



สมาคมนวัตกรรมและเทคโนโลยีสร้างสรรค์ ร่วมกับ มหาวิทยาลัยเทคโนโลยีราชมงคลกรุงเทพ

การประชุมวิชาการระดับชาติเชิงสร้างสรรค์ ครั้งที่ 9

21-23
AUG.
2024

2024
CreTech

Creative
Technology &
Artificial Intelligence
at Cape Racha Hotel, Sriracha, Chon Buri

โปรแกรมประยุกต์บนเว็บเพื่อแนะนำภาพยนตร์แบบข้อความด้วย DistilBERT Text-based Movie Recommendation Web Application with DistilBERT

สุพิชชา จำปาทอง¹, ณัฐวุฒิ กลิ่นสวัสดิ์¹, นพคุณ อนัตกิจถาวร¹,

ชูเกียรติ วรสุชีป¹, และ วรินทร์ วัฒนพรรพรม^{1*}

Supischa Champathong¹, Nattawut Klinsawat¹, Noppawut Anatkijtaoworn¹,

Chukiet Worasuchee¹, and Warin Wattanapornprom^{1*}

¹สาขาวิทยาการคอมพิวเตอร์ประยุกต์ ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ประเทศไทย

¹Applied Computer Science Program, Department of Mathematics, Faculty of Science,

King Mongkut's University of Technology Thonburi, Thailand

*Tel: 66-954956245, E-mail: warin.wat@kmutt.ac.th

บทคัดย่อภาษาไทย

โครงการนี้มีวัตถุประสงค์เพื่อพัฒนาเว็บไซต์สำหรับแนะนำภาพยนตร์ให้แก่ผู้ใช้งาน โดยช่วยให้ผู้ใช้งานสามารถค้นหาภาพยนตร์ที่ตรงกับความต้องการและลดเวลาในการค้นหาได้อย่างมีประสิทธิภาพ ระบบนี้เริ่มต้นจากการกำหนดขอบเขตและข้อกำหนดความต้องการของระบบ ออกแบบประสบการณ์ผู้ใช้ (UX) และส่วนต่อประสานผู้ใช้ (UI) บนเว็บไซต์ รวมถึงการพัฒนาแนะนำภาพยนตร์เพื่อให้มีประสิทธิภาพสูงสุด โดยใช้ชุดข้อมูลจากเว็บไซต์ Kaggle ซึ่งประกอบด้วยภาพยนตร์จำนวน 45,000 เรื่องที่ฉายก่อนเดือนกรกฎาคม พ.ศ. 2560 ระบบแนะนำภาพยนตร์พัฒนาขึ้นโดยใช้เทคนิค DistilBERT ร่วมกับ Cosine Similarity และ K-Nearest Neighbor (KNN) เพื่อวิเคราะห์คำสำคัญและค้นหาภาพยนตร์ที่ใกล้เคียง ผลลัพธ์ที่ได้คือภาพยนตร์ที่ระบบแนะนำ ผู้ใช้งานแสดงความพึงพอใจทั้งในส่วนหน้าของเว็บไซต์และระบบแนะนำภาพยนตร์ โดยโครงการนี้ประสบความสำเร็จตามวัตถุประสงค์ที่ตั้งไว้

คำสำคัญ : การแนะนำภาพยนตร์ ดิสทิลเบิร์ต การลดเวลาการประมวลผล การวัดระยะแบบโคไซน์

ABSTRACT

This project aims to develop a website for recommending movies to users, enabling them to find movies that match their preferences and reduce search time efficiently. The project began with defining the system's scope and requirements, designing the user experience (UX) and user interface (UI) on the website, and developing the movie recommendation system to optimize its performance. The data set used for training the system was collected from Kaggle, consisting of 45,000 movies released before July 2017. The



recommendation system employs DistilBERT combined with Cosine Similarity and K-Nearest Neighbor (KNN) techniques to analyze keywords and find similar movies. The results are movie recommendations provided by the system, with users expressing satisfaction with both the website and the recommendation system. DistilBERT reduced the inference time from 5.5 minutes to 3.3 minutes without compromising the recommendation quality. The project successfully achieved its objectives.

Keyword: Movie Recommendation; DistilBERT; Inference Time Reduction; Cosine Similarity

1. บทนำ

อุตสาหกรรมภาพยนตร์ได้เติบโตอย่างก้าวกระโดดในสหรัฐอเมริกาและแคนาดา โดยมีการผลิตภาพยนตร์มากกว่า 700 เรื่องต่อปี [1] ภาพยนตร์เหล่านี้มีทั้งที่ผลิตโดยผู้ผลิตรายใหญ่ เช่น Disney, Warner Bros, Paramount Pictures และผู้ผลิตรายย่อย เช่น A24 รวมถึงภาพยนตร์อินดี้ต่าง ๆ ซึ่งมักจะไม่ใช่ที่รู้จักจากผู้ผลิตรายใหญ่เนื่องจากข้อจำกัดด้านงบประมาณในการประชาสัมพันธ์ ทำให้ภาพยนตร์ดี ๆ หลายเรื่องต้องถูกมองข้ามไปอย่างน่าเสียดาย

