CZECH TECHNICAL
UNIVERSITY
IN PRAGUE

F3

Faculty of Electrical Engineering
Department of Cybernetics

Bachelor's Thesis

Automated Generation of Textual Annotations for Bicycle Routes Using Large-Scale Language Models

Michaela Pejšová

May 2024
Supervisor: doc. Ing. Michal Jakob, Ph.D.

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Pejšová Michaela** Personal ID number: **507302**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Open Informatics**

Specialisation: **Artificial Intelligence and Computer Science**

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Automated Generation of Textual Annotations for Bicycle Routes Using Large-Scale Language Models

Bachelor’s thesis title in Czech:

Automatické generování textových anotací cyklistických tras pomocí velkých jazykových modelů

Guidelines:

A common feature of cycling routes presented in tourist guides is their textual description. The purpose of such descriptions is to provide users with important and useful information about the route and its surroundings, thus complementing the geographical presentation of the route shown on the map. The aim of this thesis is to investigate the possibility of generating such textual descriptions (hereafter referred to as 'route annotations') for a given route automatically, using large-scale language models. The input to such an annotator would be a cycling route defined as a path on the bicycle transport network. The output would be a textual description of the route that captures relevant information about the route.

1. Learn about generating textual annotations for transport routes.
2. Understand how language models work and analyze their potential applications in the route annotation problem.
3. Design a software system for route annotation using large-scale language models.
4. Implement the designed software system using a suitable language model and other software libraries.
5. Evaluate the implemented system on a well-chosen test set of real cycling routes. Assess both the technical attributes of the proposed approach and the subjective quality of the produced annotations.

Bibliography / sources:

[1] Boeing, G. (2017). OSMnx: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks. In Computers, Environment and Urban Systems, 65, pp. 126-139. doi:10.1016/j.compenvurbsys.2017.05.004

[2] Dräger, Markus & Koller, Alexander. (2012). Generation of landmark-based navigation instructions from open-source data. In Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 757-766.

[3] Aghzal, M., Plaku, E., & Yao, Z. (2023). Can Large Language Models be Good Path Planners? A Benchmark and Investigation on Spatial-temporal Reasoning. In ArXiv. /abs/2310.03249

[4] Fernandez, A., & Dube, S. (2023). Core Building Blocks: Next Gen Geo Spatial GPT Application. In ArXiv. /abs/2310.11029

Name and workplace of bachelor’s thesis supervisor:

doc. Ing. Michal Jakob, Ph.D. Artificial Intelligence Center FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **28.01.2024** Deadline for bachelor thesis submission: **24.05.2024**

Assignment valid until: **21.09.2025**

doc. Ing. Michal Jakob, Ph.D. Supervisor’s signature	prof. Dr. Ing. Jan Kybic Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce her thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt

Student's signature

Acknowledgement / Declaration

I would like to express my gratitude to my supervisor, doc. Ing. Michal Jakob, Ph.D., for providing me with an exciting thesis topic. His guidance, advice, and productive discussions held every week were essential for the success of this project.

I am also grateful to everyone who participated in the user testing and shared their opinion. The feedback from real users is priceless.

And finally, I would like to thank my family and my partner for their support.

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university thesis.

24 May 2024, Prague

....

Abstrakt / Abstract

Cyklistické průvodce obvykle poskytují detailní vylíčení cyklistických tras. S rychlým pokrokem v oblasti umělé inteligence, zejména velkých jazykových modelů, se otevírá možnost generovat tyto popisy automaticky pomocí AI.

Tato práce popisuje vývoj automatického anotátoru cyklistických výletů s využitím AI. Anotátor má za cíl generovat popisy výletů podobné těm napsaným lidmi, se zaměřením jak na technické aspekty trasy, tak na významné body zájmu. Kromě toho anotátor dokáže přizpůsobit popis preferencím konkrétního uživatele.

Provedeme rešerši existující literatury týkající se preferencí cyklistů a relevantních technologií využitelných pro vývoj anotátoru, s důrazem na velké jazykové modely. Na základě poznatků z rešerše navrhneme koncept anotátoru, poté anotátor implementujeme a poskytneme konkrétní implementační detaily.

Popisy vygenerované anotátorem jsou vyhodnoceny na existujících trasách, aby bylo možné objektivně posoudit přesnost a kvalitu výsledku. Provedeme také uživatelské testování za účelem získání zpětné vazby od cyklistů.

Výsledky evaluace naznačují, že nejmodernější velké jazykové modely jsou pro tvorbu popisů tras použitelné, pokud jsou mají k dispozici relevantní data. Jsou schopny napodobovat popisy psané lidmi, a navíc nabízejí výhody jako aktuálnost nebo možnost personalizace popisu.

Klíčová slova: cyklistika, automatický popis trasy, popis itineráře, velké jazykové modely, OpenStreetMap

Překlad titulu: Automatické generování textových anotací cyklistických tras pomocí velkých jazykových modelů

Cycling guidebooks usually provide rich and descriptive narratives of cycling routes. However, with the rapid advancements in AI, particularly large language models, there is a growing potential for AI to generate these descriptions automatically.

This thesis presents the development of an AI-powered automatic cycling trip annotator. The annotator aims to generate human-like trip descriptions, with an emphasis on both the technical aspects of the route and notable points of interest. Additionally, the annotator adjusts the description to the preferences of a particular user.

We research existing literature regarding cyclists' preferences and relevant technologies utilizable for annotator development, with a focus on large language models. Based on the research findings, we propose a concept, then implement the annotator and provide specific implementation details.

The annotator's performance is evaluated on existing routes to assess the accuracy and quality of the result objectively, and small-scale user testing is run to collect feedback from cyclists.

The evaluation results suggest that the state-of-the-art large language models are utilizable for such a task when equipped with relevant data. They are capable of mimicking human descriptions and offer the advantages of being up-to-date and providing personalization.

Keywords: cycling, automatic route annotation, itinerary description, Large-scale language models, OpenStreetMap

Contents /

Tables / Figures

Chapter 1
Introduction

1.1 Motivation

The cycling guidebooks usually focus on the technical aspects of the trips, including terrain, elevation, or surface quality, along with the most appealing elements of the trip. This typically involves highlighting historical landmarks, natural landscapes, scenic views, or local eating options. Such an approach enables individuals to imagine the journey, helping them determine its suitability based on their preferences.

On the other hand, the description obtained by an automatic path planner consists of plain instructions, coordinates, and other raw data that give the user no idea about the surroundings. Extensive effort is then required to look up the context in the map, and some experience is needed to analyze the information correctly. It may feel frustrating for people who struggle with time, have difficulty reading the map, or simply prefer to get information in an easily intelligible form.

1.2 Project aims

The main objective is to create an automatic trip itinerary annotator using artificial intelligence. We use online geographic data and large language models to transform the output of a path planner, in the form of a mere line in the map, into a detailed and user-friendly description of a cycling trip. The annotator provides the cyclist with realistic expectations by stating practical information about the trip and mentioning notable natural or cultural landmarks, mimicking classical tourist itineraries written by humans. The tool is aimed to spare the user a significant amount of time and effort by eliminating the need to inspect the route manually.

Chapter 2
Research

Cycling trip itinerary annotation is a relatively unexplored research area, with minimal directly related work. The closest researched topic is generating natural language navigation instructions, which does not cover all the information relevant to our purpose. Therefore, our research focus involves two additional key areas. Firstly, we identify what information should be included in the trip description to adequately inform and attract the user, and secondly, we investigate how to use large language models to generate a natural and factually correct description.

2.1 Existing work

Current research in itinerary description focuses on producing detailed natural language navigation instructions without the use of large language models. For instance, [1] generates real-time navigation instructions for drivers using OpenStreetMap data.

