# Evidenzbasierte Definition von Spiel-Design-Elementen durch automatisierte Regelextraktion aus Spielanleitungen

Alexander Schneider
alexander.schneider@th-nuernberg.de
Technische Hochschule Nürnberg - Georg Simon Ohm
Nürnberg, Bayern, Deutschland

## ZUSAMMENFASSUNG

Die Anwendung von Entwurfsmustern zur Lösung wiederkehrender Probleme hat sich in der Praxis bewährt. Auch in der Gamification werden Muster in Form von Spielelementen eingesetzt, um Prozesse motivierender zu gestalten. Die dafür genutzten Muster müssen jedoch gefunden und definiert werden. Es gibt zwar bereits Sammlungen von Spiel-Design-Elementen, aber die Forschung kann nicht als abgeschlossen betrachtet werden, da jedes Jahr neue Spiele entwickelt werden, in denen es möglicherweise neue Elemente zu entdecken gibt. Ein empirischer Ansatz wird vom Projekt EMPAMOS verfolgt, das Spielanleitungen von Gesellschaftsspielen nach Spiel-Design-Elementen durchsucht. Werden in verschiedenen Spielen Kandidaten für Muster gefunden, werden die Textstellen gesammelt und von Fachkräften diskutiert. Am Ende des Prozesses steht die Definition eines Spiel-Design-Elementes. Diese Arbeit stellt einen Ansatz vor, der Fachkräfte für Spiel-Design-Elemente bei der Suche nach neuen Elementen unterstützt, in dem aus den gefunden Textstellen eine möglichst allgemeingültige Definition für das jeweilige Spiel-Design-Element generiert werden soll.

## SCHLAGWÖRTER

Game Design Pattern, Explainable AI, NLP, Concept Analysis, Inductive Logic Programming

## 1 EINLEITUNG

Die Suche nach Spiel-Design-Elementen und deren Beschreibung war bereits Thema von Forschungsarbeiten. Als Ergebnis dieser Arbeiten entstanden gut dokumentierte Listen von Elementen [5][21], die z.B. auch in der Gamification Anwendung finden. Was genau unter Gamification zu verstehen ist und welche Elemente dafür genutzt werden ist nicht eindeutig festgelegt [11]. Im Rahmen dieser Arbeit wird unter Gamification die Anwendung von Spiel-Design-Elementen in einem spielfremden Kontext verstanden [3].

Das Forschungsprojekt EMPAMOS [23] wurde gestartet, um die Suche und Beschreibung von Spiel-Design-Elementen auf rein empirischer Basis durchzuführen. Bislang wurden in dem Projekt rund 100 Spiel-Design-Elemente [25] identifiziert, indem die Texte von Brettspielanleitungen von Fachkräften für Gesellschaftsspiele systematisch nach wiederkehrenden Mustern für neue Elemente durchsucht wurden. EMPAMOS ist eine Kooperation zwischen der Technischen Hochschule Nürnberg und dem Deutschen Spielearchiv der Stadt Nürnberg [20], das mit über 40.000 Gesellschaftsspielen über die weltweit größte Sammlung Deutscher Gesellschaftsspiele verfügt. Für das Projekt wurden bislang rund 8300 Anleitungen digitalisiert.

Bei der Suche nach neuen Spiel-Design-Elementen werden zuerst neue Muster-Kandidaten identifiziert, für die im Anschluss in den vorhandenen Spielanleitungen nach weiteren Textstellen gesucht wird, die als empirische Evidenzen belegen, ob es sich um ein Spiel-Design-Element oder nur um eine idiosynkratische Einzelfalllösung handelt. Nur wenn sich nachweisen lässt, dass das Element auch in mindestens 20 Spielen wiederkehrend eingesetzt wird, wird von einem Spiel-Design-Element als Muster gesprochen und es als Lösung für spezifische Entwurfsprobleme auf Basis der Mustertheorie von Alexander et al. [2] interpretiert.