ในช่วงเริ่มต้นของการระบาดของ COVID-19 แพลตฟอร์มสตรีมมิ่งได้รับความนิยมเพิ่มขึ้นอย่างมาก โดยเฉพาะ Netflix ที่มีสมาชิกเพิ่มขึ้นอย่างมหาศาล [2] นอกจากนี้ยังมีแพลตฟอร์มสตรีมมิ่งใหม่ ๆ เพิ่มขึ้น เช่น Disney+, Paramount+, และ Apple TV+ ทำให้ผู้คนมีช่องทางในการดูภาพยนตร์เพิ่มขึ้นอย่างมาก จากสถิติในปัจจุบันผู้คนได้เปลี่ยนความชื่นชอบในการดูภาพยนตร์จากการดูในโรงภาพยนตร์ในปี ค.ศ. 2018 ที่ 28% และ 15% ของผู้คนที่ชื่นชอบการดูผ่านสตรีมมิ่ง ไปเป็นการดูผ่านสตรีมมิ่งที่ 36% ในปี ค.ศ. 2020 ส่วนการดูผ่านโรงภาพยนตร์นั้นเหลือเพียง 14% เท่านั้น [3]

อย่างไรก็ตาม การหาภาพยนตร์ที่ตรงกับความต้องการในการรับชมของผู้รักในภาพยนตร์ยังคงเป็นเรื่องท้าทาย เนื่องจากจำนวนภาพยนตร์ที่เพิ่มขึ้น ทางคณะผู้จัดทำจึงได้พัฒนาเว็บไซต์สำหรับการแนะนำภาพยนตร์ให้แก่ผู้ใช้งานผ่านการกรอกข้อมูลคำหลัก ช่วยให้ผู้ใช้งานสามารถ

ค้นหาภาพยนตร์ที่ตรงกับความต้องการและลดเวลาในการค้นหา

โครงการนี้มีวัตถุประสงค์เพื่อสร้างโปรแกรมประยุกต์บนเว็บที่ช่วยลดระยะเวลาในการค้นหาภาพยนตร์และเพิ่มโอกาสในการค้นหาภาพยนตร์ที่ไม่ค่อยได้รับการประชาสัมพันธ์ โดยมีการแสดงผลแบบเรสปอนซีฟ (responsive) ที่สามารถปรับเปลี่ยนขนาดของเว็บไซต์ให้เหมาะสมกับการแสดงผลบนหน้าจอที่มีขนาดและความละเอียดแตกต่างกัน ทั้งนี้จะใช้คำค้นในการแนะนำภาพยนตร์เท่านั้น นอกจากนี้ยังสามารถแสดงข้อมูลภาพยนตร์ได้

ขั้นตอนการดำเนินโครงการนี้เริ่มต้นที่การศึกษากำหนดขอบเขตและข้อกำหนดความต้องการของระบบ จากนั้นจะออกแบบ UX/UI ของเว็บไซต์ การทำงานฝั่งหน้าเว็บและฝั่งหลังบ้าน (backend) และพัฒนาและปรับปรุงการทำงานส่วนแนะนำภาพยนตร์ (recommendation) ให้มีประสิทธิภาพในระดับที่น่าพอใจ

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 การประมวลผลภาษาธรรมชาติ (Natural Language Processing)

การประมวลผลภาษาธรรมชาติ (Natural Language Processing หรือ NLP) [4] เป็นสาขาหนึ่งของปัญญาประดิษฐ์ที่ช่วยให้คอมพิวเตอร์สามารถเข้าใจความหมายของภาษามนุษย์ในข้อความได้อย่างใกล้เคียงกับความสามารถของมนุษย์เอง NLP เป็นองค์ความรู้ที่รวม



การศึกษาด้านภาษาศาสตร์เชิงคำนวณ ภาษามนุษย์ สถิติ และการเรียนรู้ของเครื่อง (Machine Learning) เข้าด้วยกัน

2.2 K-nearest neighbor (KNN) [5] เป็นขั้นตอนวิธีที่เรียบง่าย สำหรับทำนายผลลัพธ์โดยใช้ความใกล้เคียงของข้อมูล ในงานนี้ ได้ปรับ KNN ให้ใช้งานร่วมกับ Cosine Similarity ในการหาความใกล้เคียงแทนการใช้ Euclidean Distance โดยขั้นตอนในการทำงานนั้นจะวัดความใกล้เคียงของค่าแต่ละค่าด้วย Cosine Similarity ข้อมูลที่มีค่าความใกล้เคียงสูงที่สุดกับข้อมูลที่พิจารณาจำนวน K ตัวจะถูกเลือกก่อน

2.3 Cosine Similarity [6] เป็นมาตรวัดความใกล้เคียงระหว่างเวกเตอร์ 2 เวกเตอร์ โดยการคำนวณค่ามุมโคไซน์ (Cosine) ระหว่างเวกเตอร์ทั้งสอง ค่าที่คำนวณได้จะอยู่ในช่วง -1 ถึง 1 โดยค่าที่เป็น -1 หมายถึงเวกเตอร์ทั้งสองมีความแตกต่างกันอย่างสิ้นเชิง ค่าที่เป็น 0 หมายถึงไม่มีความใกล้เคียงกัน และค่าที่เป็น 1 หมายถึงเวกเตอร์ทั้งสองมีความเหมือนกันสูตรของ Cosine Similarity มีดังนี้:

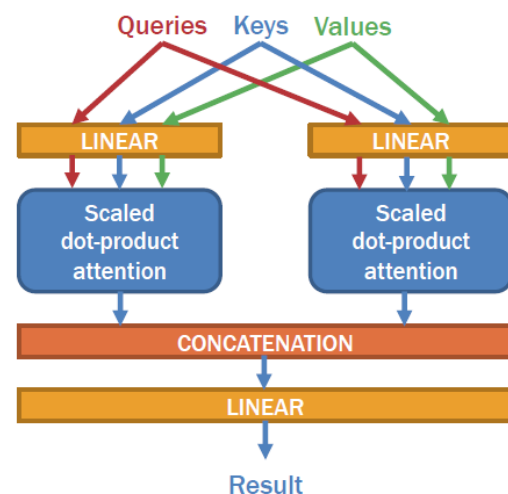
$$\cos \theta = A \cdot \frac{B}{||A|| ||B||}$$