Similarly, the thesis [2] focuses on generating comprehensive natural language navigation instructions specifically for cyclists, also utilizing OpenStreetMap data. The author develops a metric to identify the most salient landmarks, which are then used as orientation points in the generated instructions.

An attempt to use a custom-trained neural network for the instruction generation task is presented in [3]. The authors developed a neural model that is able to generate human-like navigational instruction using OpenStreetMap data.

While this approach also describes the trip, it creates a sequence of navigational instructions rather than an attractive trip description in the form of an article. Also, the generation of navigational instructions primarily focuses on the prominence of the landmarks, meaning that they are not necessarily interesting or engaging to cyclists.

2.2 Cyclists' preferences

The trip description should attract the cyclist and provide motivation for absolving the trip. In this section, we investigate what factors contribute to positive user perception of a cycling trip and assess what trip attributes should be contained in a trip description. Since cycling enthusiasts span a wide spectrum, we focus on understanding the motivations and preferences of cycling tourists as defined by [4]. These riders are defined as casual cyclists seeking enjoyment and tourism experience rather than competition. They are likely to plan their cycling trips based on inspiration from guidebooks, cycling journals, and other resources.

2.2.1 Points of interest

According to the survey done in [4], the users perceive as fairly or very important the following factors: good scenery (93.6 %), refreshment stops (88.7 %), villages (76 %), countryside locations (70.6 %), pubs and restaurants (70.6 %) and visitor attractions

(61.5 %). On the contrary, article [5] suggests that the main motivation is the cycling site itself and that facilities connected to tourism, such as restaurants, might be perceived as unimportant.

2.2.2 Technical aspects

The survey from [4] also states several technical aspects of the route appreciated by the users: low traffic density (93.1 %), quiet roads (95.2 %), road marking (85.1 %) and few hills (65.7 %).

2.2.3 Conclusion: Trip description features

A good cycling trip description should provide trip characteristics that the user finds useful. The literature findings suggest that it should contain a description of the road type and information about location and elevation. Also, it should simplify orientation by stating important orientation points, such as towns and villages. Regarding the points of interest, most users are interested in landscape descriptions, natural spots, tourist attractions, and possibly also refreshment options.

2.3 Large language models

Large Language Models (LLMs) [6] are artificial intelligence models designed to understand and generate human-like text. They utilize massive neural networks with billions of parameters typically pre-trained on extensive language corpora, enabling them to grasp patterns and contextual nuances within language. As a result, they can perform various natural language processing tasks, including text completion, translation, or summarization. However, their capabilities, often emerging as a byproduct of scaling the models [7], cover many other domains and are subject to research.

This chapter provides an overview of research regarding the abilities of the language models, particularly OpenAI's GPT-4, that is relevant to the purpose of generating a cycling trip annotation.

2.3.1 OpenAI GPT-4

GPT-4 [8] is, as of May 2024, the most recent and advanced large language model developed by OpenAI. [9] examines GPT-4 on various tasks, including mathematical and coding skills, interaction with humans, and real-world understanding and orientation. They claim that GPT-4 attains a form of general intelligence, as it showcases a good level of common sense, reasoning, and creativity. Furthermore, its performance on most probed tasks is comparable to the human level.

2.3.2 Language models and spatial orientation

Article [10] analyzes the capability of various LLMs to spatial reasoning and orientation in a purely artificial environment. It suggests that even relatively simple orientation tasks within a square grid are not trivial for the models. In more complicated or non-orthogonal environments, the accuracy decreases by far. The GPT-4 significantly outperforms the other models, implying its ability to create an inner representation of the environment. However, its performance is poor compared to humans and far from reliable. Articles [11] and [12] find that LLMs are inapplicable in long-term navigation and optimal path planning in complex environments. The performance improves in environments the model has been trained on, which implies insufficient common sense and lack of robustness.

2.3.3 Language models and geographical knowledge

Article [13] examines several LLMs on geospatial knowledge, geospatial awareness and geospatial reasoning tasks. Finds that performance in such tasks generally increases with the size of the LLM. However, even though LLMs show potential in this field, further advancements are needed mainly for lack of precision.

The relative lack of research and poor results in this field are addressed by article [14], which introduces the concept of MapGPT, an abstract model combining the strengths of LLMs and geospatial analysis. Another proposal of a similar concept, the multimodal GeoAI, is proposed in vision paper [15]. These papers are proposal-only and do not include implementation details.

Article [16] aims to present a prototype of an AI-powered GIS (Geographical Information System). The system utilizes GPT-4 as a reasoning unit, which autonomously processes given data according to the user input, using its natural language understanding and code generation skills.

2.3.4 Conclusion: Utilizable technologies

Existing research indicates that even state-of-the-art LLMs do not possess precise geographical knowledge and exhibit limited spatial reasoning abilities. Consequently, if we want an LLM to annotate a trip, passing it only its coordinates is insufficient. We can only expect the model to generate a relevant description while providing further context. The context needs to be extracted separately and inputted into the model, as shown in article [16].

Chapter 3
Approach

This chapter provides the annotator proposal, a detailed step-by-step description of the principles of the annotation process, and the definitions of used components. The specific implementation details can be found in Chapter 4.

3.1 Annotator proposal

We aim to create an annotator able to generate trip descriptions fulfilling the following requirements, based on our research findings in Section 2.2:

- **Provide essential technical information about the route:**
  - distance
  - elevation
  - terrain
  - locality
  - orientation points
- **Provide information about the environment:**
  - route surroundings
  - points of interest along the way
- **Assure the quality of the description:**
  - provide vivid and engaging narration
  - give a positive impression of the trip
  - use appropriate language
  - mimic professional human-written descriptions

The research from Section 2.3 suggests that the language models do not possess geographic knowledge that is precise enough to generate such detailed and accurate trip descriptions. However, when the language models are provided with the data, they are able to utilize it and generate satisfactory results.

Therefore, the annotator first needs to obtain the geographic data. We use online resources for this task. Then, the geographic data must be processed so the language model can understand it. Finally, the data is inputted to a language model that is instructed to generate the trip description.

3.2 Definitions

Before describing the design of the annotation process itself, we have to define the terms and functions used throughout this chapter.

3.2.1 Map and coordinates

Definition 3.1. *Map* $M = (F, C)$ comprises of a set of features F within a space described by a coordinate system. The set of all coordinates, denoted C, is a subset of $\mathbb{R}^2$.

Definition 3.2. *Coordinate pair* $c \in C$ is an ordered pair $(latitude, longitude)$, defining the position of a single point. Also referred to as *coordinates* or *point*.

Definition 3.3. Function $dist$: $C \times C \to \mathbb{R}_0^+$ is the distance between two points.

Definition 3.4. Function $neighborhood$: $C \times \mathbb{R}^+ \to (N \subseteq C)$ assigns a coordinate pair c its circular neighborhood N, centered in c and with a radius of D, as a set of coordinate pairs.

$$neighborhood(p, D) = \{x \in C | dist(x, c) \leq D\}$$

3.2.2 Feature

Definition 3.5. *Feature* $f = (g, a)$, where $g \subseteq C$, $a \subseteq \mathbb{T}$, represents a real object in a map $M = (F, C)$, $f \in F$.

Definition 3.6. Function g: $F \to (C_f \subseteq C)$ assigns a feature its geometric representation. The features are of the following geometrical types:

- single point (defined by one coordinate pair)
- line or a polygonal chain (defined by an ordered list of coordinates $C_r = (c_1, ..., c_n)$, where $c_1 \neq c_n$)
- polygon (defined as an area enclosed by an ordered list of coordinates $C_r = (c_1, ..., c_n)$, where $c_1 = c_n$)

Definition 3.7. Function $neighborhood$: $F \times \mathbb{R}^+ \to C^n$ assigns a feature $f = \{c_1, ..., c_n\}$ its neighborhood within distance D as a set of coordinate pairs.

$$neighborhood(f, D) = \bigcup^{c \in f} neighborhood(c, D)$$

Definition 3.8. Function $centroid$: $F \to C$ assigns a feature f its centroid.

