Machine-Learning-Modelle zur automatisierten Textklassifikation von mathematischen Aufgabenbearbeitungen

Tim Lutz

Universität Landau

Der vorliegende Artikel beschreibt eine Vorgehensweise zur automatischen Auswertung von Aufgaben mit textbasierten Antwortmustern, die sich nicht deterministisch auswerten lassen. Es wird ein Machine-Learning-Modell vorgestellt, welches sich eignet, die Aufgabenstellung ‚Was ist größer: 2n oder n+2?' automatisiert auszuwerten. Der Artikel entwickelt Fragestellungen zur didaktischen Qualitätssicherung der entwickelten Machine-Learning-Modelle und beginnt mit deren Operationalisierung. Abschließend wird der Aufbau einer Datenbank vorgestellt. Die Datenbank wird auch Modelle zur Auswertung der Arbeit mit physischen Materialien enthalten und kostenlos zur Verfügung stehen.

Einsatz von Machine-Learning für die Bedarfe von automatisierter Auswertung

Die händische Auswertung und Interpretation von Schülertexten mit mathematischen Inhalten wird traditionell in der Mathematikdidaktik genutzt, um Stufen des Verständnisses zu definieren. Dabei kommt es auch zur Beschreibung fachdidaktischer Aspekte mathematischer Objekte. Möchte man jedoch solche Aufgaben für automatisiert auswertbare Tests verwenden, stößt man schnell an die Grenzen klassischer deterministischer Auswertungsverfahren. In der Praxis werden dann alternative Aufgaben erstellt, die sich leichter automatisiert auswerten lassen. In diesem Beitrag soll am Beispiel einer Aufgabe zum Variablenverständnis nach Küchemann (1981) exemplarisch dargestellt werden, inwiefern der Einsatz von Machine-Learning es erlaubt, dennoch solche offenen Aufgaben auszuwerten, ohne die Aufgabenstellung anpassen zu müssen. Um den Diagnosecharakter der betrachteten Aufgabe zu beschreiben, wird im Folgenden zunächst das Variablenverständnis nach Küchemann erläutert. Zum besseren Verständnis muss man wissen: Küchemann spricht nicht von Variablenverständnis, sondern von der Bedeutung von Buchstaben in mathematischen Kontexten. Erst Küchemanns letzte Kategorie des Verständnisses (, die zugleich das Wesentliche seiner höchsten Stufe des Variablenverständnisses beschreibt,) trägt die Bezeichnung ‚Variable'.

Theoretischer Hintergrund

"Children's interpretations of the letters" bei Küchemann

Küchemann unterscheidet sechs Kategorien der Bedeutung von Buchstaben in mathematischen Kontexten.

- *Letter evaluated*: Von Aufgabenbearbeitungsbeginn an ersetzt die Person den Buchstaben durch eine Zahl. Diese Kategorie bezieht sich auf eine informatische Evaluation im Sinne einer Einsetzung, nicht im Sinne eines bereits fertig berechneten Ergebnisses. Beispiel: Gibt man vor: ‚2 mal x', so arbeitet die Person anstelle mit ‚x' von Beginn an mit einer Zahl, die ihr geeignet erscheint, d.h. sie denkt beispielsweise an ‚2 mal 5' und richtet alle weiteren Überlegungen auf diesen Ausdruck.

Digitales Lernen in Distanz und Präsenz. Herbsttagung des Arbeitskreises Mathematikunterricht und digitale Werkzeuge am 25. September 2021. | DOI: 10.17185/duepublico/76037

- *Not used*: Der Buchstabe wird nicht wahrgenommen, so als ob er nicht da wäre. Bestenfalls wird wahrgenommen, dass hier ein Buchstabe steht, ihm wird jedoch keine Bedeutung beigemessen.
- *Used as an object*: Der Buchstabe wird verwendet wie der Name eines Objekts oder gar als Objekt selbst, physisch gedacht.
- *Used as a specific unknown*: Der Buchstabe steht für eine eindeutig bestimmte Zahl, welche mir noch unbekannt ist. In dem Wissen, dass es sich um eine Zahl handelt, kann schon mit dem Buchstaben operiert werden. Diese Kategorie ist vergleichbar mit dem Einzelzahlaspekt (Malle, 1993).
- Used as a generalised number: Der Buchstabe steht für viele Zahlen. In dem Wissen, dass Zahlen für den Buchstaben eingesetzt werden können, kann bereits mit dem Buchstaben operiert werden. Diese Kategorie ist vergleichbar mit dem Bereich- oder Simultanaspekt (Malle, 1993).
- Used as a variable: Nach Küchemann ist dieses Level das des allgemeinsten Verständnisses in Schülerantworten. Der Buchstabe wird als Repräsentant einer eventuell näher zu bestimmenden Bandbreite unspezifischer Werte wahrgenommen. Dazu gehört auch, dass die bearbeitende Person um die Existenz systematischer Beziehungen zwischen solchen Mengen weiß und mit diesen reflektiert umgehen kann. Diese Kategorie beinhaltet somit auch den Veränderlichenaspekt (Malle, 1993).