Für jedes neue Spiel-Design-Element ist dabei in einer Datenbank vermerkt:

- In welchem Spiel das Element gefunden wurde
- Die Textstelle, die den Einsatz des Spiel-Design-Elements belegt
- Der Absatz vor und nach der relevanten Textstelle

Anschließend wird ausgehend von den empirischen Textevidenzen nach einer möglichst allgemeingültigen Definition für jedes Spiel-Design-Element gesucht, mit dessen Hilfe es möglich ist, die Elemente anhand ihrer spezifischen Eigenschaften trennscharf voneinander zu unterscheiden. Lässt sich keine allgemeingültige Definition für ein Spiel-Element finden oder überschneidet sich eine Definition mit der eines anderen Elements, dann ist das ein Zeichen, dass entweder die gefundenen Elemente noch genauer definiert und abgegrenzt werden müssen oder die sich überschneidenden Elemente eigentlich ein zusammengehörendes Element bilden.

Diese Arbeit stellt ein neues Verfahren vor, das die Fachkräfte bei der Suche nach allgemeingültigen Definitionen für Spiel-Design-Elemente unterstützt, indem die gefundenen Evidenzen automatisch zu einem formalen Regelsatz verdichtet werden. Dieser induktiv gewonnene Regelsatz kann als Ausgangspunkt sowohl für die Definition neuer Spiel-Design-Elemente als auch für die Suche nach neuen Evidenzen für bereits bekannte Spiel-Design-Elemente dienen.

## 2 PROBLEMSTELLUNG

Die Suche nach einer möglichst allgemeingültigen Definition für ein Spiel-Design-Element geht über die reine Klassifikation eines

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
*Veröffentlicht durch die Gesellschaft für Informatik e.V.*
*in K. Marky, U. Grünefeld & T. Kosch (Hrsg.):*
*Mensch und Computer 2022 – Workshopband, 04.-07. September 2022, Darmstadt*
© 2022 Copyright held by the owner/author(s).
https://doi.org/10.18420/muc2022-mci-ws08-440

bekannten Elements hinaus. Mit Hilfe von Positiv- und Negativbeispielen für Brettspielanleitungen, die ein gesuchtes Element enthalten bzw. nicht enthalten, wurde eine Support Vector Maschine (SVM) trainiert, die zuverlässig erkennt, ob in einem gegebenen Text ein gesuchtes Spiel-Design-Element enthalten ist [25]. Im Falle einer SVM zur Klassifikation von Spiel-Design-Elementen in Texten lassen sich Rückschlüsse auf die Art eines Elementes ziehen, indem man die von der SVM erstellten Wortlisten auswertet, deren Wörter oder Wortkombinationen besonders stark den Einsatz oder die Abwesenheit eines Elements vorhersagen. Mit LIME [15] existiert ein Verfahren, dass diese Informationen unabhängig vom verwendeten Modell erzeugen kann. Im Falle von Textklassifikation entstehen so Wortlisten, die bei der Entscheidung als besonders wichtig eingestuft wurden. Erst durch die Interpretation von menschlichen Fachkräften können aus diesen Wörtern semantische Bezüge zu beschriebenen Spiel-Design-Elementen hergestellt werden. Für die Ableitung einer Definition sind diese Wortlisten jedoch ungeeignet, da sie nur eine Menge häufig benutzter Wörter darstellen, aber diese semantisch und grammatikalisch nicht miteinander in Beziehung setzen und keine Verallgemeinerungen erzeugen.

Die Definitionen der Spiel-Design-Elemente sind die Grundlage für die Suche nach weiteren Evidenzen, die die Mustergültigkeit des Elements belegen und die Voraussetzung dafür, dass mehrere Fachkräfte gleichzeitig im Korpus auf die Suche gehen können. Nur wenn alle beteiligten Fachkräfte auf Grundlage derselben Definition ihre Bewertungen zu den Spiel-Design-Elementen vornehmen, kann eine objektive, d.h. intersubjektiv nachvollziehbare Mustersprache für Spielelemente entstehen. Die Definitionen werden im Laufe der Entwicklung der Mustersprache regelmäßig mit neu gefundenen Elementen überprüft und ggf. überarbeitet. Nach jeder Überarbeitung ist es notwendig, die Definition mit den bestehenden Definitionen abzugleichen, um mögliche Überschneidungen von Elementen aufzudecken. So enthielt zum Beispiel die Definition für das Spiel-Design-Element „Rolle" ursprünglich den folgenden Aspekt: „Spieler übernehmen Rollen, die sich in ihren Zielen und Fähigkeiten unterscheiden". Nachdem jedoch Spiele gefunden wurden, die den Spielern zwar besondere Fähigkeiten zuordnen, sie aber keine individuellen Ziele verfolgen lassen, wurde der Ziel-Aspekt aus der Rollendefinition entfernt und als Basis für ein neues Spiel-Design-Element namens „Mission" verwendet. Nach der Refaktorisierung der beiden Elemente, müssen diese selbst gegen den vorhandenen Datenbestand abgeglichen werden. Dieser Vorgang ist bisher nicht automatisiert und muss manuell vorgenommen werden.