Definition 3.9. Function $saliency$: $F \to \mathbb{r}$ assigns a feature its saliency score.

Definition 3.10. *Tag* $t \in \mathbb{T}$, is an ordered pair $(key: value)$. The value can be of several types:

- single value:
  $(key:\ value)$
- set of values:
  $(key:\ \{value_1, ..., value_n\})$
- boolean value
  $(key:\ True/False$[1]$)$

[1] If the value is set to *True*, it means that any of the values for the particular key applies. This notation is, therefore, used for simplification, as we do not have to state every value. Specifying the tag value to *False* is redundant, as it is the default option when no tag with the particular key is present.

The set of all tags is denoted by $\mathbb{T}$. We use tags to assign attributes to features. The *key* usually stands for a category, and the *value* for a specification.

Definition 3.11. Function a: $F \to (T \subseteq \mathbb{T})$ assigns a feature its attributes represented by a set of tags.

Definition 3.12. Function $filter$: $F \times T \to (F' \subseteq F)$ filters the features based on the provided tag set.

3.2.3 Path

Definition 3.13. *Transportation network graph* $G = (J, W)$ is a subset of transportation-related map features - junctions (single-point shaped features) J and ways W.

Definition 3.14. *Way* $w \in W$ is a special type of polygonal chain shaped feature defining one edge in the transportation network graph.

Definition 3.15. Function len: $W \to \mathbb{R}$ assigns a way its length.

Definition 3.16. Function e_s: $W \to \mathbb{R}$ assigns a way its start elevation.

Definition 3.17. Function e_e: $W \to \mathbb{R}$ assigns a way its end elevation.

Definition 3.18. Function r: $W \to R$ assigns a way a road type. R is the set of all road types.

Definition 3.19. *Path* within a road network graph $G = (J, W)$ is an ordered list of ways $P = (w_1, w_2, ..., w_n \in W)$ such that for every pair (w_i, w_{i+1}), there exists a common node. In our specific case, we consider the path directed, starting in w_1 and ending in w_n.

3.2.4 Segment

Definition 3.20. *Segment* s is a subsequence $(w_i, ..., w_j)$ of a path $P = (w_1, w_n)$. We denote $\mathbb{S}$ a set of all possible subsequences of path P. [2]

Definition 3.21. Function *segmentation*: $P \times \mathbb{N} \to \mathbb{S}^n$ creates a sequence of segments $S = (s_1, ..., s_n)$, such that S contains $n \in \mathbb{N}$ segments and $S \approx P$, meaning that S contains the same sequence of ways as P.

Definition 3.22. Function *size*: $\mathbb{S} \to \mathbb{N}$ returns the number of ways of segment s.

$$size(s) = j - i$$

Definition 3.23. Function len: $\mathbb{S} \to \mathbb{R}$ assigns a segment $s = (w_i, ..., w_j)$ combined length of its ways:

$$len(s) = \sum_{k=i}^{j} len(w_k)$$

Definition 3.24. Function r: $\mathbb{S} \to (\subseteq R)$ assigns a segment s a set of road types of its ways:

$$r(s) = \{r(w) | w \in s\}$$

[2] There are only a few differences between a way and a segment. They are of the same shape - a polygonal chain. The main difference is in their length and origin - a way is the minimal building block, creating an edge in the road network graph, whereas a segment is created by one or more paths.

Definition 3.25. Function $e_s\colon \mathbb{S} \to \mathbb{R}$ assigns a segment its start elevation according to its first way.

$$e_s(s) = e_s(w_i)$$

Definition 3.26. Function $e_e\colon \mathbb{S} \to \mathbb{R}$ assigns a segment its end elevation according to its last way.

$$e_e(s) = e_e(w_j)$$

Definition 3.27. Function *merge*: $\mathbb{S} \times \mathbb{S} \to \mathbb{S}$ merges two consecutive segments $s_1 = \{w_{1_1}, ..., w_{1_n}\}$ and $s_2 = \{w_{2_1}, ..., w_{2_n}\}$, creating a new segment

$$s_{1,2} = \{w_{1_1}, ..., w_{1_n}, w_{2_2}, ..., w_{2_n}\}$$

Definition 3.28. Function *neighborhood*: $\mathbb{S} \times \mathbb{R}^+ \to C^n$ assigns a segment $s = \{w_1, ..., w_n\}$ its neighborhood within distance D as a set of coordinate pairs.

$$neighborhood(s, D) = \bigcup^{w \in s} neighborhood(s, D)$$

Definition 3.29. Function *direction*: $S \to \{north, northeast, ..., northwest\}$ assigns a segment its direction.

3.2.5 Textual objects

Definition 3.30. *String str* is an ordered set of characters $(c_1, ..., c_n) \in \mathbb{A}$, where $\mathbb{A}$ is an alphabet. We consider $\mathbb{A}$ to be the standard Unicode alphabet.

Definition 3.31. *User preference* is a string, denoted U for recognition.

Definition 3.32. Function *parse*: $U \to (T \subseteq \mathbb{T})$ returns a set of tags with respect to the given string.

Definition 3.33. Function *introduction*: $P \to \mathbb{A}^n$ describes the path P with a string. We denote the resulting string as A_I.

Definition 3.34. Function $annotate_n$: $N \to \mathbb{A}^n$ describes a $N = neighborhood(s)$, $s \in \mathbb{S}$ with a string. We denote the resulting neighborhood description as A_N.

Definition 3.35. Function $annotate_s$: $\mathbb{S} \times A_N \times F_s \to \mathbb{A}^n$ describes a segment $s \in \mathbb{S}$ with a string, using the neighborhood description A_N and a set of features F_s. We denote the resulting neighborhood description as A_s.

Definition 3.36. Function $annotate_f$: $A_I \times A_s^n \times U \to \mathbb{A}^k$ creates the final annotation using the introduction, the segment annotations, and the user preference strings.

3.3 Overall annotation approach

The annotation process can be split into the following steps.

1. **Input** - consisting of the trip path and user preference string
2. **User preference parsing** - transform user preference string from natural language to tags
3. **Path processing** - compute necessary statistics for the path
4. **Segmentation** - divide the path into logical segments
5. **Segment processing** - compute necessary statistics for the segments
6. **Segment neighborhood** - get neighborhood of each segment
7. **Segment neighborhood description** - get description of land use in the neighborhood
8. **Feature extraction** - extract features around each segment, either default or based on user preference
9. **Feature salience** - compute salience score for features and filter them
10. **Segment annotation** - annotate each segment in natural language using information about the particular segment and significant features around
11. **Introduction generation** - annotate overall trip statistics
12. **Final annotation** - generate final annotation based on introduction, segment description, and user preference

The connections between the steps and dependencies on external sources are illustrated in Figure 3.1. Each step is further described in its respective subsection in Section 3.4, and specific implementation details of each step can be found in Section 4.4.

