

Wissensvermittlungen im Drama annotieren

Annotationsguideline

Q:TRACK, Universität Stuttgart, Universität zu Köln

Version 0.9, Juli 2021

Inhaltsverzeichnis

1	Einleitung	2
2	Annotationstool	2
3	Allgemeines	2
4	Markables	2
5	Tagstruktur	2
6	Figuren	3
7	Figurenrelationen	5
7.1	Gerichtete Relationen	5
7.2	Ungerichtete Relationen	6
7.3	Erläuterung von Relationstags	6
8	Wissen über Wissen	7
9	Sonstiges	7
9.1	Bruder, Oheim & Co.	7
9.2	Automatische Schlüsse	7
10	Annotationen finalisieren	8

1 Einleitung

Das Projekt Q:TRACK befasst sich mit der Wissensvermittlung in deutschen Dramen. Genauer geht es um die Weitergabe von Wissen zu Familienrelationen: Wann sind den Figuren und dem Publikum welche Familienrelationen bekannt? Der Schlüsselmoment vieler Dramen besteht darin, dass eine Figur von einer ihr bisher unbekannten Familienrelation erfährt. Ein prominentes Beispiel ist der Moment, in dem Ödipus (Sophokles: König Ödipus) aufdeckt, dass er seinen Vater getötet und seine Mutter geheiratet hat. Um diese Wissensweitergabe zu modellieren, benötigen wir manuell erstellte Annotationen. In einem ersten Schritt wollen wir Textstellen identifizieren, an denen Wissen über Figurenrelationen vermittelt wird. Wie wir dabei vorgehen, wird im Folgenden erläutert.

2 Annotationstool

Als Annotationstool verwenden wir den CorefAnnotator, den man hier herunterladen kann: <https://github.com/nilsreiter/CorefAnnotator/>. Bitte nutzt immer die Version, die als „Latest Release“ markiert ist. Insbesondere die Beta-Versionen nutzen wir in der Annotation nicht. Wenn eine neue Version verfügbar ist, wird auf der Startansicht des CorefAnnotators darauf hingewiesen.

3 Allgemeines

Wir annotieren den Text des Dramas als Lesetext und nicht als Aufführungstext. Das Wissen aus der Figurentafel steht den Leser:innen also zur Verfügung und auch die Namen der Sprecher:innen werden als dem Publikum bekannt angenommen.

4 Markables

Annotiert werden alle Textstellen, an denen sich der Wissensstand zum Thema Figurenrelationen einer der beteiligten Figuren und/oder des Publikums verändert ODER an denen sich der Wissensstand über den Wissensstand einer (anderen) Figur ändert (z. B. das Publikum erfährt, dass A weiß, dass B und C Schwestern sind).

Idealerweise sollen möglichst kurze Textstellen annotiert werden, die dadurch sehr präzise angeben, an welcher Stelle die Wissensvermittlung stattfindet. Oft handelt es sich dabei um Sätze oder Teilsätze. Jedoch lässt sich die Wissensvermittlung nicht immer an einen genauen Satz binden. Wo unvermeidbar können deshalb auch längere Passagen mit Redeanteilen mehrerer Figuren annotiert werden.

5 Tagstruktur

Die annotierten Textstellen werden mit Tags versehen, die die folgenden Informationen zur Wissensvermittlung formalisieren:

1. Wer (oder was) ist die QUELLE des Wissens?
2. Wer ist das ZIEL des Wissens?
3. Welches WISSEN wird vermittelt?
4. optional: ATTRIBUTE, z. B. um Lügen zu markieren

Die Tags enthalten diese Informationen in einer festen Reihenfolge:

```
transfer(QUELLE, ZIEL, WISSEN, ATTRIBUTE)
```

Beispiel:

```
transfer(nathan, saladin, siblings(tempelherr, recha))
```

d. h. Nathan berichtet Saladin, dass der Tempelherr und Recha Geschwister sind. Die einzelnen Komponenten werden im Folgenden näher erläutert.