Anwendung der Kategorien von Küchemann auf die Aufgabe „Was ist größer: 2n oder n+2?“

Küchemann (1981) untersuchte das Antwortverhalten auf die Frage „Was ist größer: 2n oder n+2?“

> *How then did children solve the item successfully and what made them hesitate and consider the effect of n instead of simply choosing one of the expressions as being the larger? The answer proposed here is that they were able, in effect, to establish a second-order relationship between 2n and n+2.* (Küchemann, 1981, S. 112)

Aus den Ausführungen von Küchemann ergibt sich, dass eine zu n relative Antwort als richtige Antwort auf die Frage nahelegt, dass der Bearbeiter ein Verständnis der Buchstaben als ‚used as a variable‘ erreicht hat. Dies ist nach Küchemann die höchste Stufe des Verständnisses. Antwortet der Aufgabenbearbeiter nicht mit einer relativen Antwort, so hat er vermutlich noch nicht dieses Level erreicht. Bei Küchemann antworteten 71% der Schüler mit der Antwort 2n und nur 6% richtig mit einer einfachen konditionalen Aussage wie „2n, wenn n>2“. Eine Antwort, die keine Fallunterscheidung dieser Art macht, erreicht also höchstens die Stufe der Variable als ‚generalised number‘. Sie rezitiert unter Umständen nämlich nur, dass 2n immer schneller wächst als n+2, und bringt diese Aussage fälschlicherweise mit der absoluten Größe des Ausdrucks in Verbindung. Dies könnte Folge eines slope-height confusion Fehlers sein (Glazer, 2011).

Das Projekt aldiff der Pädagogischen Hochschule Heidelberg entwickelte einen Test der elementaren Algebra für Studienanfänger der maßgeblich mit STACK, in manchen Fällen mit der automatisierten Vorverarbeitung vor der Zuführung zu STACK arbeitet (Lutz, 2021b).

Im Projekt aldiff wurde die Aufgabe von Küchemann in ihrer Übersetzung nach Oldenburg (Oldenburg, 2009; Oldenburg et al., 2013) strenger korrigiert. Nur Antworten, die mindestens 2 Fälle unterscheiden, also z.B. Aussagenpaare über $n>2$ und $n<2$, werden als richtig gewertet. Damit soll eine Verschärfung der Forderung nach einer konditionalen Antwort, die mehrere Bereiche berücksichtigt, erreicht werden. Trotz der strikteren Korrektur in aldiff wird die Aufgabe von ca. 1/3 der Probanden richtig beantwortet (korrekt: 169, falsch (nicht vollständig korrekt): 332, leere Antwort: 21 (im Test als falsch gewertet)). Als Antwort wurde von den meisten Probanden ein kurzer Antwortsatz unter Verwendung von algebraischen Symbolen gegeben.

Notwendigkeit eines offenen Antwortformats

In der vorgestellten Aufgabe ist für die Interpretation ‚used as a variable' das offene Antwortformat entscheidend. Die Lösungsidee „es kommt auf n an" muss der Proband selbstständig entwickeln. Es wäre daher nicht anzuraten die Aufgabe z.B. in eine teilweise Ankreuzaufgabe zu überführen, wie etwa: „Was ist größer 2n oder n+2" Auswahlmöglichkeiten: „2n", „n+2", „2n, falls"

Es soll damit nicht ausgesagt werden, dass man generell keine Ankreuzaufgaben zur Untersuchung des Variablenverständnisses nach Küchemann nutzen kann. Beispielsweise eignet sich Küchemanns Aufgabe „L+M+N =L+P+N" Ankreuzmöglichkeiten „nie", „immer", „falls…" gut um bei der Antwort „nie" auf ‚used as an object' schließen zu können. Aber, um die Aufgabe „Was ist größer 2n oder n+2?" für eine Erkennung von ‚used as a variable' nutzen zu können, dürfen also keine Kategorien vorgegeben werden. Genau darum geht es in dieser Aufgabe nämlich: selbst zu erkennen, dass eine Fallunterscheidung notwendig ist, weil die behauptete Aussage nicht pauschal, sondern nur in Abhängigkeit von n beantwortet werden kann. Es ist nicht möglich diese Aufgabe automatisch mit STACK auszuwerten, ohne bereits den Aufgabensteller zu beeinflussen. Deshalb wurde die Aufgabe im aldiff Test als ‚freie Textantwort' umgesetzt und musste manuell bewertet werden.