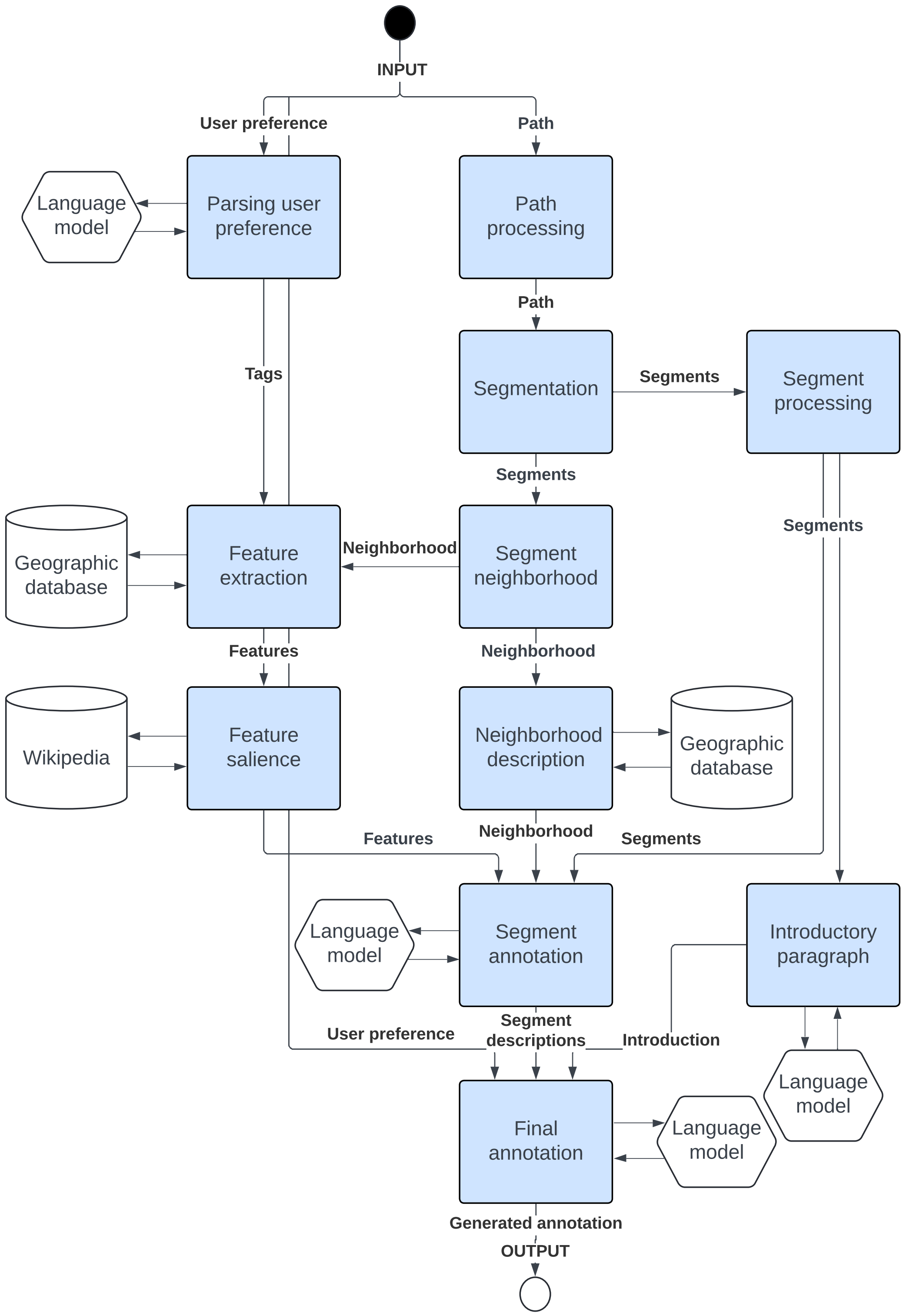

Figure 3.1. Illustration of the complete annotation process

3.4 Process details

In this section, we provide a detailed approach to particular annotation steps.

3.4.1 Input

The annotator input is a path P delineating the journey, consisting of a sequence of ways, and a user preference U expressed in natural language as a string. The user preference should describe the interests of a specific user and is used to personalize the trip annotation.

3.4.2 User preference parsing

The input user preference string is a natural language sentence or a sequence of keywords, but might be also empty. For instance, a user might express their interests by stating `"I am a railway fan"` or list keywords as `"train, railway"`.

A language model is instructed to parse the user preference - generate relevant tags to enable the annotator to identify points of interest aligned with the user's interests. The resulting tag set for the example user preference could be the following:

```
{
    'railway': True,
    'tourism': 'railway'
    'historic': 'railway'
    'amenity': ['train_station', 'railway_station'],
    'leisure': 'track',
    'landuse': 'railway'
}
```

These tags will be further used to customize map feature retrieval.

3.4.3 Path processing

To be able to describe the terrain accurately, more detailed statistics regarding the elevation and road characteristics are needed. Let $P = (w_1, ..., w_n)$ be the input path. We compute or estimate additional information for each $w \in P$:

- elevation grade computed as $\frac{e_e(w)-e_s(w)}{len(w)}$
- elevation grade description as an enumeration of elevation grade
- elevation extremes
  - global maximum:

$$max_{global}(w) = \begin{cases} True & \text{if } w \in argmax_{w \in P} e_s(w), \\ False & \text{otherwise.} \end{cases}$$

  - global minimum:

$$min_{global}(w) = \begin{cases} True & \text{if } w \in argmin_{w \in P} e_s(w), \\ False & \text{otherwise.} \end{cases}$$

  - local maxima:

$$max_{local}(w) = \begin{cases} True & \text{if } (e_s(w_{i-1}) - e_s(w_i) > t) \wedge (e_s(w_{i+1}) - e_s(w_i) > t), \\ False & \text{otherwise.} \end{cases}$$

  - local minima:

$$min_{local}(w) = \begin{cases} True & \text{if } (e_s(w_i) - e_s(w_{i-1}) > t) \wedge (e_s(w_i) - e_s(w_{i+1}) > t), \\ False & \text{otherwise.} \end{cases}$$

  $t \in R^+$ ... threshold

Based on road type $r(w)$, we further estimate the following:

- road width
- surface
- traffic density

3.4.4 Segmentation

Since the path usually consists of many very short ways unsuitable for description, we group them into segments based on elevation grade similarity and way length. We do it in several iterations, gradually increasing tolerance for segment merging. The size of the resulting segmentation S should not exceed the maximum of N_{max} segments.

```
Pseudocode:
Input:  path

S = path
While size(S) > N_max:
    new_S = []
    last = None

    adjust thresholds

    For each (segment in S):
        If (last is None):
            last := segment
        Else if (threshold conditions satisfied):
                last := merge(last, segment)
        Else:
            append last to new_S
            last := segment
    S := new_S
return S
```

We mark the resulting list of segments as S. S covers the same route as path P, but is more conveniently segmented. The segmentation S also approximates the original elevation profile of the path P, facilitating its description.

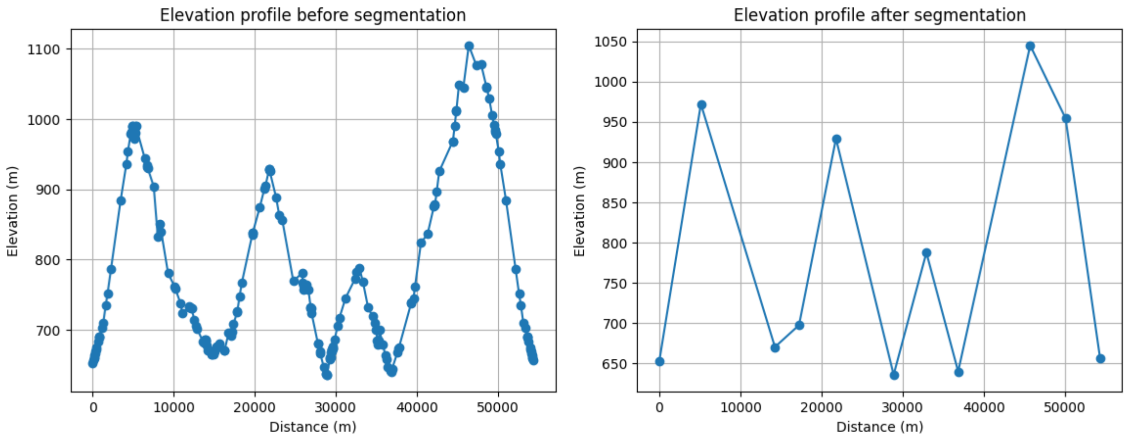

Figure 3.2. Elevation profile before and after segmentation ($N_{max} = 10$)

3.4.5 Segment processing

To simplify orientation within the route, we enrich each segment with further important information. The following are added:

- cumulative length to express the progression of the trip
- direction

For each segment $s = (c_1, ..., c_n)$, we compute its cardinal or intercardinal direction by evaluating the segment azimuth.