QUELLE. Die häufigste Quelle von Wissen sind andere Figuren, die die Information verbal vermitteln. In diesem Fall wird das Argument mit der ID der Figur gefüllt (siehe Abschnitt 6). Die Figur, die die Quelle des Wissens ist, muss nicht unbedingt in der Szene anwesend sein, wenn das Wissen z. B. medial durch einen Brief vermittelt wird. Bei Botenberichten ist im Normalfall die:der Sender:in des Boten Quelle des Wissens. Ausnahmen kommen vor, wenn die Boten selbst eine aktive Rolle im Drama spielen und z. B. die Nachricht verfälschen oder eigene Interessen damit verfolgen.

In manchen Fällen stammt das Wissen nicht von einer anderen Figur, sondern wird zum Beispiel durch ein Objekt oder eine Beobachtung ausgelöst. Für diese Art der Informationsquelle ist keine Formalisierung vorgesehen, sie kann deshalb in Anführungsstrichen frei formuliert werden, z. B.:

```
transfer("Handschrift in Buch", saladin, parent_of("Assad", tempelherr))
```

d. h. aus der Handschrift in dem Buch leitet Saladin ab, dass Assad der Vater des Tempelherrn ist. Wird als Quelle eine Figur angegeben, so bedeutet dies, dass die Figur das Wissen verbal geäußert hat. Non-verbale Wissensvermittlungen werden ebenfalls umschrieben, z. B. "optische Ähnlichkeit mit Assad".

ZIEL. Das Ziel von Wissen sind immer Figuren oder das Publikum. Zu ihrer Formalisierung bei der Annotation siehe Abschnitt 6. Eine Annotation als Ziel von Wissen setzt voraus, dass die entsprechende Information auch verstanden wurde.

WISSEN. Das Wissen, für das wir uns in diesem Annotationsprojekt interessieren, sind Familien- und Liebesrelationen zwischen den Figuren. Abschnitt 7 führt aus, welche Relationen auf welche Weise annotiert werden. Außerdem annotieren wir auf einer Metaebene das Wissen über das Wissen anderer Figuren, siehe Abschnitt 8.

ATTRIBUTE. In zusätzlichen Attributen können wir relevante Merkmale der Wissensvermittlung vermerken. Attribute sind optional. Folgende Attribute sind vorgesehen:

- **lie:** Mit diesem Attribut werden Lügen markiert. Die:Der Sender:in vermittelt in diesem Fall ein Wissen, das nicht mit ihrem:seinem eigenen übereinstimmt. Mit dem Attribut **lie** wird eine Aussage über die **QUELLE** des Wissens getroffen.
- **uncertain:** Mit diesem Attribut kann ausgedrückt werden, dass das vermittelte Wissen nicht gesichert ist, sondern es sich z. B. nur um eine Vermutung handelt.

```
transfer(tempelherr, nathan, relatives(tempelherr, recha), uncertain)
```

d. h. eine Aussage des Tempelherrn weckt in Nathan die Vermutung, dass der Tempelherr und Recha verwandt sein könnten. Mit diesem Attribut wird eine Aussage über das **ZIEL** des Wissens getroffen.

- **already-known:** Mit diesem Attribut wird markiert, dass nur die Quelle des Wissens der Ansicht ist, dass sie neues Wissen vermittelt, dieses dem Ziel aber bereits bekannt ist.

```
transfer(appiani, marinelli, engaged(appiani, emilia), already-known)
```

d. h. Appiani erzählt Marinelli, dass er mit Emilia verlobt ist, Marinelli weiß dies aber bereits.

Ergänzungen der Liste im Verlauf des Annotationsprojektes sind möglich.

6 Figuren

Figuren werden in den Tags mit ihrer DraCor-ID repräsentiert. Diese kann auf der Webseite von Ger-DraCor nachgeschlagen werden: <https://dracor.org/ger>. Im TEI-XML-Dokument zu jedem Drama werden die IDs am Anfang des Dokuments definiert. Alternativ ist zu jedem Drama unter den Downloads auch eine Figurenliste (List of characters) verfügbar.