โดยที่ $A \cdot B$ เป็นผลคูณเชิงสเกลาร์ของเวกเตอร์ A และ B และ $||A||$ เป็นขนาดของเวกเตอร์ A ส่วน $||B||$ เป็นขนาดของเวกเตอร์ B ในการคำนวณ Cosine Similarity มักมีการหารด้วยผลคูณของขนาดของเวกเตอร์ทั้งสอง เพื่อให้ผลลัพธ์ไม่ขึ้นอยู่กับขนาดของเวกเตอร์ ทำให้ผลลัพธ์อยู่ในช่วง -1 ถึง 1

2.3 DistilBERT Bidirectional Encoder Representations from Transformers (BERT) [5] เป็นโมเดลภาษาที่ช่วยเพิ่มประสิทธิภาพในการทำงานด้านการประมวลผลภาษาธรรมชาติ (NLP) อย่างมาก โดยมีความสามารถในการประมวลผลข้อความจากทั้งสองทิศทางพร้อมกัน พัฒนาโดย Google AI เดิมทีโมเดลภาษาส่วนใหญ่จะประมวลผลข้อความได้เพียงทิศทางเดียว ไม่ว่าจะเป็นจาก

ซ้ายไปขวา หรือจากขวาไปซ้ายเท่านั้น แต่ BERT สามารถประมวลผลข้อมูลจากทั้งสองทิศทาง คือจากซ้ายไปขวาและจากขวาไปซ้ายพร้อมกัน แล้วนำผลลัพธ์มารวมกัน ทำให้ BERT สามารถวิเคราะห์บริบทจากข้อความได้ดีซึ่งมากขึ้น และมีความเร็วสูง เนื่องจากขั้นตอนในการหาบริบทสามารถทำการประมวลผลแบบขนาน (Parallelization) ได้

ส่วนประกอบหลักที่ทำให้ BERT สามารถวิเคราะห์บริบทได้อย่างมีประสิทธิภาพคือ Multi Head Attention ซึ่งประกอบด้วย Attention จำนวน 8 ตัว แต่ละตัวจะมีการคำนวณที่แยกจากกัน ทำให้ได้ผลลัพธ์ที่แตกต่างกัน จากนั้นจะมีการรวบรวมผลลัพธ์เหล่านี้เพื่อใช้ในขั้นตอนการประมวลผลถัดไป



รูปที่ 1 Multi Head Attention

ข้อมูลนำเข้าของ Attention มาจากข้อความที่ผ่านการแปลงเป็นตัวเลขด้วย Input Embedding และมีการเพิ่มค่าตำแหน่งของคำด้วย Positional Encoding เพื่อให้ได้เวกเตอร์ที่แทนคำต่าง ๆ จากนั้นจะมีการจัดกลุ่มข้อมูลออกเป็น 3 ชุด ได้แก่ เมทริกซ์ Query, Key และ Value ซึ่งมาจากแนวคิดของการขอข้อมูลจากฐานข้อมูล โดยเราจะใช้ค่าที่ต้องการค้นหาเทียบกับ Key ในฐานข้อมูล หากพบค่าที่ตรงกัน เราจะใช้ข้อมูลจาก Key นั้น



ในขั้นตอน Scaled Dot-Product Attention หรือเรียกสั้น ๆ ว่า Attention ที่ BERT ใช้ จะมีการคำนวณหาค่าความใกล้เคียงระหว่าง Query และ Key จากการคูณเมทริกซ์ และปรับผลลัพธ์ให้มีค่าที่เล็กลงเพื่อให้ง่ายต่อการประมวลผล โดยใช้รากที่สองของขนาดของเวกเตอร์ Key และใช้ Softmax Activation Function เพื่อแปลงผลลัพธ์ที่ผ่านการลดค่าให้อยู่ในรูปของค่าความน่าจะเป็นที่เรียกว่า Attention Weight ในที่สุดจะใช้ค่า Attention Weight ในการถ่วงค่า Value ด้วยการคูณเมทริกซ์ ทำให้โมเดลสามารถเข้าใจว่าควรให้ความสนใจกับคำใดมากกว่า

DistilBERT [7] เป็น BERT ที่มีขนาดเล็กและเร็วกว่า ด้วยการลดจำนวน Layer ของ Encoder ที่ใช้ลงจาก 12 ชั้น เหลือ 6 ชั้น ลดจำนวนของพารามิเตอร์ที่ใช้ลงจาก 110 ล้านตัว เหลือ 66 ล้านตัว มีการใช้เทคนิคการกลั่นความรู้ในการลดขนาดของ DistilBERT โดยเป็นการฝึก DistilBERT ให้เลียนแบบการทำนายผลลัพธ์ของโมเดล BERT ทำให้ DistilBERT ยังคงมีประสิทธิภาพใกล้เคียงกับ BERT โดยในงานนี้ ได้เลือกใช้ DistilBERT ในการทำ Word Embedding แทนการทำ Word Embedding ทั่วไป เพื่อใช้บริบทของภาพยนตร์ในการทำภาพยนตร์ที่ใกล้เคียง