The azimuth α_{deg} is the clockwise direction relative to true north [17]. It can be computed using the following steps:

1. Get latitude and longitude from $c_1 = (lat_{1_{deg}}, lon_{1_{deg}})$ and $c_n = (lat_{n_{deg}}, lon_{n_{deg}})$.
2. Convert the latitudes and longitudes to radians.

$$lat_1 = \frac{\pi}{180} \cdot lat_{1_{deg}}$$

$$lat_2 = \frac{\pi}{180} \cdot lat_{n_{deg}}$$

$$lon_1 = \frac{\pi}{180} \cdot lon_{_{deg}}$$

$$lon_2 = \frac{\pi}{180} \cdot lon_{n_{deg}}$$

3. Compute the longitude difference (angular distance) and convert it to radians.

$$\Delta\lambda = lon_2 - lon_1$$

4. Compute the azimuth in radians α_{rad}.

$$\alpha_{rad} = arctan2\left(\sin(\Delta\lambda) \cdot \cos(lat_2), \cos(lat_1) \cdot \tan(lat_2) - \sin(lat_1) \cdot \cos(\Delta\lambda)\right)$$

5. Convert the azimuth to degrees.

$$\alpha_{deg} = \alpha_{rad} \cdot \frac{180}{\pi}$$

Based on the azimuth, we assign segment s the value on i^{th} position of

$$(north, northeast, east, southeast, south, southwest, west, northwest),$$

where $i = round(\alpha/45) mod\ 8$.

3.4.6 Segment neighborhood

For each segment $s \in S$, $s = (c_1, ..., c_n)$, we compute its neighborhood within the distance of D_{max} as $neighborhood(s, D_{max})$.

The neighborhood approximates the area directly observable by the cyclist, or nearby. Within the neighborhood, we fetch potentially interesting trip features and also describe its environment.

3.4.7 Segment neighborhood description

We retrieve information about the land use features within the neighborhood boundaries from the geographic data source. Each land use feature f is assigned either left or right direction, based on the cyclist's perspective:

Let x, y be latitude and longitude of $centroid(f)$, $(lat_1, lon_1) = c_1$ and $(lat_n, lon_n) = c_n$.

The direction can be obtained by evaluating the following cross-product:

$$R_s(x, y) = (lon_n - lon_1) \cdot (y - lat_1) - (x - lon_1) \cdot (lat_n - lat_1);$$

$$direction(x, y) = \begin{cases} left & R_s(x, y) > 0, \\ right & R_s(x, y) < 0, \\ none & \text{otherwise.} \end{cases}$$

We separately analyze the distribution of land use types on both sides by calculating their respective percentages. If the specific land use type rate exceeds a pre-defined threshold, it is contained in the neighborhood description. Finally, we automatically generate descriptions for both sides and merge them into one complete natural language description of the neighborhood. A rule-based description generation is used for this task.

3.4.8 Feature extraction

To describe the points of interest along the journey, we must fetch them from our geographic data source. For the $neighborhood(s_i)$ of each segment $s_i \in S$, we extract a set of features F_i such that

$$\forall f \in F_i \colon n(s_i) \cap g(f) \neq \emptyset$$

. Additionally, we filter F_i to ensure relevance. The default filtering tag set T_d contains tags regarding the following attributes:

- natural features
- historic features
- leisure-related features
- tourist attractions
- locations (cities, towns, villages, etc.)

We merge this default tag set with the customized user preference tag T_u set and retrieve features based on their union:

$$T = T_d \cup T_u$$

After we have fetched the features, we compute additional statistics, such as distance from the segment or their area. Finally, we determine the feature order in which they appear along the segment.

3.4.9 Feature salience

Since there might be an unnecessarily high amount of features around a particular segment, we need to assess their salience to be able to filter them. We compute the salience score of a feature based on multiple criteria:

- **Name:** Named features inherently possess greater significance than their unnamed counterparts.
- **Area:** Larger features naturally draw more attention and are thus considered more significant.
- **Distance:** Closer features are generally more noticeable and should receive higher salience scores.
- **Information available:** Features with more associated information are deemed more salient due to their depth of content, which implies their importance.
- **Wikipedia language coverage:** Features represented across multiple languages on Wikipedia are seen as more globally relevant and thus more salient.
- **User preference tags alignment:** Features that match user preferences are given greater salience, as they are more likely to meet user needs effectively.

Features around each segment $s \in S$ are sorted based on the salience score, and their count is limited to a maximum of n_{max_s}.

The value of n_{max_s} is set to

$$n_{max_s} = \left\lceil \frac{L(s)}{1000} \right\rceil$$

to ensure that the maximum number of selected features corresponds to the segment length in kilometers and therefore the coverage is consistent throughout the trip. We will address the chosen features as *points of interest*.

3.4.10 Segment annotation

It is crucial to preprocess the available information about segments to avoid overwhelming the language model with a significant amount of poorly structured information when creating the final annotation.

Each segment is annotated separately using a language model. The language model is instructed to create a dense description of the given segment. Its input consists of:

- segment attributes (length, direction, elevation statistics, etc.)
- segment neighborhood description
- points of interest around with any available information listed

The output is a string, which is a natural language paragraph accurately summarizing important segment information. The most essential features that the segment descriptions should contain are:

- information about terrain and elevation
- significant landmarks or points of interest

3.4.11 Introduction generation

We generate a brief paragraph about the trip as a whole, which is used for the introduction. It is generated by a language model instructed to create an introductory paragraph, provided with the following input:

- overall trip statistics (length, total elevation gain, etc.)
- location of the start and the end of the trip

The output is a natural language paragraph accurately describing the trip attributes, such as location, length, and elevation.

3.4.12 Final annotation

The annotation process results in an article consisting of a header, an introductory paragraph, and a detailed trip description split into multiple paragraphs. It is supposed to mimic a human-written trip description. Such an article is generated using a language model given:

- prompt with precise instructions regarding:
  - paragraph composition
  - input format description
  - language
- user preference
- introductory paragraph
- sequence of segment descriptions

The language model processes the input data according to the instructions in the prompt and generates the article - the final result of the annotator.

Chapter 4
Implementation

In this chapter, we will describe how the proposed annotator is implemented. We show the overall architecture and describe the components, used technologies, data sources, and specific implementation details.

4.1 Architecture

The annotator is written in Python 3 and runs locally. For some tasks, it uses remote services via API calls. There are several libraries wrapping up the API calls. Also, the annotator does not use locally stored data; all the data is fetched from remote sources.

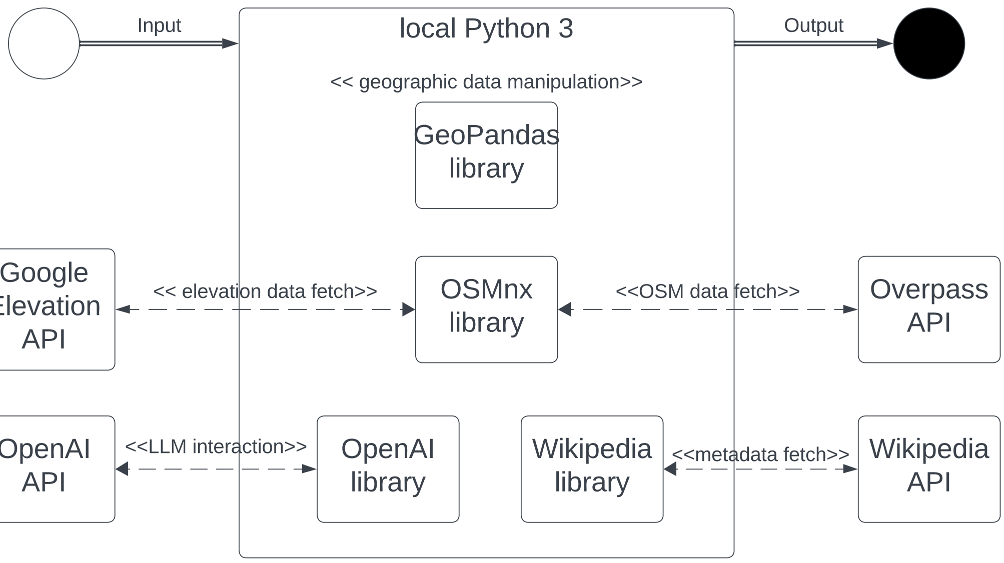

Figure 4.1. Implementation architecture.

