### **Textkorpus**

Das Textkorpus ist eine XML-Datei, die auf dem TEI-Standard basiert und das Wurzelelement <teiCorpus> hat.[[1]](#footnote-1) Die Texte innerhalb des Korpus können in verschiedenen Sprachen vorliegen, müssen jedoch gemäß den Open-Data-Anforderungen standardisiert und in Unicode kodiert sein. Der gezielte Zugriff auf die Texte aus verschiedenen Sprachen oder Gruppen kann anhand der im <TEI>-Element erfassten Metadaten erfolgen. Hierbei ist die Anwendung des xml:id-Attributs beim Element <Text> empfehlenswert. Dieses Attribut besteht aus vier Teilen, die Informationen über die Lizenzbedingungen[[2]](#footnote-2), über die Zugehörigkeit des Textes zu einem bestimmten Repertoire, die Sprache des Textes und die laufende Nummer des Textes unter anderen Materialien mit gleicher Zugehörigkeit zum Repertoire enthalten.

Folgende Beispiele

* *cc\_deu\_deu\_1* = 1. Text aus dem deutschen Repertoire in deutsche Sprache
* *cc\_deu\_eng\_1* = Derselbe 1. Text aus dem deutschen Repertoire in englische Sprache
* *cc\_deu\_eng\_2* = 2. Text aus dem deutschen Repertoire in englische Sprache usw.

Für die Benennung der Namen der Repertoires und Sprachen werden Abkürzungen gemäß dem ISO-639-3-Standard verwendet. Die Abkürzung cc am Anfang des Attributwertes weist auf die Freinutzungsmöglichkeit der Datei hin.

Im TEI-Korpus sind die Märchen einzeln in separaten TEI-Elementen erfasst. Für jedes Märchen liegt eine codierte oder zur Codierung bereitgestellte Textfassung vor. Die Basistexte befinden sich in verschiedenen Repositorien und sind untereinander sowie mit dem codierten Text entsprechend referenziert. Die Texte, die nicht den OpenData-Forderungen entsprechen, werden im Textkorpus in abgeleiteter Form dargestellt und mit bibliographischen Angaben zur primären Textquelle versehen.

Die Hauptaufgabe des Korpus besteht darin, eine standardisierte Segmentierung der Daten sowie einen gezielten Zugriff auf diese Daten zu gewährleisten. Das entwickelte Datenmodell ermöglicht es, Szenen im Text auseinander zu differenzieren und in ihr eingebettete Motive zu erfassen. Dafür sind das Element ***<seg>*** und in ihm eingebettete Tupel von Attributen ***a, b, c*** und ***d*** zuständig. Die optimale Anzahl von Motiven innerhalb einer Szene beträgt fünf. Dies bedeutet, dass für jedes Motiv jeweils zwei Attribute ***b*** und ***c*** erfasst werden können, sowie weitere zwei gemeinsame Attribute ***a*** und ***d*** für alle fünf Tupel. Wie bereits bekannt, liegt der Segmentation des Textes der Auftritt einer handlungstragenden Person zugrunde, dementsprechend alles, was bei diesem Auftritt passiert und abgesehen davon, ob das passierte mit einem oder mehrere c Elementen (d.h. Wortlaute) erfasst ist, kann nur ein einziges d Element (d.h. handlungstragende Person bzw. Personen) haben.

Ein Beispiel für die Annotation des 10. Abschnitts in unserem Text wäre wie folgt (N gilt für die Null):

*<seg*

*a1:ana = “a551“*

*b1:ana = “F“ c1:ana = “ Höfliches\_Verhalten“*

*b2:ana = “H“ c2:ana = “ Höfliches\_Verhalten“*

*b3:ana=“HF“ c3:ana=“Wache\_besänftigen“*

*b4:ana=“N“ c4:ana=“ N“*

*b5:ana=“N“ c5:ana=“ N“*

*d:ana=“ rHD\_rHF“ >*

*</seg>*

### **Künstlicher Assistent für die semiautomatische Annotation der Texte**

Ein zentraler Bestandteil der maschinellen Analyse von Märchen ist ein künstlicher Assistent mit Vorhersagefunktion.[[3]](#footnote-3) Dieser Assistent ist dafür zuständig, in einer umfangreichen Menge von Daten die Textabschnitte zu erkennen, die über die gesuchten inhaltlichen Eigenschaften verfügen.