Diese Formulierungen und Redefinitionen werden von Fachkräften vorgenommen, die neue Elemente diskutieren oder bestehende Elemente angesichts neuer Evidenzen überarbeiten. Die Fachkräfte kommen aus dem Spielearchiv oder sind Studierende, die für die Suche und Musterbeschreibung geschult werden. Eine maschinelle Unterstützung findet hierbei nur bei der Suche von Textstellen mit Hilfe einer Volltextsuchmaschine statt. Die Interpretation und Auswertung der Textstellen obliegt jedoch allein den menschlichen Fachkräften, was die evidenzbasierte Suche und die Definition von Spiel-Design-Elementen aufwändig und fehleranfällig macht. Die Volltextsuche findet nur exakte Textstellen und dies erfordert Geschick beim Formulieren von Suchanfragen, um weitere Kandidaten zu finden bzw. nicht zu übersehen. Die Datenbasis wächst jedes Jahr mit dem Erscheinen neuer Gesellschaftsspiele, in denen sich weitere Textbeispiele für bekannte oder neue Elemente verbergen können.

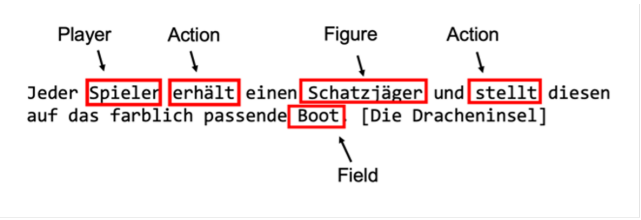


**Abbildung 1: Beispiel für Aktionen und Beziehungen von Objekten [17] für Spiel-Design-Element „Avatar" in „Die Dracheninsel" [24]**

Für jedes Spiel-Design-Element stehen in der Datenbank von EMPAMOS eine hohe zweistellige Anzahl von Beispielsätzen aus den annotierten Spielanleitungen zur Verfügung. Für das Training eines Künstlichen Neuronalen Netzwerkes (KNN) ist diese Anzahl aber zu gering, um akzeptable Erkennungsraten zu erhalten. Um Trainings- und Testdaten in ausreichender Menge und Qualität zu erhalten, wäre ein langwieriger und aufwändiger Annotationsprozess notwendig. Auch würde sich wieder das Problem, das im letzten Absatz beschrieben wurde, einstellen, dass sich aus den erzeugten Black-Box Modellen nicht direkt eine für uns Menschen verständliche Definition des Spiel-Design-Elements ableiten lässt.

Es wird eine Lösung benötigt, die in der Lage ist, aus den von menschlichen Fachkräften gefundenen Textbeispielen zu Spiel-Design-Elementen eine Unterstützung bei der Formulierung von allgemeingültigen Definitionen zu bieten. Diese Definitionen müssen von den Fachkräften ohne zusätzliches technisches Wissen zu verstehen sein. Um den menschlichen Aufwand für die Annotationen gering zu halten, muss eine Lösung auch mit wenigen Beispielen auskommen, um eine Definition zu formulieren und selbstständig weitere Evidenzen zu finden.

## 3 ARBEITSHYPOTHESE

Die zu einem vermuteten Spiel-Design-Element gesammelten Textbeispiele beschreiben alle denselben Sachverhalt. Sie sind im konkreten Anwendungsfall eines bestimmten Spieles die textuelle Repräsentation eines Musters. Wie bereits in [17] angenommen, wird für diesen Arbeit die Hypothese übernommen, dass „jedes Spiel-Design-Element durch einen abstrakten Satz von Aktionen und Beziehungen zwischen Objekten beschrieben werden kann. Lassen sich diese Aktionen und Beziehungen identifizieren, bilden sie die Basis einer formalen Beschreibungssprache für Spiel-Design-Elemente." [17].