4.2 Technologies

This section provides an overview of important libraries that are used with Python.

- **OSMnx library**

  The OSMnx [18] library simplifies the acquisition and manipulation of OpenStreetMap data (more about OpenStreetMap in Section 4.2) by providing users with direct access to the Overpass API, which operates on OpenStreetMap data and allows users to request specific geographical data based on particular locations or tags. Additionally, it incorporates elevation data using the Google Maps Elevation API.

We use OSMnx to obtain elevation data and to fetch relevant features along the path, which is the vital information base for the annotator.

- **GeoPandas library**

  GeoPandas [19] is an open-source library that extends Pandas, a widely used statistical library, to work with geospatial data. The library introduces new data structures, GeoSeries and GeoDataFrame, derived from the original Pandas Series and DataFrame, adjusted to work with geospatial data. It also provides functions for spatial operations, analysis, and visualization. GeoPandas collaborates well with OSMnx, as OSMnx gathers the data, returning them in a familiar GeoDataFrame format, and GeoPandas can perform further tasks on them.

  GeoPandas is used for geographic data manupulation and computations such as feature neighborhood or distance between two objects.

- **OpenAI library**

  The OpenAI library interacts with the OpenAI API[1], offering numerous pre-trained large language models of various capabilities and ranging prices. Each API request consists of a model configuration, prompt - the initial text or set of instructions given to the model to guide its generation, and optional parameters. The functions providing API calls accept the prompt in the form of a text string. Tokenization, encoding, and other tasks necessary for processing natural language through LLMs do not have to be dealt with, as the servers take care of them automatically. The server response is parsed into a structure containing the model's output.

  The OpenAI library instructs remote large language models to generate natural language descriptions of raw data, parse user preference, or generate the final annotation, and provides the generated results.

- **Wikipedia-API library**

  Wikipedia-API library facilitates accessing data from the Wikipedia API. Articles, summaries, links, and other metadata can be queried using this library.

4.3 Data sources

The size of worldwide geographic data is immense; therefore, we do not store it. The data used for annotation is fetched from different remote sources:

- **OpenStreetMap**

  The geographic data is sourced from the OpenStreetMap (OSM) [20] project, an open repository providing rich and diverse information about real-world features. Its dataset comprises detailed and regularly updated geographical information, including roads, landmarks, transportation networks, administrative boundaries, cultural sites, natural features, and other geographic elements.

  OSM uses a rich tagging system to describe the features. More details about the tags can be found on the OSM wiki [21].

  In this project, OSM data is fetched and used to describe the segment neighborhood and points of interest.

- **Google Elevation**

  The Google Elevation API is a service by the Google Maps Platform that provides elevation data for any point on the Earth based on geographic coordinates.

[1] https://openai.com/index/openai-api/

Since OSM does not provide elevation information, we use Google Elevation API to cover this knowledge gap.

- **Wikipedia**

  Wikipedia is a free online encyclopedia.

  The information from Wikipedia, mainly the article metadata, is used to assess the importance of selected map features.

4.4 Implementation details

Chapter 3 described the purpose of each process that our annotator uses. In this section, we provide specific implementation details.

4.4.1 Input

The input consists of a link to a file containing the path of the trip and a non-obligatory user preference in the form of a string.

The input file should preferably be in the .geojson format, as it is the most precise format. A .gpx file can be used as well. However, since this format provides a path approximation only, possibly leading to worse output accuracy.

The file content is loaded to a *GeoDataFrame* named `path`. There are multiple columns based on shape, location, and OSM tags:

- `geometry`: linestring defined by a list of GPS coordinates localizing the path
- `length`: in meters
- `highway`: information about road type (e. g. `highway: motorway`)
- and possibly other columns that are not necessarily present, such as `name, max speed,` etc.

4.4.2 User preference parsing

The user preference is parsed using a language model.

This task requires the capability to follow instructions and OSM tag knowledge. The model has to understand basic real-word associations but does not have to be highly creative. Also, the context window size might also be minimal, as the instruction length combined with the user preference input length will not exceed hundreds of tokens.

Our setup follows:

- Model: GPT 3.5
- Temperature: 0.4
- Input: raw user preference
- Prompt:

  You are creating a JSON with OpenStreetMap tags for fetching relevant map data. You will get unstructured user preference, in a form of a sentence or keywords. Based on that preference, create a JSON with up to 10 OSM tags that cover the user's desire. The JSON values should contain either boolean, list of strings, or strings, like this one:

```
{"amenity": true,
 "amenity": ["shop", "parking"],
 "amenity": "pub"}.
```

  Return just the JSON.

  Example 1: User: 'I do not want to carry food, I want to get some on the way.' You:

```
{"amenity": ["restaurant", "fast_food", "cafe", "bar"],
  "shop": ["bakery", "convenience", "supermarket", "grocery"],
  "cuisine": true}
```

Example 2: User: I am a railway fan. You:

```
{"railway": true,
 "tourism": "railway",
 "historic": "railway",
 "amenity": ["train_station", "railway_station"],
 "leisure": "track",
 "landuse": "railway"}
```

- Output: JSON-formatted tags covering user preference

4.4.3 Path processing

The `path` is enriched with elevation data using Google Elevation API, and based on that, elevation extremes are computed. The following columns are added:

- `start_elevation`: elevation of the start of the segment
- `end_elevation`: elevation of the start of the segment
- `grade`: average grade of the path
- `extremes`: containing a set possibly containing values from
  `[lowest_point, highest_peak, local_low_point, local_hilltop]`
- `elevation_description`: `grade` described in natural language, based on:

grade	elevation_description
grade < -0.1	Very Steep Downhill
-0.1 to -0.06	Steep Downhill
-0.06 to -0.025	Moderate Downhill
-0.025 to -0.01	Slight Downhill
-0.01 to 0.01	Flat
0.01 to 0.025	Slight Uphill
0.025 to 0.06	Moderate Uphill
0.06 to 0.1	Steep Uphill
grade >= 0.1	Very Steep Uphill

Table 4.1. Grade descriptions in natural language.

Based on `highway` values, we add other columns:

- `width`
- `surface`
- `traffic_density`

estimating road characteristics, based on the table below:

highway	width	surface	traffic_density
motorway	wide	paved	high
trunk	wide	paved	high
primary	wide	paved	high
secondary	narrow	paved	medium
tertiary	narrow	paved	medium
residential	narrow	paved	low
service	very narrow	paved	low
living	very narrow	paved	low
pedestrian	very narrow	paved	very low
track	very narrow	unpaved	very low
unclassified	narrow	unpaved	low
cycleway	very narrow	paved	very low
footway	very narrow	paved	very low
bridleway	very narrow	paved	very low
steps	very narrow	paved	very low
path	very narrow	unpaved	very low

Table 4.2. Road characteristics estimation.

