Semantic Search for Car Retrieval Based on Online Reviews

Lara Solà

Importance of Car Reviews

- Critical source of information for potential buyers and manufacturers
- Offer insights into performance, features, reliability, and overall satisfaction
- Perspectives from experts and everyday users
- Better understanding of characteristics that make a product appealing

Objective

- Develop a method for extracting car vector representations from reviews
- Allow users to search cars by using natural text (Japanese)
- Utilize Solr as a search platform
- Create a more effective and personalized search experience for users

Semantic Search

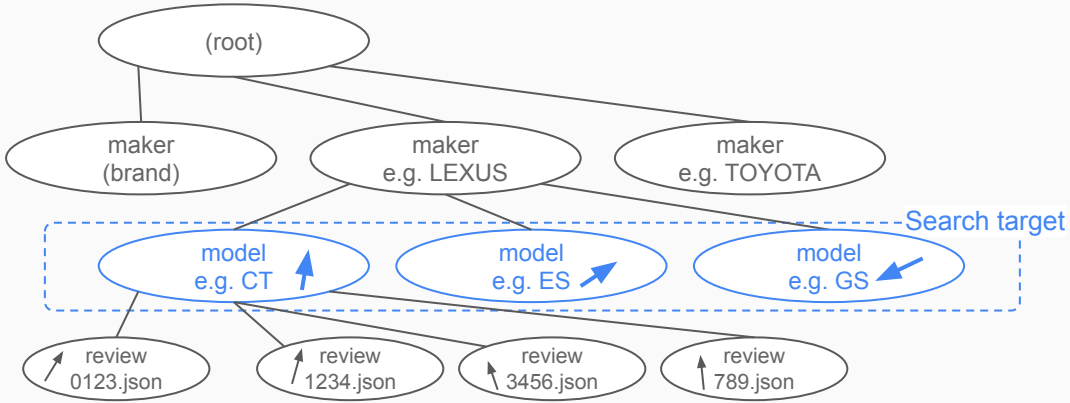
- Moving beyond traditional keyword-based search methods
- Understanding the meaning and context of user queries
- Leveraging advanced NLP techniques, including Transformer models, to process and analyze text data
- Ranking search results based on semantic relevance and contextual similarity
- Benefits of semantic search:
  - Improved search accuracy and user experience
  - Ability to handle complex, conversational, or ambiguous queries
  - Better understanding of user intent and content relationships
- Real-world applications: search engines, recommendation systems, knowledge management, and more

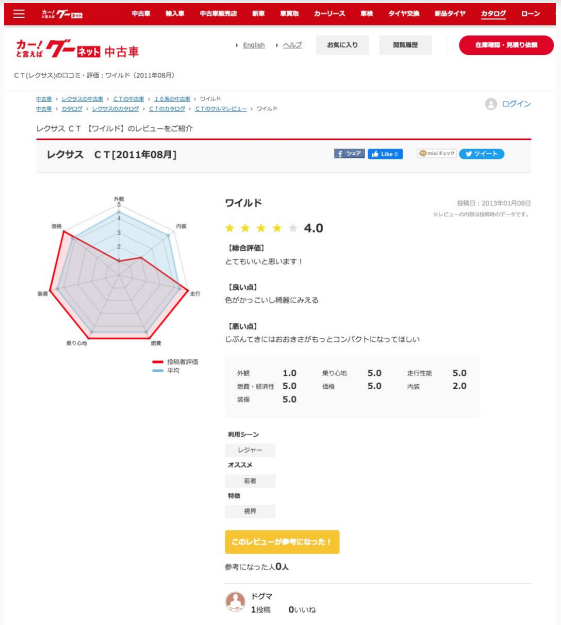
Japanese Car Reviews - A Case Study

- Objective: Extract features from Japanese car reviews for each car model
- Create vector representations to capture essential information
- Index car vectors in Solr to allow for Dense Vector Search
- Enable users to search for car models using natural language text in Japanese
  - Example 1: 家族でキャンプや買い物に行ける車（a car that can be used for camping and shopping with the family)
  - Example 2: 通勤途中に子供を学校に送れる車（a car that can take kids to school on the way to work)

Japanese Car Reviews - A Case Study
Data Source

- Car reviews crawled from the Japanese second hand car site: https://www.goo-net.com