Theoretisches Framework aldiff. Ein automatisierter Test der Algebra

Deterministisch arbeitende Systeme zur automatischen Auswertung von Aufgaben gibt es schon lange, z.B. STACK (Sangwin, 2013). Man kann nun versuchen, bei sehr einfachen Aufgaben die Eingaben durch Ersetzungen so weit zu verändern, dass sie ohne Qualitätsverlust bei der Bewertung von STACK auswertbar werden (Lutz, 2021a).

Die Definition von Ersetzungsregeln auf Basis zuvor erhobener Daten wurde für alle Aufgaben des aldiff Tests und auf Basis von Daten aus weiteren STACK Aufgabenerhebungen untersucht. Dabei wurde festgestellt, dass sobald die Antworten auch nur etwas länger als ca. drei Worte (ausgenommen algebraischer Ausdrücke) werden, eine solche Ersetzung nicht mehr weiterhilft (Lutz, 2021b). Sofern im Schnitt mehr Worte zur Beantwortung der Fragen eingetippt werden, steigt die Varianz der Antwortabgaben deutlich an. Ersetzungsregeln gehen dann Gefahr, systematische ‚false positive' oder ‚false negative' Fälle im Vergleich zu einer manuellen Auswertung zu produzieren. Denn schon unscheinbar kleine Veränderungen in einem mathematisch argumentierenden Satz können gravierende Änderungen bei der Zuordnung zu einer fachdidaktischen Kategorie nach sich ziehen. Gleichzeitig gibt es viele oberflächlich ‚optisch' deutlich voneinander abweichende Antworten, die mathematisch ähnliche Argumentationen liefern.

Beispiele: Antwortschema, das man gut durch automatische, deterministische Verarbeitung für STACK aufbereiten kann: „Das Ergebnis ist 7.“ Diese Antwort kann zum algebraischen Ausdruck ‚7‘ vereinfacht werden. Antwortschema, das man schwierig für die automatisierte Antwortverarbeitung vorhersehen kann: „Wenn n>2 ist, dann ist 2n größer, sonst ist n+2 größer. Aber bei 2 sind sie gleich.“; „Bei n=2 sind beide gleich. Für n größer 2 ist 2n größer.“…

Möglichkeiten der Nutzung von ML-Modellen in der Didaktik der Mathematik

Wenn eine Antwort mehr als ca. drei Wörter umfasst, könnte es interessant sein Machine-Learning als Technologie für automatische Auswertung bei empirisch gut untersuchten Aufgaben zu verwenden, die als sogenannte ‚Sonden‘ (Fahse 2017) dienen und schon wiederholt eingesetzt wurden.

> A “probe” for detecting abilities is a small bundle of easily carried out measurements, observing the patterns of reactions of learners to some standardized impulse together with an established correlation of those patterns to the intended ability of the learning group. (Fahse, 2017, S. 147)

In Fortführung des Projektes aldiff stellte sich die Frage, ob nicht auch diese Aufgaben automatisch mithilfe von Machine-Learning ausgewertet werden könnten, welches neue Möglichkeiten in der Mathematikdidaktik eröffnet. Im Folgenden werden verschiedene Anwendungsbereiche von Machine-Learning in der Mathematikdidaktik geordnet, um die spezifische Bedeutung fachdidaktischer Qualitätsprüfung von Machine-Learning-Modellen zu beleuchten.

A Machine-Learning als Datenanalysestrategie (Abb. 1)

Machine-Learning-Modelle können helfen Daten explorativ zu untersuchen. Methoden wie t-SNE werden genutzt, um hochdimensionale Daten in weniger Dimensionen vereinfacht darzustellen (van der Maaten & Hinton, 2008). Sie sind dazu gedacht, ein Datenset zu analysieren und zu deuten. t-SNE (und verwandte) Verfahren sind normalerweise nicht dazu gedacht, zur Laufzeit weitere Daten hinzuzufügen und zu kategorisieren (van der Maaten, 2009). Daher werden zu dieser Kategorie auch andere Methoden gerechnet, deren Ziel keine Echtzeit-Analyse neuer Daten ist. Ein weiteres Beispiel für Machine-Learning-Cluster-Methoden in der Mathematikdidaktik ist die Eyetracking Forschung. Dort muss mit großen Datenmengen umgegangen werden, um Bearbeitungsstrategien zu entdecken (Klein et al., 2021; Strohmaier et al., 2020; Zemblys et al, 2017).