DraCor-IDs stehen nur für die Figuren zur Verfügung, die im Drama auch selbst auftreten. Figuren, über die nur gesprochen wird, haben keine DraCor-ID. Für diese Figuren vergeben wir eigene IDs, die im

Normalfall dem am häufigsten für sie verwendeten Namen entspricht. Von uns vergebene Figuren-IDs, die nicht in den DraCor-Daten enthalten sind, setzen wir in Anführungszeichen, also **recha** und **nathan**, aber "**Assad**".

Darüber hinaus kommen Erwähnungen von Figuren ohne Namen vor, indem sie beispielsweise nur über ihre Relation zu einer anderen Figur erwähnt werden. Beispielsweise gibt Nathan zunächst nur bekannt, dass Recha einen Bruder hat:¹

NATHAN.

Besonders hat ein Bruder sich gefunden,

Bei dem Ihr um sie werben müßt.

→ `transfer(nathan, tempelherr, siblings(recha, SIBLING_OF[RECHA]))`

In solchen Fällen verwenden wir eine Variable, die – wie im Beispiel – die entsprechende Relation ausdrückt. Diese Variable wird in Großbuchstaben geschrieben. Sollte es in einem Drama mehrere Figuren geben, die über die gleiche Relation eingeführt werden, dann werden die Variablen zur Disambiguierung nummeriert. In *Die natürliche Tochter* von Goethe werden beispielsweise zwei Kinder des Herzogs zunächst ohne Namen eingeführt. Diese bezeichnen wir als **CHILD-1[HERZOG]** und **CHILD-2[HERZOG]**.

Im weiteren Verlauf des Textes wird die Identität der Figuren dann häufig aufgelöst. In *Nathan der Weise* stellt sich kurz darauf heraus, dass es sich bei Rechas Bruder um den Tempelherrn handelt. Diese Gleichsetzung wird dann mit der **identity**-Relation annotiert:

TEMPELHERR.

... Aber Rechas Bruder? Rechas Bruder ...

NATHAN.

Seid Ihr!

→ `transfer(nathan, tempelherr, identity(SIBLING_OF[RECHA], tempelherr))`

Die Auflösung der Variablen muss immer über die **identity**-Relation erfolgen. Im Beispiel soll deshalb nicht einfach `SIBLING(recha, tempelherr)` annotiert werden, weil der konkrete Bezug zur Variablen dann unklar bliebe.

Variablen können auch im Plural stehen, etwa **SIBLINGS[IPHIGENIE]**. Bei solchen Gruppen von Figuren muss die Auflösung der Variablen aber nicht unbedingt gleichzeitig erfolge. Im Beispiel der Iphigenie erfahren wir etwa zunächst, dass Elektra ihre Schwester und später, dass Orest ihr Bruder ist. Wir stellen die Verbindung zur Gruppe über das gleiche Schlüsselwort (**SIBLING**) her, anstelle des Plural-s erfolgt auch hier eine Nummerierung: `identity(SIBLING-1, elektra)`.

Manche Figuren werden in einem Drama nur am Rande erwähnt und haben dann keine weitere Bedeutung für den Text. Relationen dieser Figuren werden nicht annotiert. Konkret gilt das für Figuren, die

- selbst nicht im Drama auftreten und
- für die Handlung des Dramas irrelevant sind.

Letzteres kann ggf. erst nach Lesen des ganzen Dramas beurteilt werden. Die Figur Assad tritt in *Nathan der Weise* beispielsweise nicht selbst auf, ihre Relationen zu anderen Figuren sind für die Handlung aber trotzdem entscheidend.

Zusätzlich zum Wissensstand der Figuren interessieren wir uns auch für den Wissensstand des Publikums. Für das Publikum wird die ID **audience** verwendet. Die ersten Wissensbestände des Publikums entstehen üblicherweise durch die Figurentafel am Anfang des Textes. Wissensvermittlungen an das Publikum können, insbesondere am Anfang des Textes, relativ unmarkiert erfolgen. Wenn etwa eine Figur eine andere als *Bruder* anspricht, ist diese Relation für beide Figuren (normalerweise) schon vorher bekannt, für das Publikum aber unter Umständen eine neue Information.