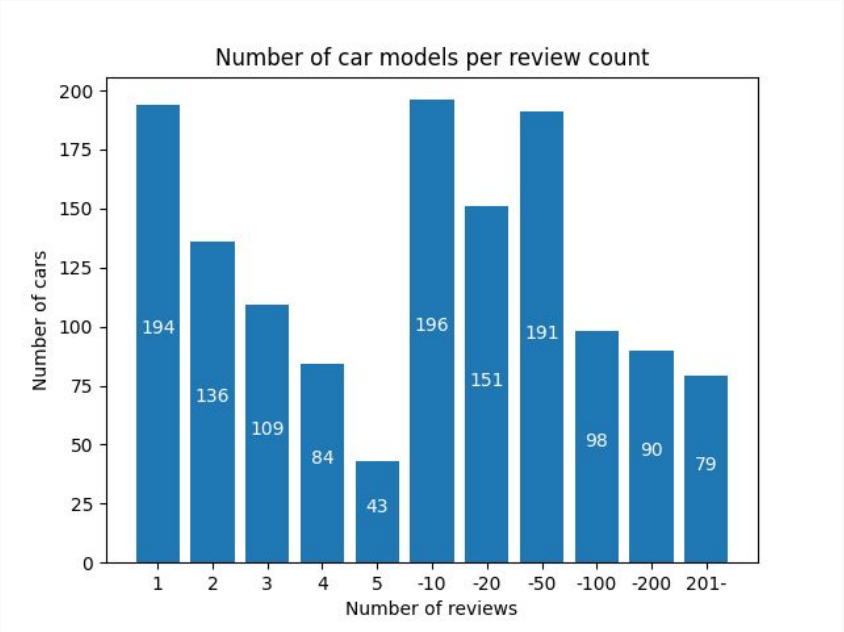


Japanese Car Reviews - A Case Study
User Review Example


```
{
  "title": "ワイルド",
  "maker": "レクサス",
  "model": "ＣＴ",
  "eval_text": "【総合評価】 とてもいいと思います！ 【良い点】 色がかっこいし綺麗にみえる 【悪い点】 じぶんてきにはおおきさがもっとコンパクトになってほしい",
  "eval_comprehensive": " とてもいいと思います！ ",
  "eval_dislike": " じぶんてきにはおおきさがもっとコンパクトになってほしい",
  "eval_like": " 色がかっこいし綺麗にみえる ",
}
```

Japanese Car Reviews - A Case Study
Data Statistics

- Number of brands: 85
- Number of models: 1,371
- Total number of reviews: 68,441
- Models without reviews are not considered for the statistics.



Overview of Vector Representation Extraction Techniques Employed

- Sentence-BERT - Efficient sentence embeddings
- Tokenization
- K-means clustering
- Method 1: Transformers model for review vector extraction and mean averaging
- Method 2: K-means clustering and Transformer model encoding
- Method 3: OpenAI's Davinci model for summarization and car vector encoding
- Comparison of methods in terms of effectiveness and practicality

Sentence-BERT - Efficient Sentence Embeddings

- Introduced by Reimers and Gurevych in 2019
- An adaptation of the BERT (Bidirectional Encoder Representations from Transformers) model specifically designed for sentence-level embeddings
- Pre-trained on large text corpora using a Siamese or triplet network structure
- Fine-tuning BERT to generate fixed-size sentence embeddings directly, instead of token-level embeddings
- Significantly faster and more efficient than vanilla BERT for sentence similarity tasks
- Widely used in applications such as semantic search and clustering
- Available through the Hugging Face's Transformers library and the official Sentence-BERT repository

Text Tokenization for SBERT Encoding and OpenAI Davinci Text 3

- Tokenization: Process of converting raw text into a sequence of tokens (words, phrases, symbols)
- Essential step to create numerical representations of text for SBERT encoding or text generation with OpenAI Davinci Text 3
- Ensures text input is formatted appropriately and compatible with the chosen model

Example:

Text = “The quick brown fox jumped over the lazy dog.”

