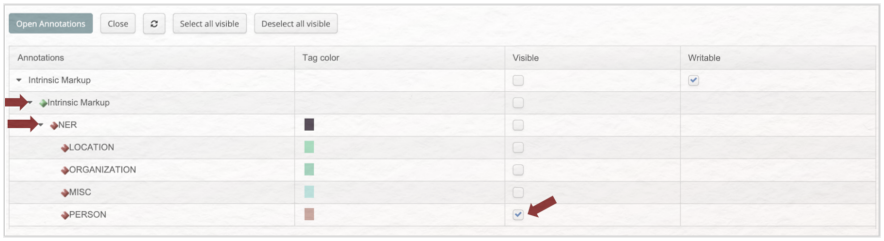


Abb. 10: Schaltflächen zur Sichtbarmachung der Annotationen im Modul Annotate *von CATMA*

Wir berechnen nun die gängigen **Bewertungskennzahlen** für Named Entity Recognition, „Precision“, „Recall“ und „F-score“ beispielhaft anhand der Kategorie PERSON. Dabei zeigt der „Precision“-Wert wie viele der annotierten Textstellen korrekt annotiert wurden. Der „Recall“-Wert gibt Aufschluss darüber wie viele der relevanten Textpassagen annotiert wurden. Mit der Berechnung des „F-Score“ werden die beiden Werte mathematisch kombiniert, sodass sich ein Richtwert für die Bewertung des Modells ergibt.

Aufgabe 2: Zählen Sie nun auf den Seiten 2–6 (es handelt sich dabei um von CATMA vergebene Seiten) folgende Werte aus: Wie viele Textstellen hat der Stanford Named Entity Recognizer mit dem Tag „PERSON“ belegt? Wie viele dieser Textstellen wurden fälschlich mit dem Tag „PERSON“ annotiert? Wie viele Figuren wurden nicht gefunden (Achtung: Das Konzept der literarischen Figur geht über das der Eigennamen hinaus)? Berechnen Sie nun den Wert für „Precision“ indem Sie die Zahl der korrekt gefundenen Personennamen durch die Gesamtzahl der mit dem Tag „PERSON“ annotierten Textstellen dividieren. Berechnen Sie dann den Wert für „Recall“, indem Sie die Zahl der korrekt gefundenen Personennamen durch die Gesamtzahl der Personennamen dividieren. Berechnen Sie anschließend den F-Score nach folgender Formel: 2 x ((Precision x Recall) / (Precision + Recall)).

Dass die Werte insgesamt noch nicht ausreichend sind, um die automatische Annotation für die Literaturanalyse zu nutzen, liegt daran, dass das Tool für Sachtexte optimiert ist. Außerdem entspricht das linguistische Konzept der Entitäten nicht dem der literarischen Figur. Da der Stanford Named Entity Recognizer einige Funktionalitäten mitbringt, die das sogenannte **Training eines eigenen Modells** (vgl. Machine Learning) unterstützen, kann man hier allerdings nachhelfen. Das bedeutet, dass das Tool für das eigene Textkorpus (vgl. Korpus) optimiert werden kann.
Grundsätzlich gibt es dafür zwei Ansätze. Entweder wird ein korpusspezifisches, sehr enges Modell trainiert, dass für die eigene Analyse gute Ergebnisse erzielt. Die andere Möglichkeit ist, ein generisches Modell zu entwickeln, das auch auf andere Texte mit ähnlichen Eigenschaften übertragen werden kann. In dieser Lerneinheit trainieren wir ein Modell, dass für Goethes *Wahlverwandtschaften* optimiert ist und nicht gut auf andere Texte übertragen werden kann. Dazu legen wir zuerst ein Trainingskorpus an, das aus dem ersten Kapitel besteht, das wir per Copy-/Paste in eine TXT-Datei überführen, die wir dann „GoetheWahlverwandtschaftenErstesKapitel.txt“ nennen und in dem Ordner ablegen, in dem auch der Stanford Named Entity Recognizer liegt (Ordner-Name „stanford-ner-2018-02-27“). Wollten Sie ein übertragbares Modell trainieren, müssten Sie für das Trainingskorpus ungefähr gleich große Ausschnitte aus verschiedenen Texten zusammen kopieren, die ähnliche Eigenschaften wie die zu untersuchenden haben. Ein generisches Modell, dass auch für Goethes *Wahlverwandtschaften* gute Ergebnisse erzielen sollte wäre z. B. mit Hilfe eines Trainigskorpus erstellt, das folgende Eigenschaften besitzt: Erstellt aus zufällig ausgewählten Passagen aus 50 deutschsprachigen Romanen des 19. Jahrhunderts, Gesamtumfang mindestens 40.000 Tokens (vgl. Type/Token).
Erstellen Sie mit Hilfe des im Stanford Named Entity Recognizer enthaltenen Tokenizers aus Ihrem Trainingskorpus eine Wortliste. Öffnen Sie dazu die Commandline ihres Computers (auf dem Mac finden Sie diese unter Dienstprogramme → Terminal, bei Windows gehen Sie auf das Windows-Symbol unten links, geben dann cmd in die Suchleiste ein und wählen das entsprechende Programm aus). Gehen Sie nun in Ihrer Ordnerstruktur auf den Ordner, in dem der Stanford Named Entity Recognizer liegt. Nutzen Sie dafür den Commandline-Befehl „Change Directory“, der cd abgekürzt wird. Wenn der Stanford Named Entity Recognizer in Ihrem Downloads-Ordner ist, geben Sie also ein:

```
cd downloads/stanford-ner-2018-10-16
```