B Machine-Learning als Strategie zur Analyse eines Probanden Input in Echtzeit (Abb. 1)

Im Gegensatz zu A sind die für B vorgeschlagenen Machine-Learning-Modelle auch für Einzelanwender konzipiert, um während der Aufgabenbearbeitung in Echtzeit Entscheidungen zu treffen.

B1 Machine-Learning als Hintergrund- oder Vorab-Strategie (Abb. 1)

Machine-Learning-Modelle können im Hintergrund agieren und dadurch helfen Schülereingaben zur Laufzeit zu erkennen, um die gewonnenen Daten dann deterministischen Systemen zuzuführen. Ein schon länger immer wieder untersuchtes Themenfeld dieser Kategorie ist die handschriftliche Erkennung von mathematischen Formeln (Wu et al., 2020). Der erkannte Ausdruck wird im Anschluss dann oft an eine CAS Komponente weitergegeben, die dann auf Basis des vom ML erkannten Ausdrucks arbeitet. Ein anderes Beispiel dieser Kategorie sind Lernanwendungen, die individuelles Feedback zu Aufgabenstellungen mit physischen Materialien ermöglichen, vermittelt über AR und begleitet von vorprogrammierten Feedbackbäumen (Lutz 2021c). Arbeitet ein Schüler beispielsweise am

Themengebiet Baumdiagramm unter Verwendung von Kopf/Zahl Plättchen und Stäben als Pfade, so kann mit ML die Position der Objekte auf dem Tisch bestimmt werden. Logisch vordefinierte Feedbackbäume entscheiden dann auf Basis der ML-Positionsbestimmung, wie das System reagieren soll. ML ermöglicht diese Aufgaben ohne (auch optische) Veränderung der Materialien physisch durchzuführen und digital zu bewerten.

B2 Machine-Learning als Meta-Strategie (Abb. 1)

Machine-Learning-Modelle können Lernprozesse auf einer Metaebene gestalten. Beispielsweise können sie die nächste sinnvolle Aufgabe oder Aufgabenpaket für einen Lernenden vorschlagen (Götz & Wankerl, 2019). In diese Kategorie fallen auch Empfehlungen auf Basis des Nutzerverhaltens, z. B. Personen, die häufig mit Videos lernen und einen Lernerfolg erreichen, werden künftig auch eher Videos vorgeschlagen oder ähnliche Mechanismen als Teil von learning analytics (Picciano, 2012; Virvou et al., 2020).

B3 Machine-Learning als Strategie zur direkten automatisierten Aufgabenauswertung (Abb. 1)

Machine-Learning-Modelle können aber auch, an die vorderste Stelle der Auswertung von Aufgaben treten, wie das Modell in diesem Artikel. Die Kategorisierung einer Aufgabenbearbeitung wird von der Maschine bewertet, nachdem diese zuvor gelernt hat, Kategorien zu erkennen. Diese Kategorien sind sinnvollerweise durch Menschen erstellt, mit dem Ziel didaktische Unterscheidungen treffen zu können. Durch Supervised-Learning erlernt die Maschine die Kategorien selbst zu bestimmen. Der didaktische Einfluss von ML-Modellen wird größer in den Kategorien B1(wenig) bis B3(viel) (Abb. 1). Daher müssen die Modelle auch zunehmend kontrolliert werden. Dazu gehört beispielsweise der Bericht von Gütemaßen und eine Beurteilung der Einschränkung von Input-Daten und Output-Deutung des Modells (Lutz, 2021 d). Wichtige Aspekte bei der Erstellung von Machine-Learning-Modellen sind Verfügbarkeit und Datenschutz. Das fertige Modell soll verfügbar sein. Potentielle Anwender des Modells (andere Forscher und Lehrer) sollen leicht auf das Modell zugreifen können. Zugleich sollen keine Studien-Daten vom Anwender an den Betreiber der Software übermittelt werden. Außer wenn dies explizit vom Anwender gewünscht ist z.B. zur Modellverbesserung.

