Anwendungen der künstlichen Intelligenz
Question-Answering Projekt

Mustafa Kilci

Levent Esen

10. Februar 2024

Inhaltsverzeichnis

Motivation

In unserem Projekt steht das Thema Benchmarking im Vordergrund. Ziel ist es, ein Frage-Antwort-System zu entwickeln, das auf einer umfangreichen Wikibase basiert, die rund 300.000 Wikipedia-Seiten umfasst. Das Hauptmotiv hinter diesem Vorhaben ist ein Wettbewerb zwischen zwei Teams, um die höchste Trefferquote bei der Beantwortung von Fragen zu erzielen. Benchmarking in diesem Kontext bedeutet, die Leistungsfähigkeit verschiedener Methoden des Frage-Antwort-Systems zu vergleichen und zu bewerten. Die Herausforderung besteht darin, eine effektive Methode zu entwickeln, die eine hohe Trefferquote bei der Beantwortung von Fragen aus der Wikibase gewährleistet, wobei Faktoren wie Genauigkeit, Geschwindigkeit und Effizienz berücksichtigt werden müssen. Die Motivation für unser Team bestand darin, nicht nur im Wettbewerb erfolgreich zu sein, sondern auch zur Entwicklung und Weiterentwicklung von Technologien im Bereich des Natural Language Processing (NLP) beizutragen.

Lösungsansatz

Unser Lösungsansatz basiert auf mehreren Schritten, um eine präzise Beantwortung von Fragen aus der Wikibase zu ermöglichen. Zunächst verwenden wir **Elasticsearch** für die Volltextsuche in der Wikibase, um relevante Dokumente zu den gestellten Fragen zu finden. Anschließend nutzen wir **FastText**, um die semantische Ähnlichkeit zwischen Frage und potenzieller Antwort zu bestimmen. Dies ermöglicht es uns, die relevantesten Sätze innerhalb der gefundenen Dokumente zu identifizieren.
Wir trainieren und verwenden scikit-learn‘s LogisticRegression **Klassifikator**, um Fragen nach dem Schema von Li und Roth zu klassifizieren. Eine Kombination aus dem Klassifikator und **Spacy(NER)** für Entitätserkennung ermöglicht uns dann spezifische Informationen wie Personen, Orte, Zeitangaben und andere Entitäten aus den Sätzen zu extrahieren.

Implementierung

Für die Implementierung unseres Lösungsansatzes haben wir verschiedene Schritte durchgeführt:

1. **Vorbereitung der Daten und Modelle**: Wir haben zunächst die erforderlichen Modelle und Daten geladen, darunter das vortrainierte FastText-Modell für die semantische Ähnlichkeit und das trainierte Klassifikationsmodell für die FrageTypenbestimmung.

2. **Normalisierung der Fragen**: Die gestellten Fragen wurden mithilfe von Spacy normalisiert, indem Satzzeichen und Stoppwörter entfernt sowie die Tokens in Kleinbuchstaben umgewandelt wurden.

3. **Elasticsearch-Volltextsuche**: Wir haben Elasticsearch verwendet, um relevante Dokumente in der Wikibase zu finden, die potenzielle Antworten auf die gestellten Fragen enthalten. Es werden insgesamt 3 Dokumente zurückgegeben.

4. **Berechnung der semantischen Ähnlichkeit**: Mithilfe von FastText haben wir die semantische Ähnlichkeit zwischen den gestellten Fragen und den Passagen in den gefundenen Dokumenten berechnet. Dies erfolgte durch die Generierung von Vektoren für die Frage und den einzelnen Sätzen aus der Passage und die anschließende Berechnung der Kosinusähnlichkeit zwischen diesen Vektoren.

5. **Extraktion von Entitäten**: Mithilfe von Frage-Typen, die mit dem Klassifikator vorhergesagt werden, und Spacy NER haben wir je nach Frage-Typ eine bestimme passende Entität extrahiert. Im Ordner "classifier" befinden sich die entsprechenden Dateien und auch das Skript, um den einen classifier zu erstellen und zu testen.

6. **Erstellung der Antworten**: Basierend auf den Ergebnissen der semantischen Ähnlichkeit und der Entitätsextraktion haben wir die relevantesten Passagen aus den gefundenen Dokumenten identifiziert und die entsprechenden Antwortwörter extrahiert.