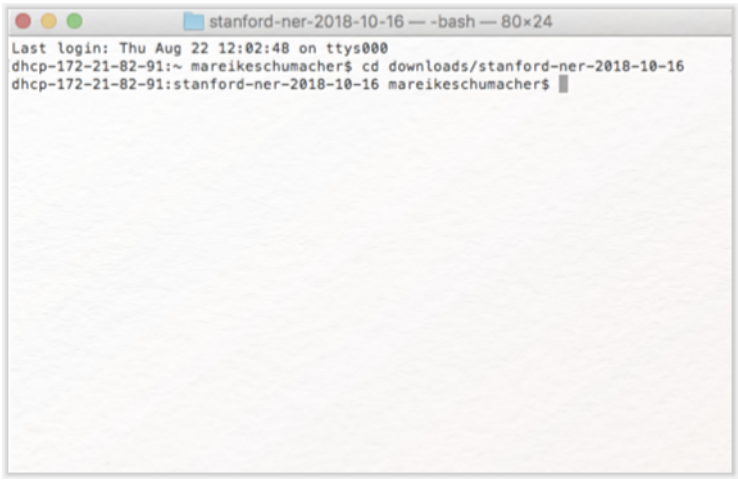


Abb.11: Commandline-Programm nach richtiger Eingabe des cd-Befehls

Ist dies nicht der Fall, ändern Sie den Befehl entsprechend Ihrer eigenen Ordner-Struktur. Geben Sie danach folgenden Befehl ein:

```
java -cp stanford-ner.jar edu.stanford.nlp.process.PTBTokenizer
    ↪ GoetheWahlverwandtschaftenErstesKapitel.txt >
    ↪ GoetheWahlverwandtschaftenErstesKapitel.tok
```

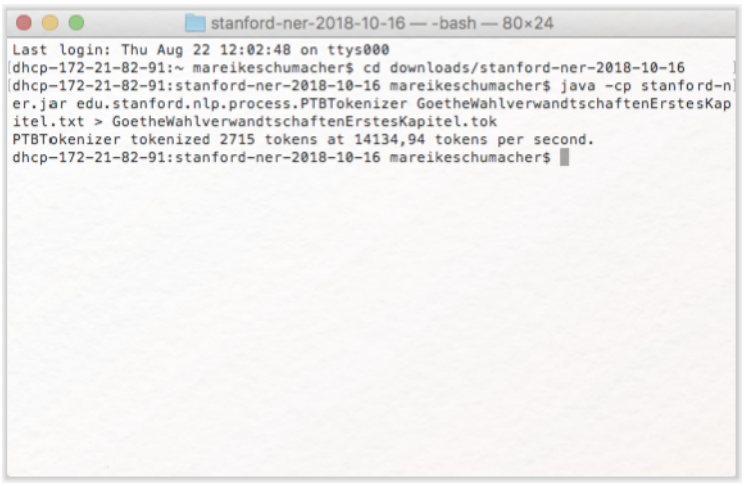


Abb.12: Commandline-Programm nach richtiger Eingabe des Tokenizer-Befehls

Sobald Sie den Befehl mit Enter bestätigt haben, wird Ihr Computer ihn ausführen und eine Datei namens GoetheWahlverwandtschaftenErstesKapitel.tok im Stanford-Ordner entwerfen. Um Ihr Trainingskorpus so annotieren zu können, dass das Tool es später lesen kann, müssen Sie diese Datei nun noch in eine Tabelle umwandeln. Auch dafür hat der Stanford Named Entity Recognizer eine Funktion, die sie aufrufen, indem Sie Folgendes in Ihre Commandline tippen:

```
perl -ne 'chomp;␣print␣"$_\tO\n"' GoetheWahlverwandtschaftenErstesKapitel.tok >
    ↪  GoetheWahlverwandtschaftenErstesKapitel.tsv
```