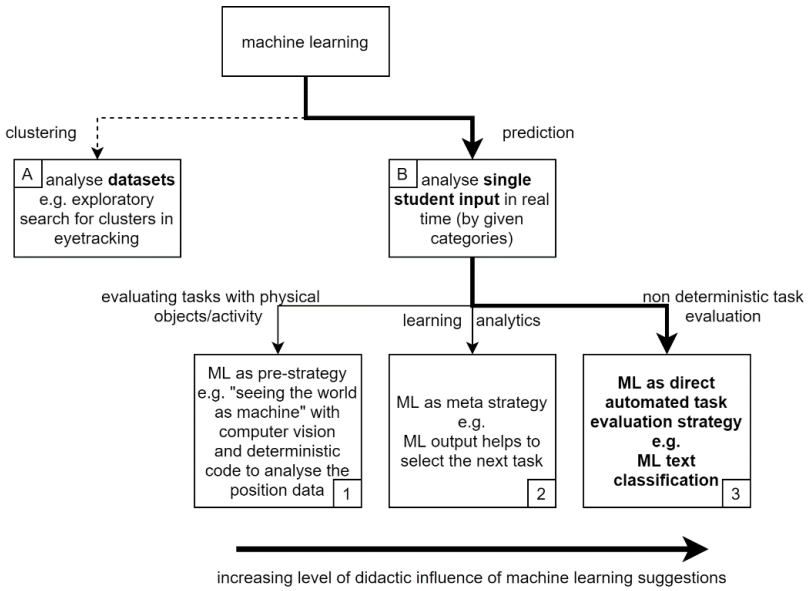

Abbildung 1. Verschiedene Anwendungen von Machine-Learning in der Didaktik der Mathematik

Methodischer Ansatz bei der Modellerstellung

Um ein Machine-Learning-Modell zu definieren, muss zunächst *Input* definiert werden: Schüler-/ Studententexte, die von Probanden im Rahmen der aldiff Studie eingegeben wurden. Um die Aufgabe automatisch auszuwerten mittels Machine-Learning muss zunächst Text in Zahlen bzw. Vektoren überführt werden, mit denen dann gearbeitet werden kann. Es gibt diverse Möglichkeiten ein solches sogenanntes Embedding durchzuführen. Es gibt sehr fortschrittliche Methoden wie Universal sentence encoder, nnlm etc. (Bengio et al., 2003). Diesen Ansätzen ist gemein, dass sie sich durch komplexeren Umgang zum Verständnis von Wörtern in Kontexten auszeichnen. In Bezug auf die Verbreitung entwickelter Modelle sind aus Gründen des Datenschutzes diese Methoden jedoch noch nicht geeignet. Der hier vorgestellte Ansatz arbeitet nur mit den Studiendaten und dessen Produkt ist auch leicht verbreitbar.

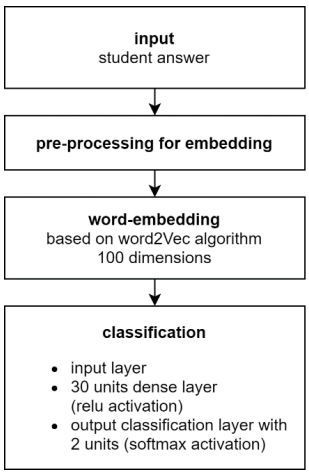

Abbildung 2. Inputverarbeitung im Modell, ausgewählt durch Maximierung der training/validation Ergebnisse.

pre-processing for embedding: Zunächst wird eine Vorverarbeitung vorgenommen. Die Vorverarbeitung hat zum Ziel die Daten der Studie in ihren didaktisch unwesentlichen Teilen, soweit möglich, zu homogenisieren. Rechtschreibung, Zeichensetzung und z.B. die Art, wie Formeln geschrieben werden, sollten, ohne zu spezielle Regeln festzulegen, vereinheitlicht werden. Solche Regeln werden im Rahmen des hier vorgeschlagenen Modells nur dann festgelegt, wenn sie über 5% der Datensätze betreffen.

word-embedding: Ein word2Vec (Mikolov et al., 2013) Verfahren wurde basierend auf den Studientexten durchgeführt. Die von den Probanden der Studie verwendeten Wörter erhalten über dieses Verfahren eine Vektordarstellung, was in diesem Zusammenhang eine Voraussetzung für die Weiterverarbeitung darstellt. Das Modell wurde mit 60% der Texte unsupervised (d.h. ohne die Vorgabe von manuell erstellten Klassen o.Ä.) trainiert (N=300).

classification: Danach erfolgt die Definition und das Training eines supervised Modells, welches als Output Kategorievorschläge (Prediction) liefert. Das Modell unterscheidet die Kategorien ‚used as a variable' und ‚not used as a variable'.