In Abbildung 1 ist ein Beispiel dargestellt, das für wichtig erachtete Aktionen (Verben) und Objekte (z.B. Spielfiguren) hervorhebt. Der Satz ist der Brettspielanleitung „Die Dracheninsel" [24] entnommen und wurde als Evidenz für das Spiel-Design-Element „Avatar" annotiert. Konkret bekommt („erhält") eine Entität „Spieler" eine Figur („Schatzjäger"), die ihn auf dem Spielfeld repräsentiert („stellt").

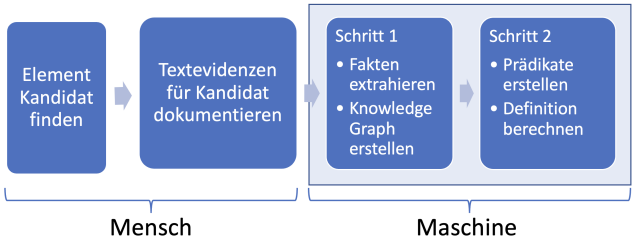


**Abbildung 2: Übersicht des Ablaufs des Verfahrens**

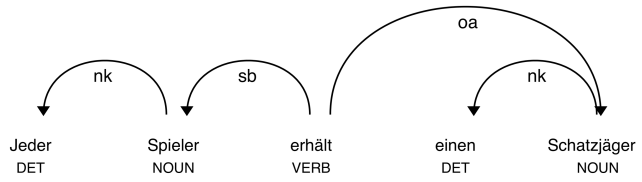


**Abbildung 3: Beispiel linguistische Merkmale DEP und POS. Beziehungen im DEP: nk = Substantiv Kernelement, sb = Subjekt, oa = Akkusativobjekt. Bezeichnungen POS: DET = Bestimmungswort, NOUN = Substantiv, VERB = Verb [22].**

## 4 LÖSUNGSANSATZ

Der geplante Lösungsansatz ist inspiriert von einer Methode zur verbalen Erklärung von Konzepten in Bildern von Rabold [14]. Ziel ist in dieser Arbeit nicht, die Ausgabe von LIME [15] um eine verbale Erklärung anzureichern, sondern eine möglichst allgemeingültige Definition für Spiel-Design-Elemente zu finden. Dafür soll das Konzept symbolische Regeln aus Bildern zu extrahieren und für dessen Erklärung zu nutzen, auf Texte übertragen werden. Es soll daher, analog zur Erklärung aus [14], dass Fenster in Türmen räumlich übereinander sind, mit Hilfe von Induktiver Logikprogrammierung (ILP) aus Textbeispielen eine allgemeine Beschreibung von Spiel-Design-Elementen abgeleitet werden.

Nachdem von Fachkräften die benötigten Textbeispiele für ein Spiel-Design-Element gesammelt wurden, geschieht die maschinelle Verarbeitung in zwei Schritten (siehe Abbildung 2). In Schritt eins werden aus den Textevidenzen für Spiel-Design-Elemente die benötigten Informationen extrahiert und in Schritt zwei in ein formales Logiksystem überführt, das Vorschläge für die Definitionen der Elemente induktiv ableitet. Beide maschinellen Verfahrensschritte werden im Folgenden vorgestellt.

### 4.1 Schritt 1: Extraktion von Fakten und Erstellung des Knowledge Graphen

Die Extraktion der Fakten findet in zwei Verarbeitungsphasen statt: der Vorbereitungsphase und der Extraktionsphase. Für die geplante Extraktion von Aktionen und Objekten wird ein geparster Korpus benötigt, der die Beziehungen der Wörter im Satz untereinander beschreibt. In der Vorverarbeitungsphase werden die Texte mit Hilfe eines Natural Language Processing (NLP) Frameworks mit semantischen Informationen angereichert. In Abbildung 3 ist eine entsprechende Anreicherung dargestellt, in der ein Dependency Tree (DEP) die Beziehungen zwischen den Wörtern darstellt und den Part-of-Speech (POS) Annotationen , die die Art der Wörter beschreiben.