2.4 Yet Another Keyword Extractor (YAKE!) เป็นเครื่องมือสกัดคำค้น (keyword extractor) แบบอัตโนมัติที่มีน้ำหนักเบาและไม่ต้องต้องการการฝึกสอน (unsupervised) โดย YAKE! ไม่ต้องการพินิจพิเคราะห์, external corpus, ขนาดของข้อความหรือโดเมนใด ๆ [8] ทำให้สามารถสกัดคำค้นที่เกี่ยวข้องกับข้อความได้อย่างมีประสิทธิภาพเมื่อทำการเปรียบเทียบระหว่าง unsupervised keyword extractor กับ supervised keyword extractor พบว่า YAKE! มีความสามารถที่โดดเด่นกว่าตัวอื่น ๆ

ขั้นตอนการทำงานของ YAKE! มีทั้งหมด 5 ขั้นตอน [9] เริ่มต้นด้วยการทำ text pre-processing และเลือกคำค้นมาเป็นตัวเลือก (candidate) จากนั้นจะทำการ feature extraction เพื่อหาคุณสมบัติ (feature) ทั้งหมด 5 อย่าง ได้แก่

1. T_{case} (Casing)
 2. $T_{position}$ (Term Position)
 3. TF_{Norm} (Term Frequency Normalization)
 4. T_{rel} (Term Related To Context)
- และ

5. $T_{sentence}$ (Term Different Sentence)

เมื่อทำ feature extraction เสร็จแล้วจะนำคุณสมบัติที่ได้ไปคำนวณคะแนนของคำ (term score) โดยใช้สูตรดังนี้:

$$S(t) = \frac{T_{rel} * T_{position}}{T_{case} + \frac{TF_{Norm}}{T_{rel}} + \frac{T_{sentence}}{T_{rel}}}$$

การหาค่า TF_{Norm} และ $T_{sentence}$ ด้วย T_{rel} นั้นเพื่อเพิ่มค่าความสำคัญให้กับคำที่ปรากฏในข้อความบ่อยครั้ง โดยเฉพาะเมื่อคำเหล่านั้นเกี่ยวข้องกับข้อความสูง หรือมีค่า T_{rel} ต่ำ ยิ่งค่า $S(t)$ น้อยเท่าไร ความสำคัญของคำจะยิ่งมากขึ้น จากนั้นจะสร้าง n-gram และคำนวณคะแนนของคำค้นที่คัดเลือกมาโดยใช้สูตรดังนี้:

$$S(kw) = \frac{\prod_{t \in kw} S(t)}{KF(kw) * (1 + \sum_{t \in kw} S(t))}$$

2024
CreTechCreative
Technology &
Artificial Intelligence
at Cape Racha Hotel, Sriracha, Chon Buri

kw จะแสดงถึง key phrase ที่เลือกเป็น candidate และ $S(kw)$ จะเป็นคะแนนในการคัดเลือกคำค้นที่สำคัญที่สุด ตัวเศษของสมการจะเป็นผลลัพธ์ของคะแนน unigram ในเวลานั้น ๆ และจากที่ได้กล่าวในสมการ Term Score ยิ่งคะแนนน้อยจะยิ่งสำคัญมาก โดยเพื่อแก้ปัญหาในส่วนนี้จึงจะทำการหาผลรวมของคะแนนทั้งหมดและถ่วงน้ำหนักความสำคัญด้วยความถี่ของคำค้นหรือ $KF(kw)$

3. วิธีการดำเนินการวิจัย

ระบบนี้พัฒนาขึ้นโดยใช้ React.js Library สำหรับสร้างเว็บไซต์ของระบบ ใช้ Figma สำหรับออกแบบเว็บไซต์ และใช้ FastAPI และ Uvicorn ในการสร้างแม่ข่าย ซึ่งเป็น Application Programming Interface (API) สำหรับการใช้งานร่วมกับเว็บไซต์ ขั้นตอนการพัฒนาเริ่มต้นจากการออกแบบ User Interface (UI) และ User Experience (UX) ตามขอบเขตใน Figma ก่อนที่จะพัฒนาเว็บไซต์ด้วย React.js จากนั้นพัฒนาส่วน Server ด้วย Uvicorn และ FastAPI รวมถึงพัฒนาระบบแนะนำภาพยนตร์

ในการพัฒนาระบบแนะนำ มีการใช้ข้อมูลจาก Kaggle ซึ่งเป็นชุดข้อมูลภาพยนตร์จำนวน 45,000 เรื่อง ที่ถ่ายก่อนเดือนกรกฎาคม พ.ศ. 2560 โดยคัดเลือกเหลือ 28,000 เรื่องที่มีชื่อเป็นภาษาอังกฤษ ข้อมูลเหล่านี้ผ่านการ Preprocessing เพื่อคัดกรองและทำความสะอาดข้อมูล จากนั้นสร้าง Word Embedding สำหรับการหาความใกล้เคียงของคำอธิบายภาพยนตร์ด้วย DistilBERT และใช้ Word Embedding นี้ในการคำนวณหาความใกล้เคียงด้วย Cosine Similarity