After tokenization, the sentence is represented as a sequence of integers, with each integer representing a token:

[101, 1996, 4248, 2829, 4415, 1999, 1996, 13336, 2163, 1012, 102]

K-means Clustering for Grouping Similar Texts

- Unsupervised machine learning algorithm
- Partition data into K distinct, non-overlapping clusters
- Minimizes the within-cluster sum of squares (inertia)
- Ideal for selecting representative samples from the dataset

Method 1 - Sentence-BERT Model for Review Vector Extraction and Mean Averaging

- Choose a pre-trained Sentence-BERT model: sonoisa/sentence-bert-base-ja-mean-tokens-v2
- Group user car reviews for a specific car model
- Utilize the sonoisa Sentence-BERT model to extract vector embeddings
- Compute the mean average of vectors to obtain the car model's vector representation

Method 1 - Pre-trained Japanese sentence-BERT Model

Model name	sonoisa/sentence-bert-base-ja-mean-tokens-v2
Max sequence length	512
Output dimensions	768
Suitable Score Functions	cosine-similarity
Size	443 MB
Fine-tuned from	cl-tohoku/bert-base-japanese-whole-word-masking

Sentence-BERT Model for Review Vector Extraction and Mean Averaging

- Conclusion:
  - Mean averaging of review car vectors leads to inaccurate results as it tends to blur the distinctions among various opinions, making it difficult to capture the essence of the overall sentiment. This highlights the need for alternative methods to accurately represent car models based on user reviews.

Method 2 - K-means Clustering and SBERT Model Encoding

- Addressing sonoisa's token input limit (512 tokens)
- Strategy for selecting and concatenating a few reviews per car model:
  - Identify three representative reviews per car model
    - Apply K-means clustering with k=3
    - Choose the review closest to each cluster center
  - Concatenate selected reviews
  - Utilize the sonoisa Sentence-BERT model to encode concatenated reviews into a car vector

Method 2 - K-means Clustering and SBERT Model Encoding

- Conclusion:
  - Unpopular car models with few reviews may rank higher, leading to mismatched results
  - Selection of only models with at least 20 reviews yields better results
  - Need for further improvement to better align with user sentiment

Method 3 - Summarize Reviews and Encode

- Challenge: SBERT token limit (512) restricting the input text length
- Goal: Utilize all car reviews without losing important information
- Proposed solution: Employ text generation models for summarizing car reviews
  - Models like GPT, T5, or OpenAI's Davinci can generate concise summaries
  - Retain crucial information from the original reviews in a condensed form
- Advantages of summarization:
  - Addresses token limit constraints of Sentence-BERT
  - Efficiently processes large volumes of reviews
  - Preserves the most relevant information for vector representation
- Workflow:
  - Preprocess and tokenize car reviews
  - Generate summaries using text generation models
  - Obtain fixed-size embeddings with Sentence-BERT from the summarized texts

Method 3.1 - Summarize Reviews with MT5 Japanese Model

- Issue: Finding a free suitable pre-trained model for Japanese summary generation
- Model discovered: tsmatz/mt5_summarize_japanese from Hugging Face
  - Fine-tuned version of google/mt5-small for Japanese summarization
    - Extension of the T5 model with multilingual capabilities
    - Adaptable for various NLP tasks: translation, summarization, classification, question-answering, etc.
  - Trained on BBC news articles from the XL-Sum Japanese dataset
  - Source text should include news story elements for optimal performance (including comments)

Method 3.1 - Summarize Reviews with MT5 Japanese Model

- Limitations observed:
  - Incomplete or missed important information from car reviews:
    - "家の近所に買い物など、短距離のチョイ乗りなら十分役目ははたせますがプライスに見合った価値はあるし、個性的な車好きには強くお薦め出来るが、日本車的な気軽さを求める人にはお薦めしない。"
      ⇒ "日本の車好きにはお薦めしない。"
      “**It's worth it for the price**, and I can strongly recommend it to those who like unique cars, but I want the casualness of a Japanese car. I do not recommend it to anyone.”
      ⇒ “Not recommended for Japanese car lovers.”
  - Altered meaning:
    - "小回りが良い感じですね、あまり見掛けないですねかわいくて乗り心地もそこそこ。燃費がやたらといいで軽トラックもありますが同じくらいです。ボディサイズがコンパクトで取り回ししやすく運転しやすかったです。パワーも不足感がなく十分に楽しめました。
      ⇒"軽トラックで運転しやすかった。"
      “It has a good turning radius, I don't see it very often. The fuel consumption is very good **There are light trucks, but they are about the same**. The body size was compact and easy to handle and easy to drive. There was no feeling of lack of power and I enjoyed it enough.”
      ⇒”It was an easy truck to drive.”
  - Overlooked majority opinions:
    - "やっぱり 古い 。燃費がやたらといいです 軽トラックもありますが同じくらいです 。古いのでかっこがわるいです。"
      ⇒ "燃費がやたらと良い。"
      “**Old after all**. The fuel consumption is very good. There are light trucks, but they are about the same. Parentheses are bad **because it is old**.”
      ⇒”Very good fuel consumption.”