Im Rahmen dieser Arbeit werden aus dem geparsten Korpus folgende Fakten extrahiert:

- Prädikate, bestehend aus Verben mit zugehörenden Subjekten bzw. Objekten
- Substantive mit Adjektiven
- Verben mit Adverbien und zugehörenden Hilfsverben
- Negationen

Die extrahierten Fakten werden in einen Knowledge Graph [4] überführt. Der entstehende Knowledge Graph enthält in den Knoten die gefundenen Objekte und Aktionen aus einer Textevidenz. An den Kanten werden die Beziehungen zwischen den Knoten gespeichert. Ein Beispiel ist in Abbildung 4 für die Textevidenz „Der Spieler mit den meisten Punkten gewinnt das Spiel" für das Spiel-Design-Element „Siegpunkte" aus dem Spiel „Zooloretto" [16] dargestellt. Der Knoten „gewinnen" hat unter anderem eine Beziehung „SUBJECT", die auf den Knoten verweist, der angibt wer oder was gewinnt. In diesem Fall ist es ein „Spieler".

Die getroffene Auswahl der extrahierten Fakten stellt einen Startpunkt dar, mit dem in einer ersten Iteration im Forschungsprozess versucht wird, die Arbeitshypothese (siehe Abschnitt 3) als Proof-of-Concept zu beweisen und die damit gewonnenen Ergebnisse zu evaluieren. Nach der Evaluationsphase, abhängig von den erzielten Ergebnissen, können weitere Fakten hinzugezogen werden, die sich als wichtig herausstellen, bzw. vorhandene Fakten wegfallen.

### 4.2 Schritt 2: Induktive Logikprogrammierung

Induktive Logikprogrammierung ist eine maschinelle Lerntechnik, die eine definitorische Hypothese aus positiven und negativen Beispielen aufbauen kann [12]. Dazu müssen sowohl die Beispiele als auch ein Hintergrundwissen vorgegeben werden. Die Repräsentation des Wissens besteht aus einer eingeschränkten Form der Prädikatenlogik. Ein- und Ausgaben erfolgen, im Gegensatz z.B. zu KNNs, im selben formalen Notationssystem. Die Form der Beschreibung der Regeln erlaubt es auf einfache Weise verbale Beschreibungen zu generieren [18]. So können die positiven und negativen Beispiele, das Hintergrundwissen und die generierten Hypothesen Domänenfachkräften ohne weitere Spezialisierungen verständlich dargestellt werden.

Die Prädikate für das Hintergrundwissen und die positiven und negativen Evidenzen, werden aus einem wie in Abbildung 4 dargestellten Graph gewonnen. Als positive Evidenzen werden nur die Graphen verwendet, die aus Textbeispielen gewonnen wurden, die den Einsatz des Spiel-Design-Elements belegen, für das eine Definition gesucht wird. Die negativen Beispiele stammen aus den Textbeispielen, die anderen Spiel-Design-Elementen zugeordnet wurden. Ein Beispiel für ein Fakt, das aus dem linken Teilgraphen in Abbildung 4 stammt, würde demnach folgendermaßen aussehen: `contains(Example, A), contains(Example, B), action(A, gewinnen), element(B, spieler)`. Als Notation wurde die Aussage in Prolog-Syntax dargestellt, eine für ILP-Systeme häufig genutzte logische Programmiersprache. Das Beispiel beschreibt eine Textevidenz, die die Elemente A und B enthält, wobei A eine Aktion „gewinnen" und B ein Element „spieler" ist.