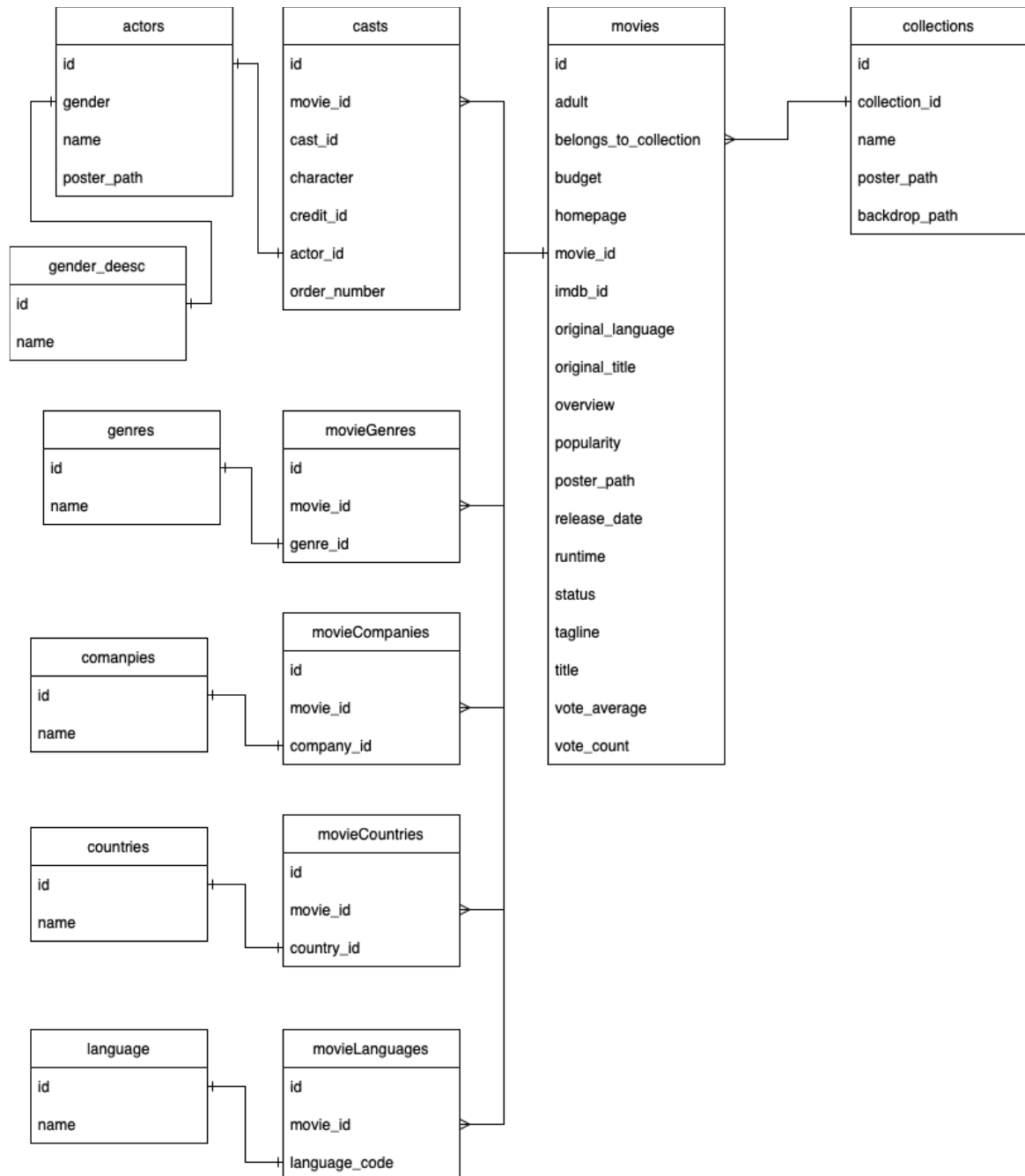
ในการใช้งานจริง ระบบจะรับข้อความและแนะนำภาพยนตร์โดยมีการหาค่าใกล้เคียง และใช้ KNN ในการเลือกภาพยนตร์ที่มีความใกล้เคียงจากค่าใกล้เคียงสูงที่สุด โดยการเทียบค่าของ Cosine Similarity และนำภาพยนตร์จากแต่ละค่าใกล้เคียงมาเรียงลำดับตามค่า Cosine Similarity อีกครั้งหนึ่ง

4. ผลการวิจัยและอภิปรายผล

4.1 ผลการทดสอบโปรแกรมประยุกต์

ในส่วนของการพัฒนาโปรแกรมประยุกต์บนเว็บนั้น ประกอบด้วยทั้งหมด 6 หน้า ซึ่งมีรายละเอียดในแต่ละขั้นตอนดังนี้ หน้าที่หนึ่งคือหน้าค้นหา ซึ่งผู้ใช้สามารถใส่คำค้นลงในช่องค้นหา แล้วระบบจะทำการค้นหาภาพยนตร์จากข้อมูลทั้งหมดใน Dataset และแสดงผลลัพธ์ในหน้าผลลัพธ์การค้นหา เมื่อได้ผลลัพธ์จากคำค้น ระบบจะแสดงรูปภาพ ชื่อ วันเดือนปีที่เข้าฉาย ความยาวของภาพยนตร์ และเนื้อเรื่องย่อ หน้าที่สองคือหน้ารายละเอียดภาพยนตร์ ซึ่งจะแสดงรายละเอียดต่าง ๆ เช่น ชื่อภาพยนตร์ ชื่อคอลเลกชัน ประเภทภาพยนตร์ วันเดือนปีที่เข้าฉาย ความยาวของภาพยนตร์ เนื้อเรื่องย่อ นักแสดง และภาพยนตร์ที่ระบบแนะนำเพิ่มเติม หน้าที่สามคือหน้ารายละเอียดนักแสดง ซึ่งผู้ใช้สามารถดูข้อมูลเพิ่มเติมเกี่ยวกับนักแสดง เช่น ชื่อ นามสกุล และภาพยนตร์ที่นักแสดงได้ร่วมแสดง