`transfer("Figurentafel", audience, fosterparent_of(nathan, recha))`

In vielen Fällen erfahren mehrere Figuren oder das Publikum etwas gleichzeitig. In diesem Fall steht an der Stelle **ZIEL** eine Liste aus all denen Entitäten, für die das Wissen neu ist. Listen werden von eckigen Klammern umschlossen, die Elemente der Liste werden durch Kommata voneinander getrennt:

`transfer(nathan, [recha, tempelherr, saladin, sittah], siblings(tempelherr, recha))`

¹Wir annotieren also zunächst nur, dass wir von der Existenz einer solchen Figur wissen. Die Existenz mancher Figuren kann natürlich bereits per Common Sense angenommen werden, etwa dass eine Figur einen Vater hat. Zur formalen Einheitlichkeit mit Figurenrelationen, bei denen das nicht der Fall ist, werden solche Fälle trotzdem annotiert.

7 Figurenrelationen

Bei der Annotation der Figurenrelationen folgen wir diesem Muster:

```
relation(CHARACTER-A, CHARACTER-B)
```

CHARACTER-A und CHARACTER-B stehen jeweils für die beteiligten Figuren. Beispiel:

Nathan ist der Vater von Recha.
→ `parent_of(nathan, recha)`

Tabelle 1 fasst die zu annotierenden Relationen zusammen. Zu allen Relationen, für die dies inhaltlich sinnvoll ist, können außerdem Varianten mit den Präfixen **grand-**, **step-**, **foster-**, **god-** oder **ex-** sowie dem Suffix **-in-law** verwendet werden. Des Weiteren ist es möglich, alle Relationen bei Bedarf mit einem Ausrufezeichen vor der Relation zu negieren:

Nathan ist nicht der Vater von Recha.
→ `!parent_of(nathan, recha)`

Auch in den Relationen können an Stelle von CHARACTER-A und CHARACTER-B in Einzelfällen mehrere Figuren in Form einer Liste in eckigen Klammern stehen:

```
cousins([marschallin, der_baron], "Jörger")
```

A) Verwandtschaft	
Gerichtete Relationen	Ungerichtete Relationen
parent_of(PARENT, CHILD) child_of(CHILD, PARENT)	
	siblings(SIBLING-A, SIBLING-B)
aunt:uncle_of(AUNT:UNCLE, NIECE:NEPHEW) niece:nephew_of(NIECE:NEPHEW, AUNT:UNCLE)	
	cousins(COUSIN-A, COUSIN-B)
B) Partnerschaft	
Gerichtete Relationen	Ungerichtete Relationen
in_love_with(LOVER, TARGET)	lovers(LOVER-A, LOVER-B) couple(PARTNER-A, PARTNER-B) engaged(PARTNER-A, PARTNER-B) spouses(PARTNER-A, PARTNER-B)
widow:er_of(WIDOW:ER, DEAD-PARTNER)	
C) Identitäten	
Gerichtete Relationen	Ungerichtete Relationen
has_name(A, NAME)	
	identity(A, B)

Tabelle 1: Zu annotierende Figurenrelationen. Wo notwendig können die Präfixe **grand-**, **step-**, **foster-**, **god-** und **ex-** sowie das Suffix **-in-law** ergänzt werden.

7.1 Gerichtete Relationen

In gerichteten Relationen wird die Relation aus der Perspektive einer der beiden beteiligten Figuren benannt. Die Relation ist asymmetrisch: Wenn A der Vater von B ist, ist B nicht auch der Vater von A. Die Reihenfolge der Figuren ist deshalb wichtig. Wir folgen hier dem Muster „A steht in dieser Relation zu B“:

Nathan ist der Vater von Recha.
→ `parent_of(nathan, recha)`

Im Fall gerichteter Relationen gibt es oft eine komplementäre Relation. Die Vater-Tochter-Relation von Nathan und Recha kann z. B. durch `parent_of(nathan, recha)` und `child_of(recha, nathan)` ausgedrückt werden. Bei der Annotation jeder Erwähnung dieser Relation im Text orientieren wir uns an der Perspektive, die an der Textoberfläche eingenommen wird. In einem Drama können also durchaus beide Relationsformen vorkommen.