4.4.4 Segmentation

We use the algorithm presented in section 3.4.4, with the threshold conditions for merging the segments as follows:

```
if (grade(segment) * grade(last) > grade_threshold ||
len(segment) * abs(grade(segment)) < min_length_threshold) &&
len(last) + len(segment) < max_length_threshold:
```

In other words, these segments have to fulfill the following conditions to be merged:

- The product of their grades has to be higher than the threshold. The threshold is initialized to 0 and gradually decreases, meaning that the segments with the same grade sign can be merged from the beginning, and the tolerance for their difference increases over time.

  OR
- The product of the segment length and the absolute grade has to be lower than the threshold. This means the segments with negligible grade or length will merge sooner. The threshold and, therefore, the tolerance increase over time.

  AND
- The resulting segment length after potential merging has to be lower than the threshold, which increases over time. This condition has to be fulfilled every time.

The values are initialized to:

```
n_max = len(path)/5000
min_length_threshold = 0
```

```
    max_length_threshold = len(path)/10
    grade_threshold = 0
```

and thresholds are being adjusted in each iteration:

```
    min_length_threshold += 0.5
    max_length_threshold += 5
    grade_threshold -= 0.0001
```

These specific values were chosen based on several experiments.

The final product, the `segments` *GeoDataFrame*, covers the same route as path, contains the same columns, and has fewer rows. Each row symbolizes a segment within the route.

4.4.5 Segment processing

We compute the azimuth for each segment in `segments` and obtain the segment direction by evaluating it. The direction is present in the `direction` column.

We also add the cumulative sum to the `cumulative` column, computed using pandas' function `cumsum`.

4.4.6 Segment neighborhood

We compute the segment neighborhood using the GeoPandas `buffer` function. The perimeter, `max_distance`, is set to 200 meters. The neighborhood boundary is approximated by a polygon. An example visualization of a segment and its neighborhood can be found in Figure 4.2 on page 24.

4.4.7 Segment neighborhood description

Information about the land use within the neighborhood of each segment is fetched from OSM using `features_from_polygon`, an OSMnx function. The tags for feature fetching are set to

```
{
    'landuse' = True
}
```

and the land use is fetched and further processed for each segment neighborhood separately. The neighborhood description process of one segment is as follows:

1) split the land use features according to their side, left or right
2) group them based on their land use type
3) compute their respective percentages
4) discard types of land use with a percentage lower than 20 %
5) generate a natural language neighborhood description for each side
6) merge the descriptions

The resulting descriptions are inserted into the `landuse_description` column in the `segments` GeoDataFrame.

4.4.8 Feature fetching

The features are fetched all at once. Therefore, we first compute the union of all segment neighborhoods, `whole_neighborhood`, using the `unary_union` function by GeoPandas.

Then, we prepare the tags for feature fetching. The following universal tags are used every time.

```
{
    'leisure': True,
    'tourism': 'attraction',
    'natural': True,
    'historic': True,
    'place': True
}
```

If user preference tags are present, they are merged with the universal ones, forming the final `tags` dictionary.

The features are fetched from OSM using the `features_from_polygon` OSMnx function. The resulting *GeoDataFrame* named `features` contains arbitrary columns based on OSM tags present in the features. Each feature is symbolized by one row.

We add the following columns to `features`:

- `closest_segment`
- `distance` - distance from closest segment
- `distance_from_start` - distance from the start of the closest segment

using the function `distance` from GeoPandas. And finally, we sort the features based on the distance from the start.

Figure 4.2 on page 24 illustrates an example feature setup around a segment.

4.4.9 Feature score

Since there are usually many features around each segment, we need to filter them. We do so based on the salience score. The score is initially set to 0 and is increased by adding points according to the feature characteristics:

- $+= 3$ if the feature is named
- $+= 1$ for each tag that matches the universal tags
- $+= 3$ for each tag that matches the user preference tags
- $+= 1$ for each language in which the feature has a Wikipedia page

We also add non-discrete values:

- $+= \frac{\texttt{non-empty\_columns}}{\texttt{total\_columns}}$
- $+= 2 \cdot \frac{\texttt{area}}{\texttt{max\_area}}$, where `max_area` is the area of the largest feature within the segment
- $+= 1 - \frac{\texttt{distance}}{\texttt{max\_distance}}$, where `max_distance` is the neighborhood perimeter

For each segment, we compute the n_max as `ceil(segment.length)` We filter the features to obtain the maximum of `n_max` features with the highest salience score for each segment while keeping the original order of the features. The following Figure 4.2 illustrates one chosen feature within a short segment.

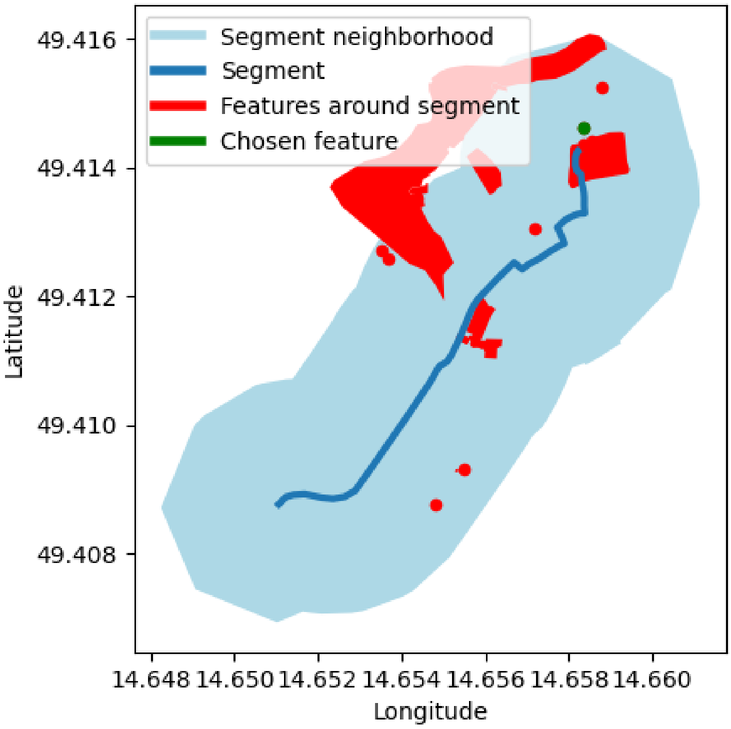

Figure 4.2. Situation around a segment. `n_max` = 1.

4.4.10 Segment annotation

To prepare data for a description of a particular segment, we:

1) Transform the respective row within the `segments` GeoDataFrame to a column: value dictionary, skipping all empty columns and columns that are of no use to the user (`geometry`, `osmid`, `closest_segment` and others).
2) Parse the row dictionary to string `segment`
3) Append the neighborhood description to the `segment`
4) Append information about each of the points of interest close to the segment:
   (i) If score > 10, mark it as `MUST MENTION`
   (ii) Transform the respective row within the `features` GeoDataFrame the same way as in 1)
5) If half of the trip was just reached, we append `Half of the trip reached` to the segment description

An example of string `segment`:

Lanes: 1, Maxspeed: 50, Oneway: False, Reversed: True, Length: 2755.561, Grade_abs: 0.108, Bridge: yes, Width: ['narrow'], Grade_classification: Moderate Uphill, Surface: ['paved'], Traffic_density: ['low', 'medium'], Start_elevation: 404.966, End_elevation: 522.083, Elevation_difference: 117.11699999999996, Elevation_grade: 0.10769314059877644, Extremes: ['local_low_point', 'local_hilltop', None], Direction: south, Cumulative_length: 3696.464, Segment: 1 Residential on your left side and a forest on your right side.