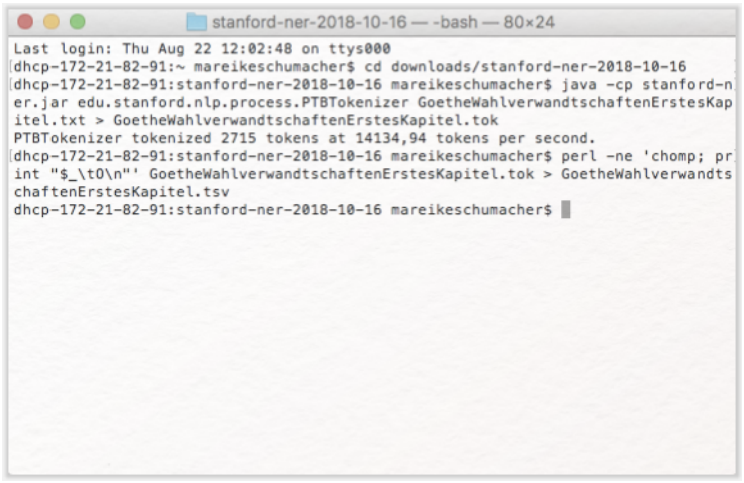
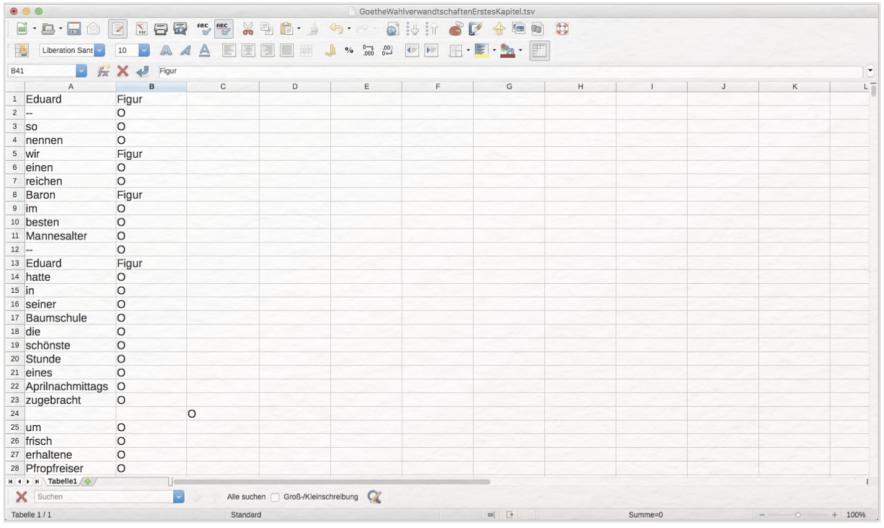


Abb.13: Commandline-Programm nach richtiger Eingabe des Tabellen-Befehls

Die Tabelle, die Sie in Ihrem Stanford-Ordner finden, nachdem Ihr Computer den Befehl ausgeführt hat, öffnen Sie nun in einem Tabellenprogramm wie LibreOffice.

Aufgabe 3: Annotieren Sie Ihr Trainingskorpus, indem Sie bei jeder Figurenreferenz in der zweiten Tabellenspalte das O gegen das Wort Figur austauschen. Was fällt Ihnen beim Annotieren auf?



	A	B
1	Eduard	Figur
2	--	O
3	so	O
4	nennen	O
5	wir	Figur
6	einen	O
7	reichen	O
8	Baron	Figur
9	im	O
10	besten	O
11	Mannesalter	O
12	--	O
13	Eduard	Figur
14	hatte	O
15	in	O
16	seiner	O
17	Baumschule	O
18	die	O
19	schönste	O
20	Stunde	O
21	eines	O
22	Aprilnachmittags	O
23	zugebracht	O
24		O
25	um	O
26	frisch	O
27	erhaltene	O
28	Pfropfreiser	O

Abb 14: Das Trainingskorpus als Tabelle

Sie haben nun Ihr eigenes Trainingskorpus erstellt. Dies ist ein zentraler Bestandteil für das Training des Stanford Named Entity Recognizers, da hier Beispiele der Kategorie, die Sie automatisch annotieren wollen im Satzzusammenhang enthalten sind. Neben diesen Beispielen benötigt das Tool noch eine Reihe von sogenannten Features (vgl. Feature), anhand derer die Beispiele analysiert werden können. Ein solches Feature wäre z. B. die Orthografie (Personennamen bestehen oft aus zwei Wörtern, die beide groß geschrieben sind), andere Feature-Beispiele sind vorangehende Wörter und nachgestellte Wörter. Der Stanford Named Entity Recognizer ist mit 15 solcher Features ausgestattet, die allerdings auch angepasst werden können. Sowohl die Trainingsdaten als auch die Features werden in einer Properties-Datei (vgl. Property) spezifiziert.
Erstellen Sie eine Properties-Datei, indem Sie in einem Programm wie BBEdit (Mac) oder Notepad (Windows) ein neues Dokument öffnen, in das Sie Folgendes hinein kopieren:

```
trainFile = GoetheWahlverwandtschaftenErstesKapitel.tsv
serializeTo = ner-modelWahlverwandtschaften.ser.gz
map = word=0,answer=1
useClassFeature=true
useWord=true
useNGrams=true
noMidNGrams=true
```

```
maxNGramLeng=6
usePrev=true
useNext=true
useSequences=true
usePrevSequences=true
maxLeft=1
useTypeSeqs=true
useTypeSeqs2=true
useTypeySequences=true
wordShape=chris2useLC
useDisjunctive=true
```

Speichern Sie die Datei unter „GoetheWahlverwandtschaften.prop“. Sie können die Features anpassen, indem Sie die Liste der use-Befehle ergänzen. Weitere Features, die Sie ergänzen können, finden Sie auf dieser Webseite.

