Some weights of the model checkpoint at bert-base-multilingual-cased were not used when initializing BertForSequenceClassification: ['cls.predictions.transform.LayerNorm.weight', 'cls.predictions.transform.dense.weight', 'cls.predictions.bias', 'cls.predictions.decoder.weight', 'cls.predictions.transform.dense.bias', 'cls.predictions.transform.LayerNorm.bias', 'cls.seq\_relationship.weight', 'cls.seq\_relationship.bias']

- This IS expected if you are initializing BertForSequenceClassification from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a BertForSequenceClassification model from a BertForPreTraining model).

- This IS NOT expected if you are initializing BertForSequenceClassification from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassification model from a BertForSequenceClassification model).

Some weights of BertForSequenceClassification were not initialized from the model checkpoint at bert-base-multilingual-cased and are newly initialized: ['classifier.weight', 'classifier.bias']

You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference

TRAIN TEST

HYPERPARAMETER

Es gibt auch fortgeschrittenere Techniken wie das adaptive Einstellen der Lernrate während des Trainings (z.B. Adagrad, Adam), die in der Lage sind, die Lernrate automatisch an die Bedürfnisse des Modells anzupassen.

Ja, das kann durchaus vorkommen. Es gibt verschiedene Gründe dafür, warum man bei einem Modell mit den gleichen Hyperparametern unterschiedliche Evaluationsergebnisse erhalten kann:

Zufälligkeit: Die initialen Gewichte des Modells werden normalerweise zufällig gesetzt, was dazu führen kann, dass das Modell unterschiedliche Ergebnisse liefert, wenn es mit den gleichen Daten trainiert wird.

Datenvariation: Die Daten, auf denen das Modell trainiert wird, können von Durchlauf zu Durchlauf unterschiedlich sein, insbesondere wenn die Daten zufällig sortiert oder partitioniert werden. Das kann dazu führen, dass das Modell unterschiedlich trainiert wird und dadurch unterschiedliche Evaluationsergebnisse liefert.

Evaluationsmetriken: Die Wahl der Evaluationsmetriken kann ebenfalls dazu führen, dass sich die Ergebnisse unterscheiden. Manche Metriken sind empfindlicher auf bestimmte Arten von Fehlern als andere, so dass unterschiedliche Metriken unterschiedliche Ergebnisse liefern können.

Unvorhergesehene Ereignisse: Es gibt auch andere Faktoren, die dazu führen können, dass ein Modell bei jedem Training unterschiedliche Ergebnisse liefert, wie zum Beispiel Änderungen an der Hardware oder der Softwareumgebung, oder Änderungen an den Daten.

Aus diesen Gründen ist es wichtig, ein Modell mehrmals mit verschiedenen Hyperparametern und zufälligen Startgewichten zu trainieren, um sicherzustellen, dass man ein robustes Modell hat, das konsistente Ergebnisse liefert.

Die Aufteilung der Trainings- und Testdaten hat einen wichtigen Einfluss auf die Modellgüte. Eine schlechte Aufteilung kann zu einem über- oder unterangepassten Modell führen. Hier sind einige Punkte zu berücksichtigen:

Trainingsgröße: Wenn die Trainingsdaten zu klein sind, kann das Modell nicht genügend Muster in den Daten lernen und wird wahrscheinlich unteranpassen. Wenn die Trainingsdaten zu groß sind, kann das Modell zu komplex werden und zu Overfitting neigen.

Testgröße: Eine zu kleine Testgröße kann zu einer ungenauen Bewertung des Modells führen, da das Modell möglicherweise nicht auf alle Variationen der Testdaten generalisieren kann. Eine zu große Testgröße kann zu einer erhöhten Rechenzeit führen, ohne die Modellgenauigkeit signifikant zu verbessern.

Zufälligkeit: Die zufällige Aufteilung der Trainings- und Testdaten kann zu unterschiedlichen Ergebnissen führen. Um dies zu minimieren, kann eine Kreuzvalidierung verwendet werden, bei der das Modell mehrere Male mit verschiedenen Trainings- und Testdaten trainiert und bewertet wird.

Repräsentativität: Die Trainings- und Testdaten sollten repräsentativ für das Problem sein, das das Modell lösen soll. Wenn die Trainingsdaten nicht die relevanten Merkmale des Problems enthalten, wird das Modell nicht in der Lage sein, eine gute Vorhersage zu treffen.

Zusammenfassend lässt sich sagen, dass eine gute Aufteilung der Trainings- und Testdaten dazu beitragen kann, dass das Modell gut generalisiert und in der Lage ist, genaue Vorhersagen zu treffen.

In machine learning und Data Science wird oft der Begriff "random\_state" verwendet, um die Reproduzierbarkeit von Ergebnissen sicherzustellen. Der Parameter "random\_state" stellt sicher, dass bei der Zufallsauswahl bestimmter Datenpunkte oder Parameter immer dieselben Ergebnisse erzielt werden, wenn er auf denselben Wert gesetzt wird.

Wenn Sie beispielsweise ein Modell trainieren und "random\_state=42" setzen, erhalten Sie bei jeder Ausführung des Codes dieselben Trainings- und Testdaten sowie dieselben Initialisierungsparameter. Dies ermöglicht eine bessere Vergleichbarkeit von Ergebnissen und erleichtert das Debugging von Fehlern.

In der Regel ist es empfehlenswert, "random\_state" auf einen bestimmten Wert zu setzen, um reproduzierbare Ergebnisse zu erzielen. Wenn "random\_state" nicht festgelegt ist, wird bei jeder Ausführung des Codes ein anderer Zufallswert verwendet, was zu unterschiedlichen Ergebnissen führen kann.

## Evaluierung Sentiment-Analyse Spiegel Online

Ebene wie oft jedes Sentiment pro Medium 🡪 ist lage grundsätzlich negativer positiver etc

Ebene Wie sentiment nach Entity

Weiterführende statistische und regressionelle analyse vllt auch in schluss

begriffe ner doppeldeutig - NER

# Fazit

Eine TrainDatenbasis in der nicht aus der Politik sondern über die Politik gesprochen wird

Vllt Spezifizierung schlecht, lieber XXLkorpus der dann ohnehin politische Sprache enthält so sehr feinjustiert -- Overfitting

Grundsätzlich technisch möglich, Modellgüte und damit Ergebnisqualität und Validität sowie Genauigkeit kann noch erheblich verbessert werden. Während die ersten beiden Kriterien vor allem durch erheblich umfangreicheres Finetuning und Training verbessert werden kann, kann die Genauigkeit auch durch zusätzliche Untersuchungsmodelle verbessert werden. Nur durch die Nennung und die Kategorisierung des Sentiments dieser Nennung kann zwar ein Richtungsweiser geschaffen werden, wie politische ausgerichtet ein Medienprodukt ist, um voll umfänglich und abschließend bestimmen zu können, welcher politischen Strömung das Medium angehört und wie stark, sind weitere, auch qualitative Methoden vonnöten. Politische Meinung bleibt für Deep-Learning-Modelle schwer einzustufen, die aktuelle Entwicklung des Maschinellen Lernens im Verbund mit Computergesteuerter Sprachverarbeitung durch Neuronale Netze zeigt aber, dass hier IMMERMEHR DESWEGEN bald besser – aber Expertenwissen

Politishce Inhalte

Ebenso breit gesät wie die inhaltlichen Möglichkeiten, um diese Studie zu verbessen, sind die technischen Aspekte und Limitationen.

Test mit als tendentiös eingestuften medienprodukten (Tichys Einblick) --> IDee MA

Online portale Wenig Meinung – printprodukt

Cashwalls beim webscraping