7.2 Ungerichtete Relationen

In ungerichteten Relationen gibt es keine bestimmte Reihenfolge der genannten Figuren A und B. Wenn A ein Geschwister von B ist, ist B auch ein Geschwister von A:

Recha und der Tempelherr sind Geschwister.
→ `siblings(recha, tempelherr)`

Bei ungerichteten Relationen spielt die Reihenfolge der Figuren in der Klammer keine Rolle.

7.3 Erläuterung von Relationstags

Viele der Relationstags sind selbsterklärend, zu den folgenden sind jedoch Erläuterungen angebracht:

Liebesrelationen. Bei Liebesrelationen werden mehrere Phänomene unterschieden: Bei den beiden Tags `lovers(LOVER-A, LOVER-B)` und `in_love_with(LOVER, TARGET)` geht es um eine Verliebtheit, die sowohl nur einseitig als auch gegenseitig sein kann.² Entscheidend für die Unterscheidung dieser beiden Tags ist die Frage, ob nur A B liebt (oder wir jedenfalls nur davon wissen → `in_love_with(LOVER, TARGET)`) oder ob ein gegenseitiges Verhältnis besteht (→ `lovers(LOVER-A, LOVER-B)`). Für Liebesrelationen im Sinne einer zustandegekommenen Beziehung zwischen den Figuren verwenden wir das Tag `couple(PARTNER-A, PARTNER-B)`. Diese Relation gibt es nur in einer ungerichteten, auf Gegenseitigkeit beruhenden Weise. Die genaue Natur der Beziehung spielt dabei keine Rolle, also ob es sich etwa um eine Affäre oder eine offizielle Beziehung handelt. Liebesrelationen werden ihrer Natur gemäß meistens weniger klar benannt als Verwandtschaften. Nach Möglichkeit annotieren wir Textstellen, die die Liebesrelation zweifelsfrei annehmen lassen. In vielen Fällen werden aber auch vage Aussagen annotiert werden müssen, die dem Publikum trotzdem in der Summe vermitteln, dass eine Liebesrelation vorliegt.

`has_name(A, NAME)`. Hierbei handelt es sich nicht um eine Relation im eigentlichen Sinne. Das Tag wird verwendet, wenn ein bisher unbekannter Name einer Figur eingeführt wird. Namen schreiben wir in Anführungsstrichen.

`transfer(tempelherr, nathan, has_name(tempelherr, "Curd von Stauffen"))`

d. h. der Tempelherr teilt Nathan mit, dass er den Namen Curd von Stauffen trägt.

`identity(A, B)`. Dieses Tag wird verwendet, wenn zwei Figuren sich als die gleiche herausstellen oder ihre Identität behauptet wird. Beispielsweise behauptet der Baron im *Rosenkavalier* dem Kommissar gegenüber, bei der vermeintlichen Kammerzofe handele es sich um seine Verlobte Sophie:

`transfer(baron, kommissar, identity(sophie, kammerzofe), lie)`

Das Tag kann ebenfalls verwendet werden, wenn eine Figur sich eine zweite Identität zulegt, indem sie sich beispielsweise verkleidet.

`transfer("Octavian verkleidet sich", [marschallin, audience], identity(kammerzofe, octavian))`

d. h. dadurch, dass Octavian sich verkleidet, erfahren die anwesende Marschallin und das Publikum, dass es sich bei der Kammerzofe um Octavian handelt.

Die häufigste Anwendung der `identity`-Relation ist die Auflösung von Variablen, wie in Abschnitt 6 beschrieben.

²Der Einfachheit halber werden hierunter auch z. B. Heiratsabsichten gefasst, bei denen die Motivation nicht-emotionaler Natur ist.

8 Wissen über Wissen

Zusätzlich zum unmittelbaren Wissen über Figurenrelationen ist an vielen Textstellen auch das Wissen über das Wissen anderer Figuren wichtig. In solchen Fällen kann an der Stelle WISSEN auch selbst wieder ein Wissens-Tag verwendet werden.