ในส่วนของการพัฒนาหลังบ้านถูกสร้างขึ้นโดยใช้ภาษา Python เนื่องจากมี library รองรับที่หลากหลาย โดยใช้ FastAPI เป็น Web framework ซึ่งเลือกใช้ FastAPI เนื่องจากความเรียบง่ายและรวดเร็วในการออกแบบ Backend ในส่วนของฐานข้อมูลใช้ SQLite 3 เนื่องจากความถนัดของผู้พัฒนา และใช้ Yake library ในการสกัดคำค้นออกมาจากเรื่องย่อของภาพยนตร์ โดยข้อมูลที่ได้รับมาในรูปแบบ CSV ถูกจัดเก็บในฐานข้อมูลโดยแบ่งเป็นตารางต่าง ๆ รวม 12 ตารางดังรูปที่ 2 ได้แก่



รูปที่ 2 ความสัมพันธ์ของแต่ละตาราง

1. casts ตารางรวบรวมข้อมูลของบทบาทนักแสดงในภาพยนตร์เรื่องต่าง ๆ
2. actors ตารางรวบรวมข้อมูลรายละเอียดของนักแสดงต่าง ๆ เช่น วันเกิด เพศ ชื่อ

3. collections ตารางรวบรวมข้อมูลเกี่ยวกับชุดหนังต่าง ๆ โดยหนังที่เป็นภาคต่อ จะเป็นหนังที่อยู่ใน collection เดียวกัน



4. movies ตารางรวบรวมข้อมูลรายละเอียดต่าง ๆ ของภาพยนตร์ เช่น ชื่อ เรื่องย่อ วันที่ฉาย และความยาวภาพยนตร์

5. genres เป็นตารางที่รวบรวมข้อมูลเกี่ยวกับประเภทของภาพยนตร์ต่าง ๆ โดยจะมีข้อมูลอย่างชื่อของประเภทแต่ละประเภทอยู่

6. movieGenres เป็นตารางเก็บข้อมูลประเภทของภาพยนตร์อยู่ในภาพยนตร์แต่ละเรื่องนั้นสามารถมีประเภทได้มากกว่า 1 ประเภท จึงจำเป็นต้องมีการสร้างตารางเพิ่มขึ้น

7. companies เป็นตารางเก็บรวบรวมข้อมูลชื่อบริษัทผู้ผลิตภาพยนตร์ต่าง ๆ

8. countries ตารางเก็บข้อมูลชื่อประเทศต่าง ๆ ที่มีอยู่ใน dataset

9. languages ตารางเก็บข้อมูลชื่อภาษาต่าง ๆ ที่มีอยู่ใน dataset

10. movieCompanies เป็นตารางเก็บข้อมูลว่าภาพยนตร์แต่ละเรื่องนั้น บริษัทใดเป็นผู้ผลิต

11. movieCountries เป็นตารางเก็บข้อมูลว่าภาพยนตร์แต่ละเรื่องนั้น มีการผลิตขึ้นโดยประเทศใด

12. movieLanguages เป็นตารางเก็บข้อมูลว่าภาพยนตร์แต่ละเรื่องนั้น มีการใช้ภาษาใดเป็นหลักในภาพยนตร์

หลังจากสร้างตารางแล้วก็ทำการเชื่อมต่อ API กับฐานข้อมูลผ่าน library sqlite3 บน Python และทำการ Hosting ผ่านเว็บไซต์ Heroku ภายใต้อัปเดตชื่อ muvi-backend

หลังจากเสร็จสิ้นขั้นตอนการพัฒนาโปรแกรมประยุกต์บนเว็บ เราได้นำเว็บไซต์ไปทดสอบกับผู้ใช้งาน เพื่อประเมินความพึงพอใจของผู้ใช้งาน กลุ่มตัวอย่างในการทดสอบประกอบด้วยผู้ที่ผ่านเกณฑ์การคัดเลือก โดยมีเกณฑ์คือดูภาพยนตร์ไม่น้อยกว่า 1 เรื่องต่อสัปดาห์ จำนวนทั้งหมด 10 คน แบ่งเป็นนักเรียน-นักศึกษา อายุ 18-24 ปี จำนวน 5 คน พนักงานบริษัทเอกชน อายุ 18-24 ปี จำนวน 2 คน อายุ 25-35 ปี จำนวน 2 คน และคนว่างงาน อายุ 18-24 ปี

จำนวน 1 คน การทดสอบนี้ดำเนินการทั้งในรูปแบบออนไลน์และออนไลน์ เราได้กำหนด Task Scenario ให้แก่ผู้ใช้งานเพื่อจำลองสถานการณ์การใช้งานต่าง ๆ ดังนี้:

Task Scenario 01: การค้นหา

ผู้ใช้งานต้องการค้นหาภาพยนตร์ซักเรื่อง แต่ไม่แน่ใจว่าจะดูเรื่องอะไรดี จึงเข้ามาใช้งานเว็บไซต์ Muvi เพื่อค้นหาภาพยนตร์โดยใช้คำค้นที่มีอยู่ในใจ

Task Scenario 02: การดูข้อมูล

เมื่อผู้ใช้งานได้ค้นหาภาพยนตร์และพบเรื่องที่ต้องการแล้ว จึงอยากดูข้อมูลเพิ่มเติมเกี่ยวกับประเภทของภาพยนตร์ ความยาว ปีที่ฉาย นักแสดง และภาพยนตร์อื่น ๆ ที่เกี่ยวข้อง

