Projektbericht zum Modul Data Mining Wintersemester 20221/2022

Reproduktion des Papers *Context-Sensitive Visualization of Deep Learning Natural Language Processing Models*[1]

Max Henze

2. März 2022

1 Einleitung

Neuronale Netzwerke mit Transformern sind ein beliebtes Hilfsmittel in NLP und Modelle wie BERT [2] oder GPT 2 [3] gehören schon lange zum state-of-the-art, was durch ihre sehr guten Ergebnisse [4, 5, 6] in diversen Bereichen untermahlt wird. Doch die Tatsache, dass Einblicke in diese sehr komplexen Modelle und ein somit besseres Verständnis für von ihnen getroffene Entscheidungen nicht sehr einfach ist, war Anlass für Dunn et al. um sich in ihrer Arbeit [1] näher damit zu beschäftigen. Konkret: Dunn et al. entwickelten in ihrem Paper einen Ansatz zur abstufenden Visualisierung der Wichtigkeit von Wörtern, wie sie von einem Neuronalen Netz bei der Klassifikation bewertet werden.
Ziel dieser Arbeit ist es das ganze Paper von Dunn et al. zu replizieren. Es existiert kein Code und keine sonstigen Beigaben zum Paper, sodass jegliche Replikationen aus den Wortlauten und Beschreibungen der Autoren erzeugt werden müssen.
Durch diese Arbeit sollen die Experimente von Dunn et al. erneut verifiziert werden. Der erzeugte Code soll anderen Lesern als Hilfestellung beim Verständnis des Papers dienen und somit eine einfache Möglichkeit bei der Erstellung weiterer Experimente geben.

2 Umfang der Replikation/Reproduktion

Der Umfang dieser Replikation umfasst die gesamte Arbeit von Dunn et al. Diese lässt sich zu der Aussage zusammenfassen, dass bei der Klassifikation von Dokumenten bestimmte Wörter einen größeren Einfluss haben als andere. Dies geht mit unserer Intuition einher, da zum Beispiel *und* weniger aussagt als *bombastisch* und somit letzteres den Text in einen vermutlich eher positiven Kontext rückt. Nun gibt es Techniken wie leave-one-out, welche über alle Token in einem Dokument durch iterieren, das gerade betrachtete Token entfernen und dann den Text neu klassifizieren. Die Wahrscheinlichkeit, der Neuklassifikation kann also mit der Wahrscheinlichkeit des Originaltextes verglichen werden wodurch eine gewisse Wichtigkeit des Wortes innerhalb der Klassifikation deutlich wird. Basierend auf diesen Techniken propagieren Dunn et al. nun folgende Behauptung:

- „Unser Ansatz schaut auf die Kombination von Wörtern und Sätzen um deren Einfluss auf die Ausgabe des Modells zu erkennen, was zu einer Visualisierung führt, welche kontextsensitiver zum Originaltext ist.“[1]

3 Methoden

3.1 Modellbeschreibung

Bei dem in der Arbeit von Dunn et al. benutztem Modell handelt es sich um ein BERT-base Modell ohne genauere Angaben.

3.2 Datenbeschreibung

Der im Originalartikel und dieser Replikation verwendete Datensatz ist das large Movie Review Dataset [7] der Universität Stanford. Dieser umfasst 50.000 Dokumente mit einer Teilung in 25.000 Trainingsdokumente und 25.000 Testdokumente und kann unter `https://ai.stanford.edu/~amaas/data/sentiment/` heruntergeladen werden. Dies wird im Jupyter Notebook automatisch gemacht, falls der Datensatz noch nicht vorhanden ist. Der Datensatz hat eine vorgegebene Ordnerstruktur, so befinden sich die Trainingsdokumente und Testdokumente in eigenen Ordnern, wobei positive und negative Dokumente nochmals in eigene Ordner unterteilt sind. Eine Datei ist in der Struktur `x_y.txt` angegeben wobei x die Dokumentenid und y die Sternewertung von 0 bis 10 ist. Die Dokumente unterscheiden sich stark in der Länge, so gibt es Dokumente mit knapp über 50 Zeichen aber auch solche mit über 13.000 Zeichen. Die Dokumente an sich sind nicht aufbereitet, enthalten englische Altagssprache und Sonverzeichen.

```
Fair drama/love story movie that focuses on the lives of
blue collar people finding new life thru new love.The acting
here is good but the film fails in cinematography,screenplay,
directing and editing.The story/script is only average at best.
This film will be enjoyed by Fonda and De Niro fans and by
people who love middle age love stories where in the coartship
is on a more wiser and cautious level.
It would also be interesting for people who are
interested on the subject matter regarding illiteracy.......
```

Abbildung 1: Beispieltext eines positiven Trainingsdokuments

Zusätzlich zu der Unterteilung in Trainings- und Testdaten, wurden die Trainingsdaten wie im Originalartikel angegeben noch in 20 Prozent Validierungsdaten geteilt.