Die Entwicklung eines solchen Assistenten erfordert die Erfüllung der folgenden drei Aufgaben:

1. Auswahl einer geeigneten Vorgehensweise für die maschinelle Textanalyse.
2. Anpassung dieser Vorgehensweise an die Forschungsfrage.
3. Umformulierung des gewonnenen Wissens über das Forschungsobjekt in die Sprache der ausgewählten Vorgehensweise.

Besonders wichtig bei der letzten Aufgabe ist die Anpassung zwischen den konzeptuellen und empirisch beobachtbaren Elementen. Eine Szene innerhalb der Geschichte wird dabei als empirisch beobachtbare Grenze für den Anfang bzw. das Ende eines Motivs betrachtet. Für die Maschine bedeutet dies eine Zeichenkette, bei der das kleinste erkennbare Element ein Wort und das größte ein Absatz ist. Auf inhaltlicher Ebene entspricht dies dem Textabschnitt mit veränderten Figuren, ihren Attributen, Ortschaften und Handlungen. Wir können zwar der Maschine beibringen, diesen Wechsel zu erkennen, jedoch müssen wir zuvor deutlich machen, welcher Zusammenhang zwischen den Roh-, Trainings-, Test- und Zieldaten besteht.

Der rohe Text, wie er uns gewöhnlich vorliegt, ist durch die folgenden explizit erkennbaren Zeichen wahrnehmbar: Wort, Satz, Absatz und Text. Die Grenzen der somit organisierten Textteile entsprechen jedoch nicht unbedingt dem von uns gesuchten Szenenwechsel im Text.

Die Aufbereitung des rohen Textes zu Trainingsdaten erfordert die Umwandlung der sogenannten Istzeichen in Sollzeichen. Dabei ist es oft notwendig, die Anzahl der Absätze im Text zu korrigieren. Die Absätze können entweder zusammengelegt oder in mehrere Teile aufgeteilt werden. In manchen Fällen kann auch ein einziger Satz einem solchen Absatz entsprechen.

Messdaten sind den Rohdaten gleich, sie enthalten bibliographische Metadaten und stellt wohlgeformte XML Daten dar. Zieldaten sind annotierten Messdaten. Diese können teilweise oder vollständig annotiert werden. Die bereits annotierten Teile der Messdaten werden bei der automatischen Analyse der Texte nicht miteinbezogen. Sie können allerdings als Trainingsdaten verwendet werden.

Für die automatische Erkennung der Motive und Episoden im Text verwenden wir den Algorithmus der logistischen Regression (LogisticRegression). Ein Modell, das von diesem Algorithmus gesteuert wird, versucht, Merkmale zu finden, die für die Erkennung der gesuchten Episoden (positiv etikettierte Datensätze) sowie ihrer Gegenteile (negativ etikettierte Datensätze) anwendbar sind, und ordnet sie auf beiden Seiten einer imaginären Trennlinie an.

Entscheidend für die Einbettung der Elemente in das Modell sind die nach den TF\*IDF-Werte skalierten Merkmale der positiv und negativ bewerteten Datensätzen. Vor der Berechnung der TF\*IDF-Werte findet die Datenbereinigung statt, wobei sogenannte Stoppwörter und andere Wörter, die für den Vergleich irrelevant sind, von Datensätzen entfernt werden.

Aufgrund der üblicherweise begrenzten Anzahl positiv etikettierter Trainingsdaten wird eine mehrfache Kreuzvalidierungsmethode angewendet. Diese Methode teilt die gesamten Trainingsdaten in gleichmäßige Abschnitte auf, die als Validierungs- und Trainingsdaten verwendet werden. Das trainierte Modell sollte nun in der Lage sein, die charakteristischen Merkmale (Wortformen) für positiv und negativ etikettierte Daten zu trennen und sie als eine sortierte Liste der entsprechenden Zeichengruppen darzustellen.