Sie haben nun ein Trainingskorpus erstellt und manuell nach Ihren Vorstellungen annotiert. In Ihrer Properties-Datei haben Sie spezifiziert, welche Trainingsdatei verwendet werden soll und welche Features beim Training berücksichtigt werden. Sie sind nun startklar, um Ihr eigenes Modell erstellen zu lassen.

Aufgabe 4: Kopieren Sie folgenden Befehl in Ihre Commandline:

```
java -cp stanford-ner.jar edu.stanford.nlp.ie.crf.CRFClassifier -prop
    ↪ GoetheWahlverwandtschaften.prop
```

Es kann eine Weile dauern, bis Ihr Computer das Training abgeschlossen hat. Anschließend finden Sie eine Datei namens ner-modelWahlverwandtschaften.ser.gz in Ihrem Stanford-Ordner. Sie können nun zurück zur grafischen Benutzeroberfläche des Stanford Named Entity Recognizers wechseln. Laden Sie über Classifier → Load Classifier → ner-modelWahlverwandtschaften.ser.gz Ihr eigenes NER-Modell in das Tool. Falls Sie die Benutzeroberfläche wieder neu öffnen mussten, rufen Sie über File → Open file Goethes *Wahlverwandtschaften* auf. Klicken Sie anschließend auf „Run-NER“. Schauen Sie sich die Ergebnisse Ihres Trainings an. Entsprechen Sie Ihren Erwartungen? Wie können Sie nun die Qualität Ihres Modells erfassen? Wie können Sie das Modell noch präziser werden lassen?

4. Lösungen zu den Beispielaufgaben

Aufgabe 1: Scrollen Sie durch Ihren annotierten Text und schauen Sie sich die in unterschiedlichen Farben markierten Eigennamen an. Was fällt Ihnen auf?

Der Stanford Named Entity Recognizer annotiert Figurennamen meist korrekt. Allerdings findet das Tool nicht alle Referenzen auf Personen. Personalpronomen und Beschreibungen bleiben z. B. unberücksichtigt. Für literarische Texte sind aber häufig gerade Beschreibungen, Spitznamen oder indirekte Referenzen auf Figuren interessant.

Aufgabe 2: Zählen Sie nun auf den Seiten 2–6 (es handelt sich dabei um von CATMA vergebene Seiten) folgende Werte aus: Wie viele Textstellen hat der Stanford Named Entity Recognizer mit dem Tag „PERSON“ belegt? Wie viele dieser Textstellen wurden fälschlich mit dem Tag „PERSON“ annotiert? Wie viele Figuren wurden nicht gefunden (Achtung: Das Konzept der literarischen Figur geht über das der Eigennamen hinaus)? Berechnen Sie nun den Wert für „Precision“ indem Sie die Zahl der korrekt gefundenen Personennamen durch die Gesamtzahl der mit dem Tag „PERSON“ annotierten Textstellen dividieren. Berechnen Sie dann den Wert für „Recall“, indem Sie die Zahl der korrekt gefundenen Personennamen durch die Gesamtzahl der Personennamen dividieren. Berechnen Sie anschließend den F-Score nach folgender Formel: 2 x ((Precision x Recall) / (Precision + Recall)).

- Anzahl von Textstellen, die mit dem Tag PERSON belegt wurden: 27,
- Anzahl von Textstellen ,die fälschlich mit dem Tag „PERSON“ belegt wurden: 1,
- Anzahl von Figurenreferenzen, die nicht gefunden wurden: 182.

Ihre Zahl kann hier abweichen, wenn Sie ein anderes Konzept von Figur verfolgt haben als wir. Mit diesen Werten ist:

- Precision = 0,96296 oder 96,3%,
- Recall = 0,1244 oder 12,44% und
- F-Score = 0,22034 oder 22,03%.

Das bedeutet, dass das Tool zwar relativ präzise Figuren erkennt, die meisten als Personen annotierten Passagen

also tatsächlich auf Figuren verweisen. Der sehr niedrige Recall-Wert zeigt, dass nicht sehr viele Figurenreferenzen gefunden wurden. Das liegt hier vor allem daran, dass das Konzept der literarischen Figur mehr beinhaltet als nur die Bezeichnung einer Figur mit einem Eigennamen. Der F-Score kombiniert die beiden oberen Werte mathematisch. Das Ergebnis zeigt, dass eine Domänenadaption für die Literatur sehr sinnvoll ist, da vor allem der Recall-Wert verbessert werden muss. Das kann durch das Training eines eigenen NER-Modells erreicht werden.

