OST
Ostschweizer
Fachhochschule

9 Word Embeddings

Systemtechnik BSc
FS 2025

Aufgaben

Lernziele

Nach dem Bearbeiten dieser Übungsserie ...

- wissen Sie, was *Word-Vektoren* sind und können Sie vortrainierte Word-Vektoren als *Embedding-Layer* für die Verarbeitung von natürlicher Sprache (NLP) verwenden).
  Aufgaben: 1
- können Sie Daten aus mehreren Quellen zu einem einzigen Datensatz integrieren und für Training und Validierung verwenden.
  Aufgaben: 1
- können Sie Texte aus einem Korpus vorverarbeiten (cleaning, stemming, lemmatization) und mit einem gegebenen Dictionary *vektorisieren* und als Input für Modelle verwenden.
  Aufgaben: 1

Vortrainierte Word-Embeddings

In dieser Übungsserie lernen wir, wie man ein Textklassifizierungsmodell trainiert, das vortrainierte Worteinbettungen verwendet. Wir arbeiten mit dem Newsgroup20-Datensatz, einem Satz von 20.000 Nachrichten auf Messageboards die zu 20 verschiedenen Themenkategorien gehören. Für die vortrainierten Worteinbettungen werden wir die GloVe Embeddings verwenden.

GloVe, abgeleitet von *Global Vectors*, ist ein Modell für verteilte Wort-Repräsentationen (Word-Vectors). Bei dem Modell handelt es sich um einen unüberwachten Lernalgorithmus zur Ermittlung von Vektordarstellungen für Wörter. Dies wird erreicht, indem Wörter in einen sinnvollen Vektorraum abgebildet werden, in dem der Abstand zwischen den Wörtern mit der semantischen Ähnlichkeit zusammenhängt.
Das GloVe-Modell wird anhand der Nicht-Null-Einträge einer globalen Wort-Wort-Koinzidenzmatrix trainiert, in der die Häufigkeit des gemeinsamen Auftretens von Wörtern in einem bestimmten Korpus aufgelistet ist. Die resultierenden Word-Embeddings (Repräsentationen) zeigen interessante lineare Unterstrukturen des Wortvektorraums. Es wurde als Open-Source-Projekt in Stanford entwickelt und wurde 2014 eingeführt.

Aufgabe 1. Verwenden von Word-Embeddings für die Text-Klassifizierung

In dieser Aufgabe soll ein Textklassifizierungsmodell entwickelt und trainiert werden, das vortrainierte Worteinbettungen verwendet. Wir arbeiten mit dem Newsgroup20-Datensatz. Die Sammlung von 20 Newsgroups ist zu einem beliebten Datensatz für Experimente mit Textanwendungen von maschinellen Lerntechniken geworden, wie z. B. Textklassifizierung und Textclustering. Die Daten sind in 20 verschiedene Newsgroups (Klassen) unterteilt, die jeweils einem bestimmten Thema entsprechen. Einige der Newsgroups sind sehr eng miteinander verwandt (z. B. comp.sys.ibm.pc.hardware / comp.sys.mac.hardware), während andere in keinem Zusammenhang stehen (z. B. misc.forsale / soc.religion.christian). Hier ist eine Liste der 20 Newsgroups, die (mehr oder weniger) nach Themen geordnet sind. Ziel unseres Klassifizierers ist es, anhand des Textes, die Newsgroup (Klasse) herauszufinden.

(a) Öffnen sie das Jupyter-Notebook:
`ANN09_Pretrained_Word_Embeddings_TEMPLATE.ipynb`, welches Sie auf Moodle finden und führen Sie die ersten Zellen aus, welche die Imports ausführen. Der Datensatz wird automatisch von der Originalquelle heruntergeladen. Das Ergebnis ist ein strukturierter Datenordner mit den Klassen.

(b) **Daten einlesen und vorbereiten:** Lade alle Textdokumente ein. Weise jedem Dokument das passende Klassenlabel zu. Das Ergebnis sind zwei Listen (samples, labels) sowie die Klassennamen.

(c) **Train- und Testsplit:** Teile die Daten in Trainings- und Validierungsdaten auf.

(d) **Tokenisierung und Vokabular-Erstellung:** Tokenisiere die Texte in einfache Kleinbuchstaben-Wörter. Erstelle zudem ein Vokabular aus den häufigsten 20.000 Tokens, reserviere Index 0 für <PAD> und 1 für <OOV>. Das Ergebnis ist ein Wörterbuch vocab, das Wörter in Integer-IDs übersetzt.

(e) **PyTorch Dataset und DataLoader erstellen:** Implementiere ein Dataset-Objekt zur effizienten Bereitstellung der Trainings- und Validierungsdaten.

(f) **GloVe-Embeddings integrieren:** Lade die vortrainierten GloVe-Vektoren (100d) herunter und lese sie ein. Initialisiere eine Embedding-Matrix, in die Vektoren für Wörter aus dem Vokabular eingefügt werden. Wörter ohne GloVe-Vektor erhalten Nullen. Ergebnis: Eine GloVe-basierte Matrix zur Initialisierung der Embedding-Schicht im Modell.

(g) **CNN-Modell mit PyTorch Lightning implementieren:**

- Erstelle ein TextCNN-Modell mit mehreren 1D-Convolutional Layers, Max-Pooling, Dropout und voll verbundenen Schichten.
- Nutze die GloVe-Embeddings als festen Input.
- Implementiere Trainings- und Validierungsschritte sowie einen RMSprop-Optimierer. Inkludiere eine predict_text()-Methode zur Vorhersage beliebiger Texte.

(h) **Modell trainieren und testen:**

- Trainiere das Modell mit dem Trainer von PyTorch Lightning (15 Epochen).
- Teste die Inferenz auf neuen Beispieltexten.
- Berechne und gib die Accuracy auf den Validierungsdaten aus.

Lösungen

Lösung 1. Sie finden die Lösung auf Moodle unter `ANN09_Pretrained_Word_Embeddings_SOLUTION.ipynb`.