- Wenn es um Wissen über einen Wissenstransfer geht, wird hier wiederum das Prädikat `transfer()` verwendet.

```
transfer(recha, sittah, transfer(daja, recha, fosterparent_of(nathan, recha)))
```

d. h. Recha erzählt Sittah, dass sie von Daja erfahren hat, dass Nathan ihr Pflegevater ist.

- Handelt es sich um statisches Wissen, wird stattdessen das Prädikat `knowledge(WISSENDE:R, WISSEN, ATTRIBUTE)` verwendet. Das Argument zur Quelle des Wissens fällt hier weg.

```
transfer(daja, nathan, knowledge(recha, fosterparent_of(nathan, recha)))
```

d. h. Daja erzählt Nathan, dass Recha weiß, dass Nathan ihr Pflegevater ist.

Ebenfalls annotiert werden kann Wissen über Nichtwissen. Hierzu kann das Tag `!knowledge()` verwendet werden. In *Nathan der Weise* erfährt beispielsweise der Tempelherr von Daja, dass Recha nicht weiß, dass Nathan nicht ihr Vater ist:

```
transfer(daja, tempelherr, !knowledge(recha, !parent_of(nathan, recha)))
```

Wissen über Wissen muss nicht begleitend zu jeder Wissensannotation annotiert werden: Wenn Daja Recha erzählt, dass Nathan nicht Rechas Vater ist, dann weiß Daja anschließend, dass Recha das weiß (und andersherum weiß auch Recha, dass Daja das weiß und das Publikum weiß, dass beide es wissen). Diese Zusammenhänge sind logisch zu erschließen und werden nicht gesondert annotiert.

9 Sonstiges

9.1 Bruder, Oheim & Co.

Bezeichnungen wie *Bruder*, *Oheim* oder *Vetter* werden oft für Personen verwendet, die nicht im engeren Sinne dieses Verwandtschaftsverhältnis ausweisen. Wenn eine Figur eine andere z. B. als Vetter bezeichnet, ist deshalb oft nicht klar, ob das Verwandtschaftsverhältnis besteht oder nicht. Wir annotieren das deshalb zunächst mit dem Attribut `uncertain`. Wenn sich das wahre Verwandtschaftsverhältnis im Textverlauf klärt, wird die Annotation an entsprechender Stelle ohne das Attribut wiederholt.

9.2 Automatische Schlüsse

Aus vielen Relationen lassen sich weitere Relationen logisch ableiten. Wenn etwa gilt, dass `PARENT_OF(A,B)`, dann gilt auch `CHILD_OF(B,A)`. Letzteres muss deshalb nicht gesondert annotiert werden, sondern gilt ebenfalls sofort als bekannt.

Bei anderen Relationen sind die Schlüsse weniger deterministisch: Wenn gilt, dass `PARENT_OF(A,B)` und `PARENT_OF(C,B)`, so gilt (jedenfalls im Dramenkorpus) mit hoher Wahrscheinlichkeit auch `SPOUSES(A,C)`. Hierzu gibt es aber natürlich Ausnahmen.

Vorübergehende Lösung: Wir gehen per Default von `SPOUSES(A,C)` aus und betrachten alles dem Widersprechende als neue Information. Diese Regel ist ggf. zu überarbeiten, wenn mehr relevante Textstellen vorliegen.

Bei anderen Fällen gehen wir im Zweifelsfall davon aus, dass das Wissen nicht automatisch erschließbar ist und annotiert werden muss.

10 Annotationen finalisieren

Um die Korrektheit und Konsistenz der Annotationen sicherzustellen, empfehlen wir, nach Abschluss der Annotation folgende Checkliste durchzugehen:

- Werden alle Figuren mit DraCor-IDs auch darüber referenziert?
- Stehen alle anderen Figurenbezeichnungen in Anführungsstrichen?
- Werden alle Klammern am Ende geschlossen?
- Gibt es sonstige Tippfehler?
- Prüfe für jedes Wissen, das in den Annotationen vorkommt, wer dieses Wissen im Laufe des Stückes erlangt. Gibt es noch weitere Figuren, die dieses Wissen auch haben oder bekommen, für die das bisher nicht annotiert wurde?
- Prüfe insbesondere, ob das Publikum noch irgendwo als Ziel des Wissens ergänzt werden muss.