Method 3.1 - Summarize Reviews with MT5 Japanese Model

- Conclusion:
  - Model shows potential in capturing the overall feeling of reviews
  - Further optimization and fine-tuning needed for improved summarization in the car review domain

Method 3.2 - Summarize Reviews with OpenAI's Davinci Model

- Objective: Summarize car reviews using OpenAI's text-davinci-003 model
- Initial strategy:
  - Concatenate all reviews per car model into 3000-token chunks
  - Request summaries with a maximum of 1000 tokens
  - Recursively merge and summarize until a single summary per car review is obtained
- Pricing: $0.02 per 1k tokens for text-davinci-003
- Estimated total tokens in all reviews: 16,872,777
- Approximate cost for initial strategy: more than $320 USD
- Revised approach due to high cost:
  - Summarize up to 3000 tokens in concatenated reviews per car
  - Reduce car models sample size to 458 (models with more than 20 reviews)
  - Prioritize efficiency while preserving important information

Method 3.2 - OpenAI's Text-Davinci-003 Model

- Part of OpenAI's Codex family of models
- Based on the GPT-3 architecture: highly effective for various NLP tasks
- Powerful language model with extensive knowledge and comprehension
- Fine-tuning capabilities for specific applications or domains
- Suitable for tasks such as summarization, translation, question-answering, and more
- Accessible through OpenAI API, with a cost of $0.02 per 1k tokens
- Can be employed for summarizing large amounts of text, while maintaining essential information

Method 3.2 - Using Text-Davinci-003 Model for Summarization

1. Sign up for an API key from OpenAI
2. Install the OpenAI Python library
3. Prepare the input text to be summarized
4. Set up the Completions API request:
    a. Define the input as a "prompt"
    b. Specify the "model" as "text-davinci-003"
    c. Set the "max_tokens" for the summary length (e.g., 1000 tokens)
    d. Choose the "temperature" parameter (e.g., 0.7) for controlling randomness
    e. Configure the "top_p" parameter (e.g., 1) for controlling sampling
5. Make the API call with the configured parameters
6. Retrieve the summary from the API response

Method 3.2 - Using Text-Davinci-003 Model for Summarization

- Experimenting with different start sequences to gauge summary quality
  - Concatenate 3 car reviews and request various summaries using different `start_sequence`
- Adapting the `start_sequence` can lead to varied summary results

Example python snippet to generate a summary:

```
def summarize(text, start_sequence, max_tokens=256):
    response = openai.Completion.create(
        model="text-davinci-003",
        prompt=f"{text}\n{start_sequence}",
        temperature=0.7,
        max_tokens=max_tokens,
        top_p=1,
        frequency_penalty=0,
        presence_penalty=0
    )
    return response
```

Method 3.2 - Using Text-Davinci-003 Model for Summarization

text = “【総合評価】　おおむね満足です。中古車での購入でしたが元気に走ってくれています。　【良い点】　やっぱり燃費性能は満足です。前車は軽のターボ車で　15キロ / L程度でしたが、現在のプリウスは平均で 22キロ程度は走ってくれます。（使用エリアは大阪です。）　またゆっくり走っている分には、社内も静かです。エルグランドも所有しておりますが、それよりも静かかも・・・　【悪い点】　高速走行は余裕がないように感じます。アクセルを踏み込めばそれなりに走るんですが、とたんに燃費がダウン。またエンジン音もうるさくなるのでアクセルを踏みたくなくなります。【総合評価】　確かに燃費はいいが他車種に比べて割り高な感じはします。　周りと同じでいいという方にはおススメですが、走る楽しみは感じられません。【良い点】　燃費がいい。環境に意識していると周りから思われる。車内が静か。　【悪い点】　信号待ちで、「周りがプリウスだらけ」に遭遇する。　静かすぎて前方の歩行者が気付いてくれない。【総合評価】　老若男女を問わず誰でもすんなり受け入れられるクルマだと思います！使い勝手も従来のセダン、ハッチバックとそん色なく利便性も兼ね備えたクルマです。　【良い点】　燃費は勿論、落ち着いた外観、内装が◎　【悪い点】　ハイブリッドが故に加速に物足り無さを感じる時があります。　”