Ergebnisse

Accuracy basierte Modellwahl

Nach dem Training des Embedding-Modells werden die Probanden Antworten wortweise durch ihre Vektorenrepräsentation des Word2Vector Modells ersetzt. Das Modell schlägt für jedes Wort einen 100dimensionalen Vektor vor (Abb. 2).

Damit liegt nun einerseits eine für den Computer leichter verarbeitbare Darstellung der Probandentexte vor. Zugleich liegen aus der Phase der manuellen Bewertung der Texte Zuordnungen zu fachdidaktischen Kategorien vor. Somit sind alle Voraussetzungen für einen Supervised-Learning-Ansatz erfüllt. Supervised-Learning bedeutet, dass die Maschine Daten und deren Klassifizierung erhält und lernen soll, künftig neue Daten ebenfalls klassifizieren zu können.

Die resultierenden Vektorlisten werden zusammen mit den manuellen Textkategorien (Supervised-Learning) in ein 2-Stufen Dense-Layer-Modell mit 30 Knoten im ersten Layer und 2 Kategorisierungsknoten im 2. Layer eingespeist (Abb. 2).

Das finale Modell wurde hauptsächlich auf Basis der Accuracy optimiert. Der Validierungsdatensatz diente für vergleichende Kriterien, wie den Abbruch des Trainings zur Vermeidung von Overfitting (einer Überanpassung des Modells an den verwendeten Datensatz). Der abgetrennte Testdatensatz wurde nur einmal untersucht und hatte somit keine Auswirkung auf das Training oder die finale Modellwahl. Der Testdatensatz kann daher eine Einschätzung liefern, wie zuverlässig das Modell in der Praxis funktionieren kann.

	training 60% N=300	validation 20% N=100	test 20% N=101
Accuracy	.90	.86	.90

Tabelle 1. Training-, Validation- und Test-Split-Ergebnisse.

Tabelle 1 zeigt als Modellgüte des finalen Modells die Accuracy-Werte. Accuracy ist das Verhältnis der Anzahl korrekter Prädiktionen (true positive und true negative) zur Anzahl aller Prädiktionen. Es ist zu beachten, dass die Accuracy-Werte des Modells sich hier nur auf nicht-leere Antworten bezieht. Leere Antworten werden vom System korrekt als ‚nicht variable' klassifiziert. Eine Berücksichtigung der 21 leeren Abgaben würde die Accuracy-Werte in einem nicht vorab gefilterten Modell nochmals erhöhen. Da die Teilnehmenden keinerlei Vorinformation zur Eingabe erhalten hatten, wird aufgrund der Werte in Tabelle 1 in einer praktischen Anwendung des Modells in einer deutschsprachigen Zielgruppe am Übergang Schule-Hochschule von einer Erkennungsrate zwischen 80% und 90% ausgegangen. Verbesserungen und Erweiterungen des Modells und weitere Gütemaße werden in der Datenbank veröffentlicht. Dort findet sich auch eine Untersuchung der false positives und false negatives des Trainings- und Validierungsdatensatzes.

Komplette Tabelle und Videovortrag siehe: https://tim-lutz.de/akmdw21

Ergebnisdiskussion

Modelle, wie das hier gezeigte, sind Expertenmodelle. Diese Modelle können eine gewisse Aufgabe gut. Sie verstehen auch nur Sprache zu einer gewissen Aufgabenstellung. Die erstellten Machine-Learning-Modelle dürfen keinesfalls ohne Reflexion verallgemeinert werden. Es wird weder funktionieren einfach andere Aufgaben zu stellen und denselben Modellen vorzulegen, noch sind die so entstehenden Modelle direkt in andere Sprachen übersetzbar. Es ist wichtig zu betonen, dass auch das Antwortverhalten der Probanden ähnlich sein muss. Im Falle des vorgestellten Modells bedeutet dies z.B. digitale (eingetippte) Erhebung durch den Probanden, Sprache von Probanden aus dem Übergang Schule-Hochschule. Je näher die Situation, die von der Maschine beurteilt werden soll, der Situation der ursprünglichen Datengewinnung entspricht, desto eher ist von einer Übertragbarkeit des Modells auszugehen. Die empirische Untersuchung angrenzender Situationen, wie z.B. Einsatz in früheren Klassenstufen muss zeigen, ob es für diese Situationsveränderung bereits neue Modelle benötigt. Außerdem wird bei allen entwickelten Modellen tendenziell eine Erweiterung der Datenbasis angestrengt, um weitere Validierungs- und Optimierungsmaßnahmen treffen zu können.