Aufgabe 3: Annotieren Sie Ihr Trainingskorpus, indem Sie bei jeder Figurenreferenz in der zweiten Tabellenspalte das O gegen das Wort Figur austauschen. Was fällt Ihnen beim Annotieren auf?

Beim Annotieren müsste Ihnen aufgefallen sein, dass Sie ein möglichst klares Modell dessen verwenden sollten, was Sie als Kategorie annotieren lassen möchten. Haben Sie einen weiten Begriff von Figurenreferenzen, annotieren Sie vielleicht auch Possessivpronomen und Figurengruppen („wir“, „Männer“, „Frauen“). Bei einem engen Verständnis der Figur annotieren Sie vielleicht nur Namen, Spitznamen und Beschreibungen von Charakteren.

Aufgabe 4: Kopieren Sie folgenden Befehl in Ihre Commandline:

```
java -cp stanford-ner.jar edu.stanford.nlp.ie.crf.CRFClassifier -prop
    ↪ GoetheWahlverwandtschaften.prop
```

Es kann eine Weile dauern, bis Ihr Computer das Training abgeschlossen hat. Anschließend finden Sie eine Datei namens ner-modelWahlverwandtschaften.ser.gz in Ihrem Stanford-Ordner. Sie können nun zurück zur grafischen Benutzeroberfläche des Stanford Named Entity Recognizers wechseln. Laden Sie über Classifier → Load Classifier → ner-modelWahlverwandtschaften.ser.gz Ihr eigenes NER-Modell in das Tool. Falls Sie die Benutzeroberfläche wieder neu öffnen mussten, rufen Sie über File → Open file Goethes *Wahlverwandtschaften* auf. Klicken Sie anschließend auf „Run-NER“. Schauen Sie sich die Ergebnisse Ihres Trainings an. Entsprechen Sie Ihren Erwartungen? Wie können Sie nun die Qualität Ihres Modells erfassen? Wie können Sie das Modell noch präziser werden lassen?

Vielleicht entsprechen die Ergebnisse, die Ihr eigenes Modell erzielt, noch nicht Ihren Erwartungen. Mit der Berechnung von Precision, Recall und F-Score können Sie genau messen, wie „gut“ Ihr Modell ist. Sollte sich zeigen, dass die Werte noch nicht zufriedenstellend sind (F-Score von 50% und weniger), können Sie Folgendes tun:

1. Erweitern Sie Ihr Trainingskorpus um 10.000 Tokens (selbstverständlich müssen Sie diese dann auch noch einmal in eine Wortliste und dann in eine Tabelle umwandeln, den Inhalt dieser Tabelle in Ihre erste bereits annotierte Tabelle hinein kopieren und die 10.000 zusätzlichen Tokens manuell annotieren).
2. Diversifizieren Sie Ihr Trainingskorpus, indem Sie Ausschnitte aus unterschiedlichen Texten derselben Zeitspanne zusammen kopieren.
3. Variieren Sie die Features, indem Sie der Feature-Liste in der Properties Datei weitere use-Befehle hinzufügen. Die genauen Befehle finden Sie in der Feature Factory von Stanford NLP.

Externe und weiterführende Links

- BBedit (MacOS): https://web.archive.org/save/https://www.barebones.com/products/bbedit/download.html (Letzter Zugriff: 06.10.2024)
- CATMA: https://web.archive.org/save/https://catma.de/ (Letzer Zugriff: 06.10.2024)
- Deutsches Textarchive (DTA): https://web.archive.org/save/https://www.deutschestextarchiv.de (Letzer Zugriff: 06.10.2024)
- Goethes *Wahlverwandtschaften* (Primärtext in TXT-Version): https://web.archive.org/save/https://zenodo.org/records/10592571 (Letzter Zugriff: 06.10.2024)
- Java: https://web.archive.org/save/https://java.com/de/download/ (Letzter Zugriff: 06.10.2024)
- Libre Office: https://web.archive.org/save/https://de.libreoffice.org (Letzer Zugriff: 06.10.2024)
- Notepad (Windows): https://web.archive.org/save/https://notepad-plus-plus.org (Letzter Zugriff: 06.10.2024)
- Stanford Named Entity Recognizer Download: https://nlp.stanford.edu/software/CRF-NER.html#Download (Letzter Zugriff: 06.10.2024)
- Stanford Named Entity Recognizer (deutsche Version): https://web.archive.org/save/https://www.dropbox.com/s/mfnj349ezc1y8x1/german.conll.germeval2014.hgc_175m_600.crf.ser.gz?dl=0 (Letzter Zugriff: 06.10.2024)
- Stanford CoreNLP NER-Modelle: https://web.archive.org/save/https://stanfordnlp.github.io/CoreNLP/index.html#download (Letzer Zugriff: 06.10.2024)

Bibliographie

forTEXT. 2019a. Tutorial: Sicherheitsaufnahme für Internetprogramme Hinzufügen (Mac). 19. Januar. https://doi.org/10.5281/zenodo.11074232.