Das trainierte Modell wird durch die Vorhersagefunktion erweitert. [[4]](#footnote-4) Obwohl die Funktion in einigen Fällen scheitert, kann sie dennoch als zuverlässiges Werkzeug betrachtet werden. Der genaue Grund für das Scheitern der Vorhersagefunktion in einigen Fällen ist nicht eindeutig zu ermitteln. Es ist jedoch anzunehmen, dass dies teilweise an der Methode liegt, die Ähnlichkeit zwischen dem Modell und den Messdaten abzugleichen. Die Sigmoid-Funktion, die erfolgreich zur Berechnung der Distanz zwischen dem Modell und den Messdaten eingesetzt werden kann, könnte hierbei möglicherweise behindert werden, da der Datensatz oft beide Klassen (sowohl positiv als auch negativ) repräsentiert. In solchen Fällen könnte die angenommene Ähnlichkeit oder der Unterschied zwischen den verglichenen Inhalten fehlerhaft sein, da im geprüften Textteil sowohl negative als auch positive Merkmale gleichzeitig vorhanden sein können.

Unter Berücksichtigung des oben genannten Faktors scheint es viel effektiver zu sein, die Daten ausschließlich anhand der mit den höchsten Koeffizienten versehenen positiven Merkmale zu untersuchen. Dadurch werden die Datensätze nicht mehr über das Modell verglichen, sondern über das aus dem Modell extrahierte prototypische Metamotiv. Dieses funktioniert wie eine Suchabfrage an die Dokumentensammlung.

Ein Bild, das Screenshot, Reihe, Rechteck, Diagramm enthält.

Automatisch generierte Beschreibung

Abb. 6

Das Extrahieren der Merkmale erfolgt durch die Berechnung des Schwellenwertes für die Topklassifikationsmerkmale. Dieser Wert ist an der Stelle der Merkmal-Kette zu suchen, wo die Gewichte der positiven und negativen Merkmale bzw. Koeffizienten abgeglichen werden.

Nach dem graphischen Verfahren kann man diese Stelle wie folgt lokalisieren: Die Abbildung 6 zeigt die Schwingung zwischen den maximalen und minimalen Koeffizienten der negativ und positiv eingeschätzten Klassifikationsmerkmale. Die für beiden Kategorien geltende maximale Zahl der Merkmale (z) ist eine Variable und kann entsprechend der Situation angepasst werden, diesmal beträgt sie 200. Blaue Spalten stellen den positiven Merkmalen dar, die roten Spalten visualisieren hingegen die negativen Merkmale (Wegen der hohen Anzahl überlappen sich die Merkmale (siehe schwarze Wortwolke im unteren Bereich der Grafik). Für die Visualisierung der gesuchten Stelle wird die vorhandene Grafik an dem Treffpunkt von roten und blauen Spalten geteilt und die linke Hälfte so weit nach rechts verschoben, bis die kleinste rote Spalte unter der größten blauen Spalte platziert wird. Beachtet man die Größe von blauen und roten Spalten kann man leicht feststellen, an welcher Stelle sie übereinstimmen (s. Abb. 6.1).

Ohne graphische Darstellung kann man derselben Schwellenwert anhand folgender Formel berechnen:

Kommt die Gleichung zustande:

, *wobei ,*

so ist der Schwellenwert gleich *.*

Da die Topklassifikationsmerkmale nichts anderes darstellen als eine einfache Liste von positiv bewerteten Elementen bzw. Wörtern, ist ihre Bildung mit Hilfe einer einfachen Stufenfunktion möglich (entspricht der grünen Grafik auf der Abb. 6.1):

Ein Bild, das Screenshot, Reihe, Rechteck, parallel enthält.

Automatisch generierte Beschreibung

Abb. 6.1

Die Liste der auf diese Weise festgelegten Merkmale wird als prototypisches Metamotiv in einem Datensatz gespeichert und bei den Messdaten angelegt.

Die Erkennung des gesuchten Motivs in den Messdaten erfolgt durch die Klassifikation auf Basis der Kosinus-Ähnlichkeit. [[5]](#footnote-5) Dabei wird jeder Datensatz mit einem synthetischen Datensatz verglichen, der von uns oben als das prototypische Metamotiv definiert wurde.

1. Github: edadunashvili/VerMa/Textkorpus.xml [↑](#footnote-ref-1)
2. Dafür werden Abkürzungen cc für die freigegebene Daten und cr für die mit dem Copyright geschützte Texte. [↑](#footnote-ref-2)
3. GitHub: edadunashvili/VerMa/erthaos\_11.ipynb [↑](#footnote-ref-3)
4. GitHub: edadunashvili/VerMa/erthaos\_11.ipynb (Zellen #14-1). [↑](#footnote-ref-4)
5. W.o. Zellen # 16 – # 18. Als vorbildlich gilt die Anwendung von Needham, Mark 2016. [↑](#footnote-ref-5)