Was kann aus dem Vorgehen für andere Aufgaben gelernt werden?

Bei Aufgaben mit Textantworten, bei denen es nicht möglich ist, sie symbolisch offen über Systeme wie STACK oder geschlossen (z.B. Multiple Choice) zu stellen, weil dadurch ihre didaktische Diagnose beeinflusst würde, kann es mit ein paar hundert Schülerantworten gelingen hochspezialisierte Machine-Learning-Modelle zu entwickeln. Weil die Modelle dann auch didaktische Entscheidungen treffen, muss ihnen ein Katalog von Informationen zur Seite gestellt werden. Nur so kann der potenzielle Anwender einschätzen, ob das Modell für seine Datenbasis und Ziele geeignet ist. Insbesondere muss er über folgende Informationen verfügen:

Sind die Kategorien und Aufgaben zuvor schon didaktisch untersucht worden?

Wer gibt, welche Daten, von wem, auf welche Weise als Input in das Modell ein?

Wer hat mit welchem (didaktischen) Ziel, die Trainings, Validierungs- und Test-Daten auf welche Weise von Hand kodiert?

Wer hat mit welchem Ziel, welches Machine-Learning-Modell ausgewählt und wie verlief das Training des finalen Modells?

Wie zuverlässig verhält sich das Modell? Können (didaktische) Ursachen für systematische false positive oder false negative Fälle in Betrachtung des Trainings- und Validierungsdatensatzes vermutet werden?

Ausblick

Zur Operationalisierung der zuletzt entwickelten Fragen werden verschiedentlich Anstrengungen unternommen. Die in Entwicklung befindliche Datenbank wird einen Standard festlegen und kontinuierlich erweitern, sodass zu aufgenommenen Modellen eine gewisse Mindestinformationsbasis geschaffen wird. Dabei ist wichtig zu erläutern: Es kann nicht darum gehen, Modelle auszuschließen, weil sie vermeintlich zu schlechte Werte aufweisen. Es geht darum, den möglichen Benutzern der Datenbank einheitlich Informationen an die Hand zu geben, damit sie selbst die Anwendbarkeit (Passgenauigkeit und Nutzen) auf ihre individuelle Situation bewerten können.

Zurzeit befindet sich in Zusammenarbeit mit Christian Fahse, Uni Landau (Fahse 2013, 2017) ein Modell in Entwicklung, um drei verschiedene Argumentationstypen auf Basis der Sonde „7/0=? Begründe deine Antwort“ zu unterscheiden. Dort wird in einer Weiterentwicklung des hier beschriebenen Verfahrens z.B. zusätzlich grundsätzlich eine orthographische Ersetzung häufiger Rechtschreibfehler unternommen. Diese und andere Anstrengungen sollen vorbeugen, dass das Modell nicht aus Datenmustern lernt, aus denen es nicht lernen können soll. Auch diese Anstrengungen zielen also auf eine Kontrolle des Inputs ab.

Neben den laufzeitorientierten Überlegungen wird ebenso getestet, inwiefern sich Machine-Learning-Modelle eignen, um sie als zweite Meinung einer menschlichen Einschätzung zur Seite zu stellen. Dies hat vor allem das Potential menschliche Kodierfehler künftig zu minimieren.

Link zur Datenbank: https://tim-lutz.de/mldatenbank

Literatur

Bengio, Y., Ducharme, R., Vincent, P. & Jauvin, C. (2003). A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3, 1137-1155.

Fahse, C. (2013). Argumentationstypen. In G. Greefrath, F. Käpnick, M. Stein (Hrsg.) *Beiträge zum Mathematikunterricht 2013 (S. 300-303)*. WTM-Verlag.

Fahse, C. (2017). Issues of a quasi-longitudinal study on different types of argumentation in the context of division by zero. Proceedings of the 10th Congress of European Research in Mathematics Education (Group Argumentation).