———. 2019b. Tutorial: Sicherheitsausnahme für Internetprogramme Hinzufügen (Windows). 25. Januar. https://doi.org/10.5281/zenodo.11074222.

forTEXT. 2019c. Named Entity Recognition mit dem Stanford Named Entity Recognizer. Zenodo, 26. August. doi: 10.5281/zenodo.10592571, https://doi.org/10.5281/zenodo.10592571.

Schumacher, Mareike. 2024a. Toolbeitrag: CATMA. Hg. von Evelyn Gius. *forTEXT* 1, Nr. 4. Manuelle Annotation (7. August). doi: 10.48694/fortext.3761, https://fortext.net/tools/tools/catma.

———. 2024b. Methodenbeitrag: Named Entity Recognition (NER). Hg. von Evelyn Gius. *forTEXT* 1, Nr. 9. Named Entity Recognition (30. Oktober). doi: 10.48694/fortext.3765, https://fortext.net/routinen/methoden/named-entity-recognition-ner.

———. 2024c. Toolbeitrag: Stanford Named Entity Recognizer. Hg. von Evelyn Gius. *forTEXT* 1, Nr. 9. Named Entity Recognition (30. Oktober). doi: 10.48694/fortext.3767, https://fortext.net/tools/tools/stanford-named-entity-recognizer.

Sutton, Charles und Andrew McCallum. 2010. An Introduction to Conditional Random Fields. https://people.cs.umass.edu/~mccallum/papers/crf-tutorial.pdf (zugegriffen: 22. August 2019).

Glossar

Annotation Annotation beschreibt die manuelle oder automatische Hinzufügung von Zusatzinformationen zu einem Text. Die manuelle Annotation wird händisch durchgeführt, während die (teil-)automatisierte Annotation durch Machine-Learning-Verfahren durchgeführt wird. Ein klassisches Beispiel ist das automatisierte PoS-Tagging (Part-of-Speech-Tagging), welches oftmals als Grundlage (Preprocessing) für weitere Analysen wie Named Entity Recognition (NER) nötig ist. Annotationen können zudem deskriptiv oder analytisch sein.

Browser Mit Browser ist in der Regel ein Webbrowser gemeint, also ein Computerprogramm, mit dem das Anschauen, Navigieren auf, und Interagieren mit Webseiten möglich wird. Am häufigsten genutzt werden dafür Chrome, Firefox, Safari oder der Internet Explorer.

Classifier Ein Classifier ist ein Algorithmus, der Daten automatisch in eine oder mehrere „Klassen“ bzw. einordnet. Dazu wird der Classifier zunächst mit anmontierten Trainingsdaten trainiert, bevor er auf neue Daten getestet und angewendet werden kann. Eines der gebräuchlichsten Beispiele ist ein E-Mail-Klassifikator, der Spam und Nicht-Spam unterscheidet.

Commandline Die Commandline (engl. *command line interface* (CLI)), auch Kommandozeile, Konsole, Terminal oder Eingabeaufforderung genannt, ist die direkteste Methode zur Interaktion eines Menschen mit einem Computer. Programme ohne eine grafische Benutzeroberfläche (GUI) werden i. d. R. durch Texteingabe in die Commandline gesteuert. Um die Commandline zu öffnen, klicken Sie auf Ihrem Mac „cmd“ + „space“, geben „Terminal“ ein und doppelklicken auf das Suchergebnis. Bei Windows klicken Sie die Windowstaste + „R“, geben „cmd.exe“ ein und klicken Enter.

CSV CSV ist die englische Abkürzung für *Comma Separated Values*. Es handelt sich um ein Dateiformat zur einheitlichen Darstellung und Speicherung von einfach strukturierten Daten mit dem Kürzel „.csv“ , sodass diese problemlos zwischen IT-Systemen ausgetauscht werden können. Dabei sind alle Daten zeilenweise angeordnet. Alle Zeilen wiederum sind in einzelne Datenfelder aufgeteilt, welche durch Trennzeichen wie Semikola oder Kommata getrennt werden können. In Programmen wie Excel können solche Textdateien als Tabelle angezeigt werden.

Data Mining Data Mining gehört zum Fachbereich Information Retrieval und bezieht sich auf die systematische Anwendung computergestützter Methoden, die darauf abzielt, in vorhandenen Datenbeständen Muster, Trends oder Zusammenhänge zu erkennen. Textbasierte Formen des Data Minings sind u. a. Text Mining, Web Mining und Opinion Mining.