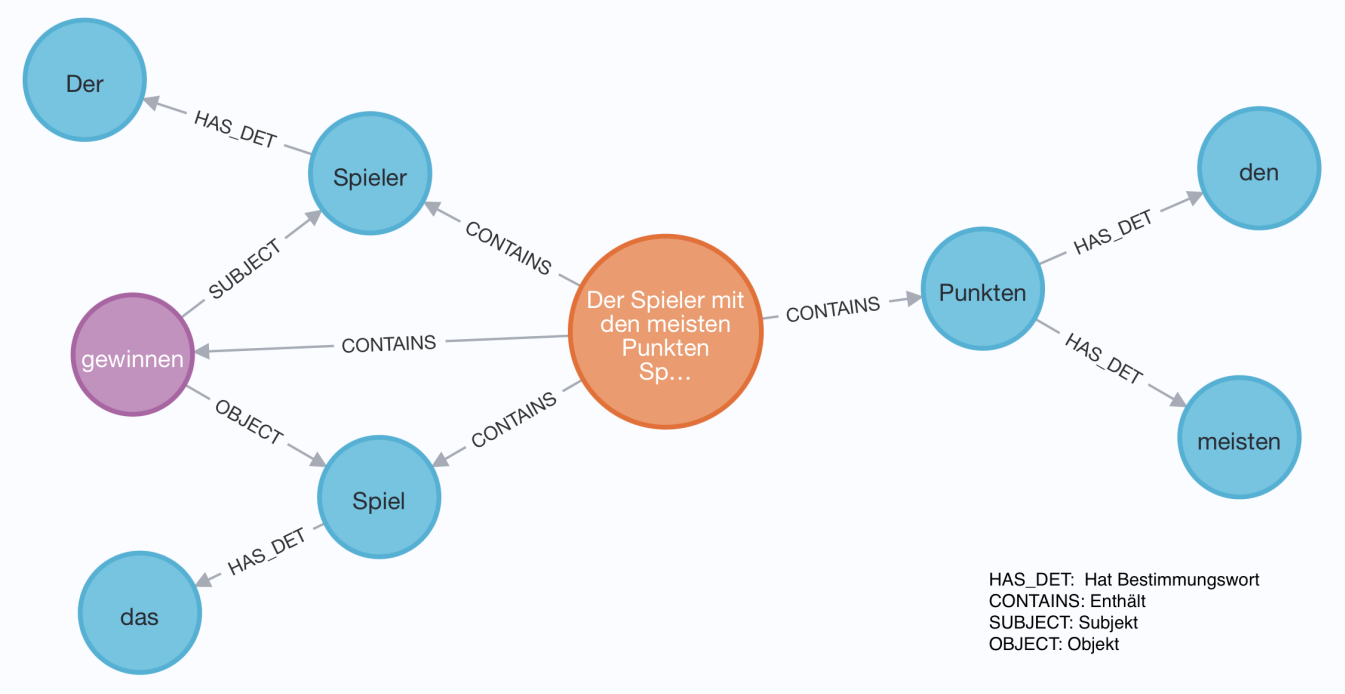


**Abbildung 4: Extrahierter Knowledge Graph aus einem Textbeispiel aus „Zooloretto" [16] für das Spiel-Design-Element „Siegbedingung".**

Aus den Beispielen und dem Hintergrundwissen wird ein ILP-Algorithmus versuchen, eine Theorie zu induzieren, die möglichst alle positiven Evidenzen ein und möglichst alle negativen ausschließt. Bezogen auf das Beispiel im letzten Absatz würde eine abgeleitete Hypothese für Siegbedingung aus demselben Code bestehen, da keine weiteren Beispiele vorhanden sind. Der Hypothese nach muss jede Beschreibung einer Siegbedingung einen „Spieler" und die Aktion „gewinnen" enthalten.

## 5 AKTUELLER STAND

Bereits implementiert wurde die Extraktion von Fakten aus 1535 Textevidenzen aus 33 Spielanleitungen für 34 Spiel-Design-Elemente. Auf Basis der Definitionen dieser Spiel-Design-Elemente wurden mit Hilfe der Annotationsplattform Inception [10] in Spielanleitungen Evidenzen und Spielelemente annotiert. Die annotierten Texte wurden mit Hilfe des NLP Frameworks spaCy [6] in einen geparsten Korpus überführt. Zur Extraktion der Fakten aus Abschnitt 4.1 wurden Semgrex Pattern [1] verwendet, deren Ausgaben zur Weiterverarbeitung in einer Graph Datenbank [13] gespeichert wurden.

Die vollständige Implementierung für die Evaluation des Verfahrens ist noch nicht abgeschlossen. Es fehlt noch der letzte Schritt zur induktiven Ableitung der Definitionen für Spiel-Design-Elemente. Die aktuell verwendete ILP-Implementierung ist Aleph [19].

## 6 PROBLEME

Der wichtigste Punkt, der noch aussteht, ist die Umsetzung der induktiven Verarbeitung der extrahierten Fakten aus den gesammelten Evidenzen für die bislang manuell beschriebenen Spiel-Design-Elemente und die Evaluation der Ergebnisse. Aus den bisherigen Ergebnissen aus der NLP Vorverarbeitung sind noch einige Probleme offen, wie z.B. die Coreference Resolution für deutsche Sprache. Das betrifft z.B. Auflösung von Pronomen, wenn „ein Spielstein" im weiteren Satzverlauf nur noch mit „er" referenziert wird. Das entsprechende Modell beherrscht nur englisch, aber an einer Erweiterung für die deutsche Sprache wird aktuell gearbeitet [8].