7. **Ausgabe der Antworten**: Schließlich haben wir die extrahierten Antwortwörter in eine CSV-Datei geschrieben, um die Ergebnisse zu speichern.

Insgesamt kombiniert unsere Implementierung Methoden aus verschiedenen Bibliotheken wie Spacy, Elasticsearch und FastText, um eine effektive und präzise Beantwortung von Fragen aus der Wikibase zu erreichen.

Ergebnis

Unser eigenes Evaluationsskript zeigt eine ungefähre Erfolgsrate von 37.14% für den Evaluationsdatensatz. Das Skript beachtet jedoch keine Reihenfolge der Antworten. Es überprüft nur, ob die Antwort gefunden wurde. Diese ist auch im Ordner "eval" zu finden. Außerdem zeigt unser Klassifikator eine Erfolgsrate von 67%.

Limitierung des Systems

Fast alle verwendeten Komponenten zeigen mehrere Limitierungen, die in diesem Abschnitt mit potenziellen Lösungsansätzen besprochen werden. Eine allgemeingültige Herausforderung ist jedoch synonyme Wörter. Damit ist gemeint, dass die Frage ein Verb enthalten könnte, welches nicht in der Passage zu finden ist oder umgekehrt genauso. Eine Lösung hierfür ist es, die Synonyme in Betracht ziehen und sie auf eine Standardform zu reduzieren.

Elasticsearch findet öfters auf Anhieb das richtige Dokument nicht. Wir glauben, dass eine Parameteranpassung dies verbessern könnte. Ein anderer "type" wie zum Beispiel "phrase" oder weitere Parameter wie "minimum_should_match", "analyzer" und "boost".

FastText extrahiert manchmal falsche Sätze aus längeren Passagen. Die einzige mögliche Lösung, die uns einfällt, ist eine bessere Normalisierung der Passagen.

Klassifikator. Unsere Überlegung ist es, dass eine Klassifikationserweiterung hier helfen könnte, um noch genauer Fragetypen vorherzusagen. Das Schema von Li und Roth bietet mehr als nur 11 Klassen. Außerdem könnte man mit einem anderen und eventuell besseren Datensatz die Erfolgsrate erhöhen.

Spacy NER zeigt uns bei diesem Projekt zwei Probleme. Einer davon ist, dass das Model nicht in der Lage ist Länder und Städte zu unterscheiden. Diese fallen alle unter derselben Entität "GPE". Eine mögliche Lösung ist es die Bibliothek "geonamescache" zu verwenden. Diese bietet eine Liste von Ländern und Städten, so dass man die Entität besser unterscheiden kann nach Abgleich mit der Liste.

Das zweite Problem ist, dass das Model Tiere und Essen gar nicht erkennt. Eine mögliche Lösung ist es, dass vorhanden Model weiter zu trainieren oder man trainiert ein komplett eigenes Model. Beide Ansätze können auch für das erste Problem angewendet werden.

Fazit

Das Projekt verfolgt das Ziel, ein Frage-Antwort-System auf Basis einer umfangreichen Wikibase zu entwickeln und die Leistungsfähigkeit verschiedener Methoden im Rahmen eines Wettbewerbs zu vergleichen.

Unser Lösungsansatz integriert verschiedene Schritte, um eine präzise Beantwortung von Fragen aus der Wikibase zu ermöglichen. Die Kombination von Elasticsearch für die Volltextsuche, FastText für die semantische Ähnlichkeit, Spacy und LogisticRegression classifier für die Entitätserkennung erlaubt es uns, relevante Dokumente zu identifizieren, semantische Ähnlichkeiten zu berechnen und spezifische Informationen zu extrahieren. Die Implementierung dieser Schritte ermöglicht eine effektive und präzise Beantwortung von Fragen aus der Wikibase.

Unser eigenes Evaluationsskript zeigt eine Erfolgsrate von etwa 37.14% für den Evaluationsdatensatz, während unser Klassifikator eine Erfolgsrate von 67% erreicht. Dennoch gibt es verschiedene Limitierungen des Systems, darunter Probleme mit Synonymen, falsch extrahierten Sätzen, und Einschränkungen in der Entitätserkennung. Potenzielle Lösungsansätze wie die Verwendung von erweiterten Klassifikationsmethoden, verbesserte Normalisierungstechniken und eine genauere Entitätserkennung könnten dazu beitragen, diese Limitierungen zu überwinden und die Leistung des Systems weiter zu verbessern.