([Comprehensive evaluation] I am generally satisfied. It was a used car purchase, but it is running well. [Good points] After all, the fuel efficiency performance is satisfactory. My previous car was a light turbo car and was about 15 km/L, but the current Prius runs about 22 km on average. (The area of use is Osaka.) Also, the company is quiet because it runs slowly. I also own Elgrand, but it might be quieter than that... [Bad point] I feel that there is no room for high-speed driving. If you step on the accelerator, it will run as it is, but the fuel consumption will suddenly go down. Also, the engine noise becomes louder, so you don't want to step on the accelerator. [Comprehensive evaluation] Fuel efficiency is certainly good, but it feels relatively expensive compared to other models. It is recommended for those who want to be the same as those around them, but it does not give them the pleasure of running. [Good points] Fuel consumption is good. People around you think that you are conscious of the environment. It's quiet inside. [Bad point] While waiting for a signal, I encounter "the surroundings are full of Prius". It's too quiet for pedestrians in front to notice. [Comprehensive evaluation] I think that it is a car that anyone can easily accept regardless of age or gender! It is a car that is as convenient as a conventional sedan or hatchback in terms of usability. [Good points] Not only is it fuel efficient, but the appearance and interior are also calm.)

Method 3.2 - Using Text-Davinci-003 Model for Summarization

Example 1:

- `start_sequence = “要点を抽出します:”`
- Response:
  - “・燃費が良い ・落ち着いた外観・内装 ・ハイブリッドなので加速が物足り無い”
  - “・Good fuel efficiency・Relaxed appearance・Interior・Because it is a hybrid, acceleration is unsatisfactory”

Example 2:

- `start_sequence = “良い点、悪い点、総合評価を抽出します。”`
- Response:
  - “【良い点】燃費性能、落ち着いた外観、内装、利便性がある。【悪い点】高速走行時の加速性能に不満がある。【総合評価】プリウスは燃費性能が高く、落ち着いた外観、内装、利便性も兼ね備えたクルマです。高速走行時の加速性能に不満がありますが、全体的に満足できるクルマだと思います。”
  - “[Good points] Fuel efficiency, calm appearance, interior, and convenience.
    [Bad point] I am dissatisfied with the acceleration performance at high speed.
    [Comprehensive evaluation] The Prius is a car that has high fuel efficiency, a calm appearance, interior decoration, and convenience. Although I am dissatisfied with the acceleration performance at high speeds, I think that it is a car that I can be satisfied with overall.”

Method 3.2 - Adjusting Summary `start_sequence` for Better Results

- Initially, the `start_sequence` "良い点、悪い点、総合評価を抽出します。" provided satisfactory summaries for 3 concatenated reviews
- However, when tested with 18 concatenated reviews, the summaries were overly condensed
- To improve summary quality, we used the concatenation of “eval_comprehensive”, “eval_dislike”, and “eval_like” fields independently
- By adjusting the `start_sequence`, we achieved better summaries for each section, capturing important details

Method 3.2 - Adjusting Summary `start_sequence` for Better Results

- Car model: toyota prius
- Num fields concatenated: 18
  - Review field: “eval_comprehensive”
  - Num tokens: 2,902
  - Start_sequence: “良い点、悪い点、総合評価を抽出します。”
  - Summary: “【良い点】燃費が良い。ハイブリッドなので優秀です。静か。スタイリッシュなデザイン。【悪い点】少し値段が高い。車内空間が狭い。【総合評価】とても良い車です。静かで燃費が良く、デザインもスタイリッシュなので、とても満足しております。”
    [Good points] Fuel consumption is good. It's a hybrid, so it's great. quiet. Stylish design. [Bad point] The price is a little high. The space inside the car is small. [Comprehensive evaluation] It is a very good car. I am very satisfied because it is quiet, fuel efficient, and the design is stylish.

- Review field: “eval_like”
- Num tokens: 655
- Start_sequence: “車のこれまでの良い点をまとめる :”

- Review field: “eval_dislike”
- Num tokens: 683
- Start_sequence: “車のこれまでの悪い点をまとめる :”

- Review field: “eval_comprehensive”
- Num tokens: 1029
- Start_sequence: “これまでの車のレビューをまとめます :”

- Total tokens: 2367

eval_like_summary: “燃費、静粛性、低燃費、スタイリッシュなデザイン、乗り心地が良い。 ”
Good fuel efficiency, quietness, low fuel consumption, stylish design, **and comfortable ride**.