Die Regeln zum Extrahieren von Fakten aus den Texten enthalten noch nicht alle grammatikalischen Konstrukte. So fehlen zurzeit Konjunktionen und Disjunktionen von Satzteilen. Konstrukte der Form A und B bzw. A oder B sind noch nicht beschreibbar. Gleiches gilt für bedingte Satzteile der Form Wenn A dann B, wie „das Spiel endet, wenn eine 6 gewürfelt wird".

Die Extraktion der Fakten beruht auf dem deutschen Sprachmodell de_dep_news_trf für spaCy [9], das eine Genauigkeit von ca. 94% im Dependency Tree Parsing aufweist, aber dennoch mit einem Fehler von ca. 6% behaftet ist. Daher werden in einigen Fällen Fakten nicht oder fehlerhaft extrahiert.

## 7 FAZIT UND AUSBLICK

Bei erfolgreicher Evaluation bietet der Ansatz eine Möglichkeit aus Textrepräsentationen von Spiel-Design-Elementen automatisiert den Vorschlag für eine alle Elemente verallgemeinernde Definition zu finden. Es wird erwartet, dass diese Definition auf Grund der in

Abschnitt 6 dargestellten Probleme Lösungen enthalten wird, die von Menschen als falsch eingestuft werden.

Eine Lösung zu erzeugen, die trotz fehlerbehafteter Sprachverarbeitung und möglicherweise falsch annotierten Beispielen den Erwartungen eines Menschen auf Anhieb entspricht, ist vermutlich in naher Zukunft nicht realisierbar. Nutzt man das Verfahren jedoch als Baustein in einem Mensch-Maschine-System, bei dem Mensch und Maschine ihre jeweiligen Stärken ausspielen können, so wird die Arbeit für Fachkräfte effektiv und effizient. Die Maschine kann auf große Datenmengen sehr schnell Regelsätze anwenden und neue Definitionen erzeugen und vorschlagen. Eine Fachkraft kann die Definition bewerten und Korrekturen vornehmen, in dem das Hintergrundwissen oder die extrahierten Fakten angepasst werden. Wird z.B. in der Domäne der Brettspiele ein Ortsname nicht als Feld auf dem Brett als Fakt erkannt, so kann ein Mensch dies im Hintergrundwissen korrigieren und die Maschine mit dem verbesserten Wissen eine neue Definition des bearbeiteten Spiel-Design-Elements präsentieren.

Die Beschreibungen der Maschine zu den gefundenen Definitionen können in eine verbale Repräsentation umgewandelt werden und sind somit für einen Menschen verständlich. Es erfordert keine speziellen Schulungen zur Interpretation der Ergebnisse. Eingebettet in ein entsprechend gestaltetes interaktives System zur Arbeit mit Texten lässt sich ein Companion-System [7] entwerfen, um im Speziellen nach Spiel-Design-Elementen zu forschen. Im Allgemeinen könnte das System auch erweitert werden, um nach Mustern jeglicher Art zu forschen, solange sich diese sich in Prädikatenlogik für das zu Grunde liegende ILP System formulieren lassen.

## DANKSAGUNGEN

Das Forschungsvorhaben wird im Rahmen einer kooperativen Promotion mit der Universität Bamberg von der Technischen Hochschule Nürnberg finanziert.

## LITERATUR

[1] Nathanael Chambers, Daniel Cer, Trond Grenager, David Hall, Chloe Kiddon, Bill MacCartney, Marie-Catherine de Marneffe, Daniel Ramage, Eric Yeh, and Christopher D. Manning. 2007. Learning Alignments and Leveraging Natural Logic. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Association for Computational Linguistics, Prague, 165–170.

[2] Christopher Alexander, Sara Ishikawa, and Murray Silverstein. 1977. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, New York.