3.3 Hyperparameter

Die im Modell einstellbaren Hyperparameter sind: Chargengröße (batch size), Lernrate, Epochenanzahl und Verlustrate des Dropout Layers. Auch hier waren keine Angaben seitens des Originalartikels vorhanden. Daher orientierte sich die Festlegung dieser Parameter an externen Berichten. Nur die Chargengröße orientierte sich nicht an diesen. Um ein Training auf einer Grafikkarte zu ermöglichen musste diese verringert werden. *Tensorflow*[1], das zur Implementierung des Modells verwendete Python Package, schlägt Werte für eine Textklassifikation mit BERT vor. Diese Vorschläge orientieren sich am selben Datensatz wie er hier Anwendung findet, sodass diese Werte im Projekt übernommen wurden und für eine sehr gute Performance des Models sorgen. Mehr dazu im nächsten Kapitel.
Über das *officials.nlp* Package wurde der AdamW [8] Optimierer implementiert, welcher beim Training die Hyperparameter des Modells anpasst.

3.4 Implementierung

Die Implementierung erfolgte über Python[2] mit Hilfe eines Jupyter Notebooks[3].
Sie gliederte sich in drei große Abschnitte: Modellerzeugung, Ergebnisverarbeitung, Visualisierung.

3.4.1 Modellerzeugung

Die Modellerzeugung wurde mit Hilfe des *tensorflow* und des *official.nlp* Package bearbeitet. Zuerst wurden die Daten in die drei Datensätze: Test, Training und Validierung unterteilt. Dies wurde mit Hilfe der `text_datase_from_directory` Funktion aus *tensorflow* erzeugt. Die

[1] `https://www.tensorflow.org/`
[2] `https://www.python.org/`
[3] `https://jupyter.org/`

Ordnerstruktur der Daten ist standartisiert und somit im richtigen Format um von der Funktion verarbeitet zu werden.
Der Encoder, also das eigentliche BERT Modell und der Preprocesser können über eine zentrale Anlaufstelle[4] heruntergeladen werden.

```
# function for building the classifiert model
# text input -> preprocessing -> encode -> droput -> dense
if not os.listdir('models'):
    def build_classifier_model():
        text_input = tf.keras.layers.Input(shape=(), dtype=tf.string, name='text')
        preprocessing_layer = hub.KerasLayer(tfhub_handle_preprocess, name='preprocessing')
        encoder_inputs = preprocessing_layer(text_input)
        encoder = hub.KerasLayer(tfhub_handle_encoder, trainable=True, name='BERT_encoder')
        outputs = encoder(encoder_inputs)
        net = outputs['pooled_output']
        net = tf.keras.layers.Dropout(0.1)(net)
        net = tf.keras.layers.Dense(1, activation=None, name='classifier')(net)
        return tf.keras.Model(text_input, net)
```

Abbildung 2: Funktion um das Klassifikationsmodell zu erzeugen. Es besteht aus einem Input-, Preprocessing-, Encode-, Dropout- und Denselayer.

Wie in Abbildung 2 zu sehen ist, wurden die einzelnen Layer mit Hilfe von `hub.KerasLayer` (*tensorflow* Package) hintereinander geschalten.
Als Lossfunktion wurde Binary Crossentropy und als Metrik die Binary Accuracy verwendet. Beide können ebenfalls über *tensorflow* geladen und verwendet werden.
Nun kann das Modell kompiliert, trainiert und dann getestet werden. Dabei ergeben sich folgende Ergebnisse:

3.5 Aufbau der Experimente

Erklären sie, wie sie ihre Experimente durchgeführt haben. Was für Ressourcen haben sie verwendet, z.B. GPU/CPU-Ressourcen. Verlinken sie ihren Code und Notebooks.

3.6 Ressourcen für Berechnungen

Beschreiben sie die Anforderungen für die Berechnungen für jedes ihrer Experimente, z.B. die Anzahl der CPU/GPU-Stunden oder die Voraussetzungen für den Hauptspeicher und GPU-Speicher. Geben sie für Zeit und Speicher eigene Abschätzungen an, bevor die Experimente gelaufen sind und vergleichen sie dies mit den tatsächlich verbrauchten Ressourcen. Sie müssen vor den Experimenten einplanen, dass diese Informationen auch durch ihren Code gemessen und gespeichert werden.

[4] `https://tfhub.dev`

4 Ergebnisse

Starten sie mit einem Überblick über die Ergebnisse. Bestätigen ihre Ergebnisse die aufgeführten Behauptungen? Dieser Abschnitt sollte hauptsächlich Fakten nennen und so präzise wie möglich geschrieben werden. Die Bewertung und Diskussion kann im späteren Kapitel "Diskussion" folgen.
Beschreiben sie dann detailliert jedes einzelne Ergebnis, das sie haben. Zeigen sie wie es mit einer oder mehreren Behauptungen in Beziehung steht. Erklären sie konkret was der Kern ihres Ergebnis ist. Gruppieren sie die Ergebnisse in logische Abschnitte. Beschreiben sie klar, wo sie über den Originalartikel hinausgegangen sind, wo sie zusätzliche Experimente durchgeführt haben und wie diese mit den ursprünglichen Behauptungen in Beziehung stehen.
Tipp 1: Drücken sie sich genau aus und verwenden sie eine klare und einfache Sprache, z.B.