eval_dislike_summary: “・燃費が実際よりも低い ・広さが狭めな ・静粛性がない ・乗り心地が少々難しい ・値段が少し高め”
Lower fuel consumption・ Narrow space・**No quietness**・**Slightly difficult to ride**・The price is a little high

eval_comprehensive_summary: “燃費がとても良く、安定してて、高性能なエンジンで、見た目もスタイリッシュで、乗り心地も良く、エアバッグや ABSなどのセーフティー装備も充実しています。”
It has very good fuel economy, is stable, has a high-performance engine, looks stylish, has a comfortable ride, and **is equipped with safety equipment such as airbags and ABS.**

Method 3.2 - Pipeline

1. For each car model group, use K-means to find a group of K reviews that maximizes the number of tokens up to a limit of 3000 (using eval_comprehensive)
2. Once the reviews are selected, concatenate the fields "eval_comprehensive," "eval_dislike," and "eval_like" individually for each group
3. Summarize each group with respective start sequences:
    - "これまでの車のレビューをまとめます:"
    - "車のこれまでの悪い点をまとめる:"
    - "車のこれまでの良い点をまとめる:"
4. Concatenate the individual summaries to produce a text in the format: "【良い点】\n {eval_like_summary}\n【悪い点】\n{eval_dislike_summary}\n【総合評価】\n{eval_comprehensive_summary}"
5. Extract a car vector embedding for each summary using the sonoisa model

Method 3.2 - OpenAI's Summarization Cost

Total number of reviews summarized: 5,213

	Price ($)	Tokens prompt	Tokens generated	Total tokens	Average tokens / review	Average price/review ($)
Like text summaries	8.15	339,735	69,190	408925	78.44	0.0016
Dislike text summaries	6.43	268,873	61,323	330196	63.34	0.0013
Comprehensive text summaries	9.38	362,735	97,297	460032	88.25	0.0018
Totals	24.78	971,343	227,810	1199153	230.03	0.0047

Method 3.2 - OpenAI's Summarization Conclusion

- Summaries returned have varying formatting
  - Challenges in presenting consistent summaries for a user-friendly application
- High cost of summarization
  - More than $320 for the current dataset
- Promising results with subsets of reviews
  - If budget is not a constraint, Text-Davinci-003 model produces high-quality summaries
- Overall assessment
  - Text-Davinci-003 model is effective for summarizing car reviews, but considerations regarding cost and formatting consistency should be taken into account

Comparison of Methods: Effectiveness and Practicality

- Method 1: Mean Averaging of Review Car Vectors
  - Effectiveness: Low
    - Inaccurate results due to blurring of opinions
    - Fails to capture the essence of overall sentiment
  - Practicality: High
    - Straightforward and computationally inexpensive
    - Less complex method
- Method 2: K-means Clustering and SBERT Model Encoding
  - Effectiveness: Moderate
    - Addresses the token input limit issue
    - Better representation of car models based on user reviews
    - However, unpopular car models with few reviews may still be ranked high
  - Practicality: Moderate
    - More computationally intensive compared to mean averaging
    - Requires selecting representative reviews and applying K-means clustering

Comparison of Methods: Effectiveness and Practicality

- Method 3: OpenAI Summarization (Text-Davinci-003 Model)
  - Effectiveness: High
    - Produces high-quality summaries
    - Captures the essence of user reviews
  - Practicality: Low
    - High cost of summarization ($320+ for the dataset)
    - Inconsistent formatting of summaries may affect user experience

In conclusion, each method has its trade-offs in terms of effectiveness and practicality. Depending on the project's goals and constraints, a suitable method should be chosen to best balance these factors.

Conclusion and Future Work

- Uneven distribution of user reviews across car models
  - Unpopular car models with few reviews are not prioritized
  - Adjustments are required according to data characteristics
- Simple average approach not effective in this car review dataset
  - Difficult to capture diverse opinions
  - Passionate reviews might be overshadowed by generic ones
- Learning opportunities from failures and successes
  - Semantic search advantages remain valid
    - Allows for searching without defining synonyms (e.g., "kawaii")
    - Imperfections still present (e.g., “日本一周をしたい ” (I want to travel around Japan) might not yield desired results)
  - Providing users with new search opportunities
  - Offering manufacturers learning opportunities from user review data
- Future exploration and improvements
  - Generate evaluation data
  - Fine-tuning the semantic search model
  - Hyperparameter search (e.g., minimum number of reviews required per model)
  - Re-ranking strategies (e.g., prioritizing popular car models)

Demo

- car search demo

Q&A Session

- Thank you for your attention!
- We will now open the floor for questions
- Please feel free to ask any questions related to the presentation