Glazer, N. (2011). Challenges with graph interpretation: A review of the literature. *Studies in Science Education*, *47*(2), 183-210. https://doi.org/10.1080/03057267.2011.605307

Götz, G. & Wankerl, S. (2020). Adaptives Online-Training für mathematische Übungsaufgaben. In F. Schacht & G. Pinkernell (Hrsg.), *Arbeitskreis Mathematikunterricht und digitale Werkzeuge: Herbsttagung, Heidelberg, 27.-28.09.2019* (S. 85-96). Franzbecker Verlag.

Klein, P., Graulich, N., Kuhn, J. & Schindler, M. (2021). *Eye-Tracking in der Mathematik- und Naturwissenschaftsdidaktik. Forschung und Praxis*. Springer Spektrum.

Küchemann, D. (1981). Chapter 8: Algebra. In K.M. Hart (Hrsg.), *Children's understanding of mathematics: 11-16* (S. 102–119). John Murray.

Lutz, T. (2021a). Automatic evaluable test of the algebra knowledge of first-year students. In *Contributions to the International STACK conference 2021*. https://doi.org/10.5281/zenodo.5036038.

Lutz, T. (2021b). *Diagnose und Förderung in der elementaren Algebra. Entwicklung eines Diagnoseinstrumentes und Vorbereitung eines Förderkonzeptes*. Springer Spektrum.

Lutz, T. (2021c). Automatisiertes Feedback in Echtzeit für die Arbeit mit physischen Materialien und ikonischen Darstellungen unter Verwendung von Machine Learning und AR. In K. Hein, C. Heil, S. Ruwisch & S. Prediger (Hrsg.) *Beiträge zum Mathematikunterricht 2021* (S. 157-160). WTM-Verlag.

Lutz, T. (2021d). Algorithmen bestimmen unsere Welt. Lass dich nicht von Algorithmen bestimmen. Über die Magie von Algorithmen und ihre Entmystifizierung. https://tim-lutz.de/algorithmen-entmystifizieren/

Malle, G. (1993). *Didaktische Probleme der elementaren Algebra.* Vieweg.

van der Maaten, L. & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9, 2579-2605.

van der Maaten, L. (2009). Learning a parametric embedding by preserving local structure. In D. van Dyk, M. Welling (Hrsg.), Proceedings of the twelth international conference on artificial intelligence and statistics, 5, (S. 384-391).

Mikolov, T., Chen, K., Corrado G. & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. https://research.google/pubs/pub41224/

Oldenburg, R. (2009). Structure of algebraic competencies. In V. Durand-Guerrier, S. Soury-Lavergne & F. Arzarello (Hrsg.), Proceedings of CERME 6 (S. 579-588). Institut National de Recherche Pedagogique.

Oldenburg, R., Hodgen, J. & Küchemann, D. (2013). Syntactic and Semantic Items in Algebra Tests. A conceptual and empirical view. In B. Ubuz, C. Haser & M.A. Mariotti (Hrsg.) *Proceedings of the Eighth Congress of the European Society for Research in Mathematics Education (CERME 8, February 6 - 10, 2013)* (S. 500-509).

Picciano, A. G. (2012). The Evolution of Big Data and Learning Analytics in American Higher Education. *Journal of Asynchronous Learning Network* 16(4), S. 9-20. https://doi.org/10.24059/olj.v16i3.267

Sangwin, C. J. (2013). *Computer aided Assessment of Mathematics.* Oxford Univ. Press.

Strohmaier, A. R., MacKay, K. J., Obersteiner, A. & Reiss, K. M. (2020). Eye-tracking methodology in mathematics education research: A Systematic literature review. *Educational Studies in Mathematics*, 104, 147-200. https://doi.org/10.1007/s10649-020-09948-1

Virvou, M., Alepis, E., Tsihrintzis, G. A. & Jain, L. C. (2020). Chapter 1: Machine Learning Paradigms. Advances in Learning Analytics. In M. Virvou, E. Alepis, G. Tsihrintzis & L. Jain (Hrsg.) *Machine learning paradigms. Intelligent systems reference library* (S. 1-5). Springer.

Wu, J. W., Yin, F., Zhang, Y. M., Zhang, X. Y. & Liu, C. L. (2020). Handwritten mathematical expression recognition via paired adversarial learning. *International Journal of Computer Vision*, *128*, 2386–2401.

Zemblys, R., Komogortsev, O. & Holmqvist, K. (2017). Using machine learning to detect events in eye-tracking data. *Behavior Research Methods* 50, 160-181. https://doi.org/10.3758/s13428-017-0860-3

DOI: 10.17185/duepublico/76037
URN: urn:nbn:de:hbz:465-20220615-162757-8