Task Scenario 03: การค้นหาโดยใช้หลายคำค้น

หากผู้ใช้งานยังไม่พอใจกับภาพยนตร์ที่ได้มา จะกลับมาทำการค้นหาอีกครั้งโดยใช้หลายคำค้นพร้อม ๆ กัน เพื่อให้ได้ภาพยนตร์ที่ตรงกับความต้องการมากขึ้น

ผู้ใช้งานจะให้คะแนนความยาก-ง่ายในการใช้งานแต่ละ Task Scenario โดยคะแนน CES เรียงจากง่ายที่สุด 1 ไปจนถึงยากที่สุด 7

ผลลัพธ์จากการทดสอบการใช้งานของผู้ใช้งานถูกวัดเป็นระดับคะแนน CES ซึ่งรายละเอียดของคะแนนจะรวมจากผู้ใช้งานแต่ละคนและจัดเรียงตามระดับความยาก-ง่ายในการใช้งาน

หลังจากการทดสอบใช้งานระบบเสร็จสิ้น ผู้ใช้งานได้รับการขอให้ตอบแบบสำรวจเกี่ยวกับเว็บไซต์และระบบแนะนำภาพยนตร์ พบว่าความง่ายในการใช้งานเว็บไซต์ได้รับคะแนนเฉลี่ย 5.7 จากคะแนนเต็ม 7 คะแนน โดยมีผู้ใช้งานบางส่วนให้คะแนนต่ำกว่าค่าเฉลี่ย โดยองค์ประกอบที่ผู้ใช้งานชื่นชอบมากที่สุดคือการจัดวางเลย์เอาต์ที่ได้รับการตอบรับเชิงบวกจาก 50% ของผู้ตอบแบบสอบถาม ส่วนความเร็วในการตอบสนองของเว็บไซต์ก็เป็นอีกปัจจัยหนึ่งที่ได้รับคะแนนชื่นชอบ โดย 30% ของผู้ตอบรู้สึกพึงพอใจ



2024 CreTech

Creative
Technology &
Artificial Intelligence
at Cape Racha Hotel, Sriracha, Chon Buri

ขณะที่การใช้สื่อบอกว่าเป็นองค์ประกอบที่เหมาะสมโดย 20% ของผู้ตอบ

ในทางตรงกันข้าม บางส่วนของผู้ใช้งานรู้สึกไม่พอใจกับการจัดวางเลย์เอาต์และความเร็วในการตอบสนองของเว็บไซต์ โดย 30% ของผู้ตอบแบบสอบถามระบุว่าเลย์เอาต์ของเว็บไซต์เป็นองค์ประกอบที่พวกเขาชอบน้อยที่สุด และอีก 30% รู้สึกว่าความเร็วในการตอบสนองของระบบยังไม่เพียงพอ ขณะที่ขนาดตัวอักษรและขนาดหน้าจอมีผู้ใช้บางส่วนระบุว่าไม่ถูกใจ โดยเฉพาะขนาดตัวอักษรที่ได้รับการตอบรับลดลงจาก 20% ของผู้ตอบ แบบสำรวจนี้ช่วยให้เห็นถึงพื้นที่ที่จำเป็นต้องปรับปรุงเพื่อเพิ่มความพึงพอใจของผู้ใช้งานในอนาคต

4.2 ผลการทดสอบระบบแนะนำ

ในการทดสอบระบบแนะนำภาพยนตร์ เราได้ทำการเปรียบเทียบระหว่างการใช้งานโมเดล BERT และ DistilBERT โดยวัดจากเวลาที่ใช้ในการประมวลผล (inference time) และประสิทธิภาพโดยรวมของการแนะนำภาพยนตร์ โดยสภาพแวดล้อมที่ทำการวัดประสิทธิภาพคือ MacBook Pro 2019 หน่วยประมวลผล Intel® Core™ i5-8579U 2.4 GHz หน่วยความจำ 8 GB โดยใช้ Python 3.11 ฐานข้อมูลที่ใช้เรียนคือ Movie Time! Open dataset [9] ที่มีภาพยนตร์ 5,043 เรื่อง

1. เวลาที่ใช้ในการประมวลผลการเรียน (Inference Time)

โมเดล BERT ซึ่งเป็นโมเดลที่มีขนาดใหญ่และซับซ้อน มีเวลาเฉลี่ยในการประมวลผลอยู่ที่ 5.5 นาทีต่อการเรียนข้อมูลทั้งหมด ขณะที่โมเดล DistilBERT ซึ่งเป็นเวอร์ชันที่ย่อขนาดลงมาจาก BERT ใช้เวลาเฉลี่ยในการประมวลผลเพียง 3.3 นาทีเท่านั้น การลดลงของเวลาการประมวลผลนี้เกิดจาก DistilBERT มีจำนวนพารามิเตอร์ที่น้อยกว่า และขนาดที่เล็กกว่า ทำให้การประมวลผลข้อมูลทำได้รวดเร็วและมีประสิทธิภาพมากขึ้น

2. ประสิทธิภาพในการทำงาน (Performance)