[3] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. 2011. From Game Design Elements to Gamefulness: Defining "Gamification". In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments (MindTrek '11)*. ACM, New York, NY, USA, 9–15. https://doi.org/10.1145/2181037.2181040

[4] Lisa Ehrlinger and Wolfram Wöß. 2016. Towards a Definition of Knowledge Graphs. In *Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems - SEMANTiCS2016 and the 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS'16) Co-Located with the 12th International Conference on Semantic Systems (SEMANTiCS 2016), Leipzig, Germany, September 12-15, 2016 (CEUR Workshop Proceedings, Vol. 1695)*, Michael Martin, Martí Cuquet, and Erwin Folmer (Eds.). CEUR-WS.org, Leipzig.

[5] Geoffrey Engelstein and Isaac Shalev. 2019. *Building Blocks of Tabletop Game Design: An Encyclopedia of Mechanisms*. Taylor & Francis, Boca Raton, FL.

[6] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. Zenodo. https://doi.org/10.5281/zenodo.1212303

[7] Thilo Hörnle, Michael Tornow, Frank Honold, Reinhard Schwegler, Ralph Heinemann, Susanne Biundo, and Andreas Wendemuth. 2017. Companion-Systems: A Reference Architecture. In *Companion Technology*, Susanne Biundo and Andreas Wendemuth (Eds.). Springer International Publishing, Cham, 449–469. https://doi.org/10.1007/978-3-319-43665-4_22

[8] Hugging Face. 2021. NeuralCoref 4.0: Coreference Resolution in spaCy with Neural Networks. Hugging Face.

[9] Hugging Face. 2022. spaCy Modell De_dep_news_trf v3.3.0. https://huggingface.co/spacy/de_dep_news_trf.

[10] Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Santa Fe, USA, 5–9.

[11] Athanasios Mazarakis. 2021. Gamification Reloaded: Current and Future Trends in Gamification Science. *i-com* 20, 3 (Dec. 2021), 279–294. https://doi.org/10.1515/icom-2021-0025

[12] Stephen Muggleton. 1991. Inductive Logic Programming. *New Generation Computing* 8, 4 (Feb. 1991), 295–318. https://doi.org/10.1007/BF03037089

[13] Neo4j. 2022. Neo4j: Graphs for Everyone. Neo4j.

[14] Johannes Rabold, Hannah Deininger, Michael Siebers, and Ute Schmid. 2019. Enriching Visual with Verbal Explanations for Relational Concepts – Combining LIME with Aleph. arXiv:1910.01837 [cs, stat]

[15] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *arXiv:1602.04938 [Cs, Stat]*. Association for Computing MachineryNew YorkNYUnited States, San Francisco California USA, 1135–1144. arXiv:1602.04938 [cs, stat]

[16] Michael Schacht. 2007. Zooloretto.

[17] Alexander Schneider. 2021. Auf Dem Weg Zur Automatischen Erkennung Und Beschreibung von Spiel-Design-Elementen in Brettspielanleitungen.

[18] Michael Siebers and Ute Schmid. 2019. Please Delete That! Why Should I? : Explaining Learned Irrelevance Classifications of Digital Objects. *Künstliche Intelligenz : KI, 33* 33, 1 (2019), 35–44. https://doi.org/10.20378/irb-46644

[19] Ashwin Srinivasan. 2004. The Aleph Manual. https://www.cs.ox.ac.uk/activities/programinduction/Aleph/.

[20] Stadt Nürnberg. 2019. Deutsches Spielearchiv Nürnberg. https://museen.nuernberg.de/spielearchiv/.

[21] Staffan Bjork and Jussi Holopainen. 2005. *Patterns in Game Design* (1st ed ed.). Charles River Media, Hingham, Mass.

[22] Technische Universität Darmstadt. 2016. DKPro Core™ Tagset Reference. https://dkpro.github.io/dkpro-core/releases/1.8.0/docs/tagset-reference.html.

[23] TH Nürnberg. 2019. Empirische Analyse Motivierender Spielelemente. https://empamos.in.th-nuernberg.de.

[24] Tom Schoeps. 2003. Die Dracheninsel.

[25] Thomas Voit, Alexander Schneider, and Mathias Kriegbaum. 2020. Towards an Empirically Based Gamification Pattern Language Using Machine Learning Techniques. In *2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)*. IEEE, Germany, 1–4. https://doi.org/10.1109/CSEET49119.2020.9206223