Domäneadaption Domäneadaption beschreibt die Anpassung einer in einem Fachgebiet entwickelten digitalen Methode an ein anderes Fachgebiet.

F-score Der F-Score steht für ein statistisches Maß, welches das Verhältnis von Genauigkeit (*Precision*) und Trefferquote (*Recall*) als gewichtetes harmonisches Mittel angibt, und deshalb als gerichtetes, harmonisches Mittel gilt.

Feature Unter Features können Einzelfunktionen eines Tools verstanden werden, die beispielsweise komplexe Funktionen wie die Visualisierung eines Textes als Wordcloud ermöglichen, oder auch kleinere Funktionseinheiten wie den Abgleich einzelner Spracheigenschaften (Properties) mit annotierten Beispieltexten darstellen.

GUI GUI steht für *Graphical User Interface* und bezeichnet eine grafische Benutzeroberfläche. Ein GUI ermöglicht es, Tools mithilfe von grafischen Schaltflächen zu bedienen, um somit beispielsweise den Umgang mit der Commandline zu umgehen.

HTML HTML steht für *Hypertext Markup Language* und ist eine textbasierte Auszeichnungssprache zur Strukturierung elektronischer Dokumente. HTML-Dokumente werden von Webbrowsern dargestellt und geben die Struktur und Online-Darstellung eines Textes vor. HTML-Dateien können außerdem zusätzliche Metainformationen enthalten, die auf einer Webseite selbst nicht ersichtlich sind.

Information Retrieval Die Teildisziplin der Informatik, das Information Retrieval, beschäftigt sich mit der computergestützten Suche und Erschließung komplexer Informationen in meist unstrukturierten Datensammlungen.

Korpus Ein Textkorpus ist eine Sammlung von Texten. Korpora (Plural für „das Korpus“) sind typischerweise nach Textsorte, Epoche, Sprache oder Autor*in zusammengestellt.

Lemmatisieren Die Lemmatisierung von Textdaten gehört zu den wichtigen Preprocessing-Schritten in der Textverarbeitung. Dabei werden alle Wörter (Token) eines Textes auf ihre Grundform zurückgeführt. So werden beispielsweise Flexionsformen wie „schneller“ und „schnelle“ dem Lemma „schnell“ zugeordnet.

Machine Learning Machine Learning, bzw. maschinelles Lernen im Deutschen, ist ein Teilbereich der künstlichen Intelligenz. Auf Grundlage möglichst vieler (Text-)Daten erkennt und erlernt ein Computer die häufig sehr komplexen Muster und Gesetzmäßigkeiten bestimmter Phänomene. Daraufhin können die aus den Daten gewonnen Erkenntnisse verallgemeinert werden und für neue Problemlösungen oder für die Analyse von bisher unbekannten Daten verwendet werden.

Markup (Textauszeichung) Die Textauszeichnung (eng. Markup) fällt in den Bereich der Daten- bzw. Textverarbeitung, genauer in das Gebiet der Textformatierung, welche durch Auszeichnungssprachen wie XML implementiert wird. Dabei geht es um die Beschreibung, wie einzelne Elemente eines Textes beispielsweise auf Webseiten grafisch dargestellt werden sollen.

Markup Language Markup Language bezeichnet eine maschinenlesbare Auszeichnungssprache, wie z. B. HTML, zur Formatierung und Gliederung von Texten und anderen Daten. So werden beispielsweise auch Annotationen durch ihre Digitalisierung oder ihre digitale Erstellung zu Markup, indem sie den Inhalt eines Dokumentes strukturieren.

Metadaten Metadaten oder Metainformationen sind strukturierte Daten, die andere Daten beschreiben. Dabei kann zwischen administrativen (z. B. Zugriffsrechte, Lizenzierung), deskriptiven (z. B. Textsorte), strukturellen (z. B. Absätze oder Kapitel eines Textes) und technischen (z. B. digitale Auflösung, Material) Metadaten unterschieden werden. Auch Annotationen bzw. Markup sind Metadaten, da sie Daten/Informationen sind, die den eigentlichen Textdaten hinzugefügt werden und Informationen über die Merkmale der beschriebenen Daten liefern.

Named Entities Eine Named Entity (NE) ist eine Entität, oft ein Eigenname, die meist in Form einer Nominalphrase zu identifizieren ist. Named Entities können beispielsweise Personen wie „Nils Holgerson“, Organisationen wie „WHO“ oder Orte wie „New York“ sein. Named Entities können durch das Verfahren der Named Entity Recognition (NER) automatisiert ermittelt werden.

Opinion Mininig Unter Opinion Mining, oder Sentiment Analysis, versteht man die Analyse von Stimmungen oder Haltungen gegenüber einem Thema, durch die Analyse natürlicher Sprache. Das Opinion Mining gehört zu den Verfahren des Text Minings.