แม้ว่า DistilBERT จะมีขนาดเล็กกว่าและพารามิเตอร์น้อยกว่า BERT แต่การทดสอบพบว่า DistilBERT ยังคงรักษาประสิทธิภาพในการแนะนำภาพยนตร์ได้ในระดับที่ใกล้เคียงกับ BERT โดยไม่สูญเสียคุณภาพอย่างมีนัยสำคัญ การทดลองแสดงให้เห็นว่าทั้งสองโมเดลสามารถแนะนำภาพยนตร์ที่สอดคล้องกับความต้องการของผู้ใช้ได้อย่างดี ซึ่งแสดงถึงประสิทธิภาพของ DistilBERT ในการประมวลผลที่รวดเร็วและแม่นยำ แม้จะมีขนาดที่เล็กลง

3. การประยุกต์ใช้ในอนาคต

ผลการเปรียบเทียบนี้แสดงให้เห็นว่า DistilBERT เป็นทางเลือกที่น่าสนใจสำหรับการใช้งานในระบบที่ต้องการการตอบสนองเร็ว โดยเฉพาะในแพลตฟอร์มที่ต้องการประมวลผลข้อมูลขนาดใหญ่และจำนวนมากในเวลาอันสั้น นอกจากนี้ DistilBERT ยังมีข้อดีในด้านการประหยัดทรัพยากรในการประมวลผล เช่น หน่วยความจำและพลังงาน ทำให้เป็นตัวเลือกที่เหมาะสมสำหรับการพัฒนาแอปพลิเคชันที่มีข้อจำกัดด้านทรัพยากร

การเลือกใช้ DistilBERT แทน BERT จึงไม่เพียงช่วยลดเวลาการประมวลผลและทรัพยากรที่ใช้ แต่ยังคงรักษาคุณภาพในการแนะนำภาพยนตร์ได้อย่างมีประสิทธิภาพ ทำให้ระบบสามารถตอบสนองต่อความต้องการของผู้ใช้ได้อย่างรวดเร็วและแม่นยำ

5. สรุปผลการวิจัย

สรุปผลการทดสอบใช้งานระบบแนะนำภาพยนตร์ พบว่าโมเดล DistilBERT มีความได้เปรียบในเรื่องความเร็วในการประมวลผล ซึ่งลดลงจาก 5.5 นาทีเป็น 3.3 นาทีเมื่อเทียบกับโมเดล BERT โดยที่ยังคงรักษาประสิทธิภาพในการแนะนำภาพยนตร์ได้อย่างดี ความพึงพอใจของผู้ใช้งานเว็บไซต์ Muvi โดยรวมอยู่ในระดับที่ดี ผู้ใช้งานชื่นชอบการจัดวางเลย์เอาต์และความเร็วในการตอบสนองของเว็บไซต์ ซึ่งเป็นจุดเด่นของระบบ อย่างไรก็ตาม

2024
CreTechCreative
Technology &
Artificial Intelligence

at Cape Racha Hotel, Sriracha, Chon Buri

ตาม ยังมีข้อเสนอแนะเกี่ยวกับการปรับปรุงขนาดตัวอักษร และการเพิ่มความเร็วในการตอบสนองที่สามารถนำไปพัฒนาให้ดีขึ้นในอนาคต

การทดสอบนี้แสดงให้เห็นถึงความสำคัญของการออกแบบ UX/UI ที่ดีและการเลือกใช้เทคโนโลยีที่เหมาะสม เพื่อตอบสนองต่อความต้องการของผู้ใช้งานได้อย่างมีประสิทธิภาพ ข้อเสนอแนะและความคิดเห็นจากผู้ใช้งานจะเป็นประโยชน์ในการพัฒนาระบบต่อไป เพื่อให้ผู้ใช้งานได้รับประสบการณ์ที่ดีที่สุดในการค้นหาและรับชมภาพยนตร์บนแพลตฟอร์ม Muvi

6. อ้างอิง

- [1] Navarro JG. U.S. & Canada: Movie releases per year 2021. Statista [Internet]. 2022 [cited 2022 Nov 29]. Available from: <https://www.statista.com/statistics/187122/movie-releases-in-north-america-since-2001/>
- [2] Netflix gets 16 million new sign-ups thanks to lockdown. BBC News [Internet]. 2020 [cited 2022 Nov 29]. Available from: <https://www.bbc.com/news/business-52376022>
- [3] Navarro JG. Theaters vs. streaming: First time movie viewing preferences in the U.S. 2020. Statista [Internet]. 2021 [cited 2022 Nov 29]. Available from: <https://www.statista.com/statistics/947757/theaters-streaming-watching-movies/>
- [4] Cloud Education, IBM. What is natural language processing? IBM [Internet]. [cited 2022 Dec 2]. Available from: <https://www.ibm.com/cloud/learn/natural-language-processing>
- [5] Horev R. Bert explained: State of the art language model for NLP. Medium [Internet]. 2018 [cited 2023 May 23]. Available from: <https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270>
- [6] Karabiber F. Cosine similarity. Learn Data Science - Tutorials, Books, Courses, and More [Internet]. [cited 2023 May 23]. Available from: <https://www.learn-datasci.com/glossary/cosine-similarity/>
- [7] Sanh V, et al. Distilbert, a distilled version of Bert: Smaller, faster, cheaper and lighter. arXiv.org [Internet]. 2020 [cited 2023 May 23]. Available from: <https://arxiv.org/abs/1910.01108>
- [8] Shrivastava I. Exploring different keyword extractors - statistical approaches. Medium [Internet]. 2020 [cited 2023 May 26]. Available from: <https://medium.com/gumgum-tech/exploring-different-keyword-extractors-statistical-approaches-38580770e282>
- [9] Movie Time! [Internet]. 2020 [cited 2023 May 26]. Available from: <https://www.kaggle.com/datasets/alexandrelemecier/movie-time/data>