> "we reproduced the accuracy to within 1% of reported value, that upholds the paper's conclusion that it performs much better than baselines." oder
>
> "We konnten die Klassifikationsrate bis auf 1% des angegebenen Werts reproduzieren. Dies unterstützt die Schlussfolgerung der Artikels, dass der Ansatz leistungsfähiger als die Baselines ist."

Oft kann man nicht die exakt gleiche numerische Zahl als Ergebnis bekommen. Deshalb müssen sie das Ergebnis bewerten, um zu entscheiden, ob ihr Ergebnis die Behauptung der Originalartikels unterstützt.
Tipp 2: Nutzen sie Tabellen und Abbildungen, um ihre Ergebnisse darzustellen.

4.1 Ergebnis 1

4.2 Ergebnis 2

4.3 Zusätzliche Ergebnisse, die nicht im Originalartikel enthalten waren

Beschreiben sie alle zusätzlichen Experimente, die über den Originalartikel hinausgehen. Dies können Experimente zu weiteren Datenmengen sein oder sie probieren andere Methoden bzw. weitere Vereinfachungen des Modells aus oder passen die Hyperparameter an. Beschreiben sie für jedes zusätzliche Experiment, was sie genau durchgeführt haben, was die Ergebnisse sind und diskutieren sie was diese Ergebnisse zeigen.

5 Diskussion

Beschreiben sie die weiterführenden Implikationen der experimentellen Ergebnisse. War der Originalartikel replizierbar bzw. reproduzierbar. Falls nicht, welche Faktoren haben dazu geführt, dass die Experimente nicht reproduziert werden konnten.

Bewerten sie, ob sie die Evidenz, die sie durch das Durchführen der Experimente erhalten haben, auch überzeugt, dass die Behauptungen des Originalartikels dadurch gestützt werden. Diskutieren sie die Stärken und Schwächen ihres Ansatzes, vielleicht haben sie aus Zeitgründen nicht alle Experimente durchführen können, oder vielleicht haben zusätzliche Experimente durchgeführt, die den Originalartikel weiter stärken.

5.1 Was war einfach?

Beschreiben sie welche Teile der Replikation/Reproduktion sich leicht umsetzen ließen. Lief der Code der Autoren problemlos? War es aufgrund der Beschreibung im Originalartikel nicht aufwändig die Methoden zu reimplementieren? Dieser Abschnitt soll den Lesenden zeigen, welche Teile des Originalartikels sich leicht für eigene Ansätze verwenden lässt.
Tipp: Machen sie keine pauschalen Verallgemeinerungen. Was für sie leicht ist, muss für andere nicht leicht sein. Geben sie genügend Kontext und erklären sie warum manche Sachen leicht waren, z.B. der Code hatte eine umfangreiche Dokumentation der Schnittstellen und viele Beispiele aus der Dokumentation passten zu den Experimenten im Artikel.

5.2 Was war schwer?

Beschreiben sie welche Teile ihrer Replikation/Reproduktion aufwändig oder schwierig waren oder viel mehr Zeit in Anspruch genommen haben, als sie erwarteten. Vielleicht waren Daten nicht verfügbar, so dass sie einige Experimente nicht verifizieren konnten, oder der Code der Autoren funktionierte nicht und musste erst debugged werden. Vielleicht dauerten auch einige Experimente zu lange und sie konnten sie deshalb nicht verifizieren. Dieser Abschnitt soll den Lesenden zeigen, welche Teile des Originalartikels schwer wiederverwendbar sind, bzw. signifikante Zusatzarbeiten und Ressourcen erfordern.
Tipp: Setzen sie sorgfältig ihre Diskussion in den richtigen Kontext, z.B. sagen sie nicht “ die Mathematik war schwer verständlich” sondern sagen sie “ die Mathematik erfordert fortgeschrittene Kenntnisse in Analysis für das Verständnis”.

5.3 Empfehlungen für die Replizierbarkeit / Reproduzierbarkeit

Geben sie Empfehlungen, wie die Autoren des Originalartikels oder andere Forschende in diesem Feld die Replizierbarkeit / Reproduzierbarkeit verbessern können.

6 Kommunikation mit den Autoren

Dokumentieren sie das Ausmaß (oder das Fehlen) der Kommunikation mit Autoren. Stellen sie sicher, dass der Bericht eine faire Beurteilung der Forschungsarbeiten ist. Versuchen sie deshalb mit den Autoren Kontakt aufzunehmen. Sie können ihnen konkrete Fragen stellen oder falls sie keine Fragen haben, den Bericht zusenden und um Feedback bitten.