POS PoS steht für *Part of Speech*, oder „Wortart“ auf Deutsch. Das PoS- Tagging beschreibt die (automatische) Erfassung und Kennzeichnung von Wortarten in einem Text und ist of ein wichtiger Preprocessing-Schritt, beispielsweise für die Analyse von Named Entities.

Preprocessing Für viele digitale Methoden müssen die zu analysierenden Texte vorab „bereinigt“ oder „vorbereitet“ werden. Für statistische Zwecke werden Texte bspw. häufig in gleich große Segmente unterteilt (*chunking*), Großbuchstaben werden in Kleinbuchstaben verwandelt oder Wörter werden lemmatisiert.

Property Property steht für „Eigenschaft“, „Komponente“ oder „Attribut“. In der automatischen Annotation dienen konkrete Worteigenschaften wie Groß- und Kleinschreibung zur Klassifizierung von Wörtern oder Phrasen. Durch die Berücksichtigung solcher Eigenschaften in den Features eines Tools kann maschinelles Lernen bestimmter Phänomene umgesetzt werden. In der manuellen Annotation können als Properties auch Eigenschaften von Annotationen benannt werden.

Reintext-Version Die Reintext-Version ist die Version eines digitalen Textes oder einer Tabelle, in der keinerlei Formatierungen (Kursivierung, Metadatenauszeichnung etc.) enthalten sind. Reintext-Formate sind beispielsweise TXT, RTF und CSV.

TEI Die *Text Encoding Initiative* (TEI) ist ein Konsortium, das gemeinsam einen Standard für die Darstellung von Texten in digitaler Form entwickelt. Die TEI bietet beispielsweise Standards zur Kodierung von gedruckten Werken und zur Auszeichnung von sprachlichen Informationen in maschinenlesbaren Texten (siehe auch XML und Markup).

Text Mining Das Text Mining ist eine textbasierte Form des Data Minings. Prozesse & Methoden, computergestützt und automatisch Informationen bzw. Wissen aus unstrukturierten Textdaten zu extrahieren, werden als Text Mining zusammengefasst.

Type/Token Das Begriffspaar „Type/Token“ wird grundsätzlich zur Unterscheidung von einzelnen Vorkommnissen (Token) und Typen (Types) von Wörtern oder Äußerungen in Texten genutzt. Ein Token ist also ein konkretes Exemplar eines bestimmten Typs, während ein Typ eine im Prinzip unbegrenzte Menge von Exemplaren (Token) umfasst.
Es gibt allerdings etwas divergierende Definitionen zur Type-Token-Unterscheidung. Eine präzise Definition ist daher immer erstrebenswert. Der Satz „Ein Bär ist ein Bär.“ beinhaltet beispielsweise fünf Worttoken („Ein“, „Bär“, „ist“, „ein“, „Bär“) und drei Types, nämlich: „ein“, „Bär“, „ist“. Allerdings könnten auch vier Types, „Ein“, „ein“, „Bär“ und „ist“, als solche identifiziert werden, wenn Großbuchstaben beachtet werden.

Unicode/UTF-8 Unicode ist ein internationaler Standard, der für jedes Schriftzeichen oder Textelement einen digitalen Code festlegt. Dabei ist UTF-8 die am weitesten verbreitete Kodierung für Unicode-Zeichen. UTF-8 ist die international standardisierte Kodierungsform elektronischer Zeichen und kann von den meisten Digital-Humanities-Tools verarbeitet werden.

Web Mining Unter Web Mining versteht man die Anwendung von Techniken des Data Mining zur Extraktion von Informationen aus dem World Wide Web. Das Web Mining ist ein Teilbereich des Data Minings und zählt zu einem der wichtigsten Anwendungsgebiete für das Text Mining.

Wordcloud Eine *Wordcloud*, oder auch Schlagwortwolke, ist eine Form der Informationsvisualisierung, beispielsweise von Worthäufigkeiten in einem Text oder einer Textsammlung. Dabei werden unterschiedlich gewichtete Wörter, wie die häufigsten Wörter, i.d.R. größer oder auf andere Weise hervorgehoben dargestellt. Die horizontale/vertikale Ausrichtung und die Farbe der dargestellten Wörter hat meistens allerdings keinen semantischen Mehrwert.

XML XML steht für *Extensible Markup Language* und ist eine Form von Markup Language, die sowohl computer- als auch menschenlesbar und hochgradig anpassbar ist. Dabei werden Textdateien hierarchisch strukturiert dargestellt und Zusatzinformationen i. d. R. in einer anderen Farbe als der eigentliche (schwarz gedruckte) Text dargestellt. Eine standardisierte Form von XML ist das TEI-XML.

ZIP ZIP steht für ein Dateiformat (zip = engl. Reißverschluss), in welchem mehrere Einzeldateien verlustfrei, komprimiert zusammengefasst werden. ZIP-Dateien werden beim Öffnen entweder automatisch entpackt oder lassen sich per Rechtsklick extrahieren.