Features around:

Name: Horky, Html_name: Horky, Name:cs: Horky, Place: suburb, Population: 720, Area: 0.0, Distance: 91.2940

Name: Větrovy, Html_name: Větrovy, Name:cs: Větrovy, Place: suburb, Population: 269, Area: 0.0, Distance: 16.25961037302737

Name: Pramen svaté Eleanory, Natural: spring, Alt_name: Eleanořin pramen, Drinking_water: yes, Area: 0.0, Distance: 67.14786695280118

We annotate the string `segment` with a language model, prompted to generate a description of the segments, with the following rule: If a feature is marked as MUST MENTION, it should be wrapped in asterisks. It is done to preserve the importance of these features for the higher-level annotation.

The setup is as follows:

- Model: GPT 3.5
- Temperature: 0.4
- Input: `segment`
- Prompt:

> You are a cycling trip annotator generating a part of a cycling trip description. Generate a dense paragraph as a part of a cycling trip.
>
> You will get an automatically generated description of a random trip segment, and your goal is to annotate it based on given information. There are both information about length, elevation and terrain, and points of interest along the way. Mention the numbers, you should round them. The cyclist is not interested in precise meters.
>
> Sum up all available information, do not make anything up. Do not change the order of features mentioned. You should at least mention all points of interest. If they are marked as MUST MENTION, you should put their name into ∗∗'s (example: MUST MENTION: Praha ->
>
> ∗
>
> Praha∗) and include all available info about them. Do not include any headers, etc, just the summarized information.
>
> Important! You are given all the lengths in meters. If a distance is shorter than 1000 m, do not include the number and call it short. Recompute longer distances to kilometers. The elevation is in meters, round the elevation to tens or hundreds of meters.
>
> Small example of output with names replaced with dashes:
>
> The previously flat road transfers to a nice, 5 km long descent, leading you to the village of _. If you are lovers of good desserts and coffee, don't forget to stop at the _ café, which is just before the church. We sway gently up and down for a while and then turn right and climb a nice hill with turns to _. We ride through the village of _ and surprisingly start climbing again. This time, however, only for a while, after about four kilometers, we are going downhill again and descend to _. The surface here is not the best, so be careful on the descent.
>
> Do not replace the names with dashes in the final annotation. If you do not have the names, do not make them up.

The output example:

> The road is a single lane with a maximum speed of 50 km/h, not one-way, but reversed. It is 2.8 km long with a moderate uphill grade. The road is paved and narrow, with low to medium traffic density. Starting at an elevation of 405 meters, it ends at 522 meters, with an elevation difference of 117 meters. The route goes south and has a cumulative length of 3.7 km. Along the way, there is a residential area on the left and a forest on the right.
>
> As for points of interest, there is a suburb named ∗∗Horky∗∗ with a population of 720, located 91 meters away. Another suburb, ∗∗Větrovy∗∗ with a population of 269, is 16 meters from the route. Additionally, there is the ∗∗Pramen svaté Eleanory∗∗ (Eleanořin pramen) natural spring with drinking water, situated 67 meters from the path.

4.4.11 Introduction generation

We need to obtain the first paragraph to complete the information base for the final annotation.[2] This paragraph should describe the trip overall, including information about:

[2] Since this process is very similar to the segment annotation, we often do not name it separately and consider it to be the $n + 1^{st}$ segment annotation.

- total trip length (= sum of segment lengths)
- total ascent (= sum of positive elevation gains)
- total descent (= sum of negative elevation gains)
- start location
- end location

The start and end locations are obtained in a richer context than the standard segment location - we also fetch the district, region, state, etc. This is done by an OSMnx function `geocode_to_gdf`, using the unique `osmid` of the first and the last way of the trip.

We use the same setup of the language model as for the segment description.

- Model: GPT 3.5
- Temperature: 0.4
- Input: introductory information (stated above)
- Prompt:

> You are a cycling trip annotator helper generating the first paragraph of a trip description. You will get an automatically generated description of the trip features.
>
> Start with a relevant name. If the trip is circular, mention it. Else improvise based on the data. Mention the length, difficulty, recommendations, etc. Write it in form of whole sentences, no bullet points.
>
> Small example of output replaced with dashes:
>
> Trip around _: A challenge in the Czech countryside The trip around the Czech town of _ covers length of 150 km, featuring total of climb of 2500 m, both uphill and downhill. It is a challenge even for most experienced cyclists. Do not replace the names with dashes in the final annotation. If you do not have the names, do not make them up.

- Example output:

> Tábor Loop Adventure: Exploring the Czech countryside in a 102.9 km circular route starting and ending at Žižkovo náměstí in Tábor. This challenging journey features a total ascent of 969.67 m and total descent of 971.76 m, offering breathtaking views of the Southwest region. It is recommended for experienced cyclists seeking a rewarding adventure in the heart of Czechia.

4.4.12 Final annotation

The final annotation is generated using a language model. The input to the language model is string `trip`, created in the following way:

- append the introductory string to `trip`
- append description of each segment to `trip`, separate every n^{th} segment with `--- split here ---` according to the desired number of paragraphs.[3]

An example of a possible input to the model:

> . . .
>
> The segment is a short, 9 km long, flat road with a slight elevation gain of 30 meters ...
> — split here —
> The segment is a 6.2 km long, flat route with low to medium traffic density. The road ...
> The road is flat with a slight grade, paved and narrow, with medium traffic density. ...
> The route is a flat, narrow, and very narrow paved road with low to medium traffic ...
> — split here —
> The segment is approximately 10 kilometers long with a very low to medium traffic density.
>
> . . .

The model setup is as follows:

[3] We chose $n = 3$ for our implemantation.

- Model: GPT-4
- Temperature: 0.7
- Input: `trip`
- Prompt:

  Adjust the given text so it sounds like a tourist book cycling itinerary description. Try to mimic what a human-written cycling trip description would sound like. Adjust the description to the user preferences, if they are not none.

  Very important! Keep the language simple, clear, concise, and neutral in tone. DO NOT USE FLORID LANGUAGE!!!

  First section is the name and the introduction. The following paragraphs are the text body.

  Keep the existing splitting of the paragraphs, which is marked as '— split here —' for your easier identification. Do not create any new paragraphs.

  You must include places marked with ∗'s, they are very important to mention. But remove the ∗'s. But you can for sure mention any other place. If it is relevant or interesting, you can add more detail. Try to maintain level of information relevant to the cyclist, as a cycling book would do.

  Do not overwhelm the user with numbers. ROUND THE NUMBERS USED, no cyclist is interested in decimal points nor precise meters.

  Try to maintain consistent level of description throughout the whole trip and a similar length of each paragraph.

- Example outputs are provided and evaluated in Chapter 5

Chapter 5
Evaluation

Since the quality of a cycling trip description can be perceived subjectively to a great extent, it is hard to define objective criteria or to evaluate the model automatically. In this chapter, we define our evaluation instances, methods, and criteria and evaluate various aspects of the annotator, including:

- Accuracy
- Financial requirements
- Time complexity
- User preference fulfillment
- Parameter modification

5.1 Evaluation instances

As a source of quality cycling trips and their descriptions, we chose the book *Srdcem na kole Českem* [22] (translated as Cycling in Czechia with passion). The book was written with the contribution of several professional cyclists and contains detailed descriptions of cycling trips. Additionally, links to .gpx files are provided to import trip plans easily.

From this book, we chose three[1] example trips:

Number	Name	Page
1	Toulavou [5.1.1]	238
2	Přes tři kopce srdcem Orlických hor [5.1.2]	104
3	Oddechovka kolem řeky Ohře [5.1.3]	117

Table 5.1. Example trips. Source: Srdcem na kole Českem [22].

We provide the basic information about each of the trips - the route, length, elevation, and difficulty as stated in the book. The full texts can be found in the book.

The .gpx files are used for the annotator input.

[1] The number of example trips is low due to the need for manual inspection and evaluation, which is a time-consuming process.

5.1.1 Example trip 1

The first example is a trip around Tábor in South Bohemia. It spans approximately 112 km, with uphill and downhill both 1026 m. The difficulty is estimated to be 4/10.

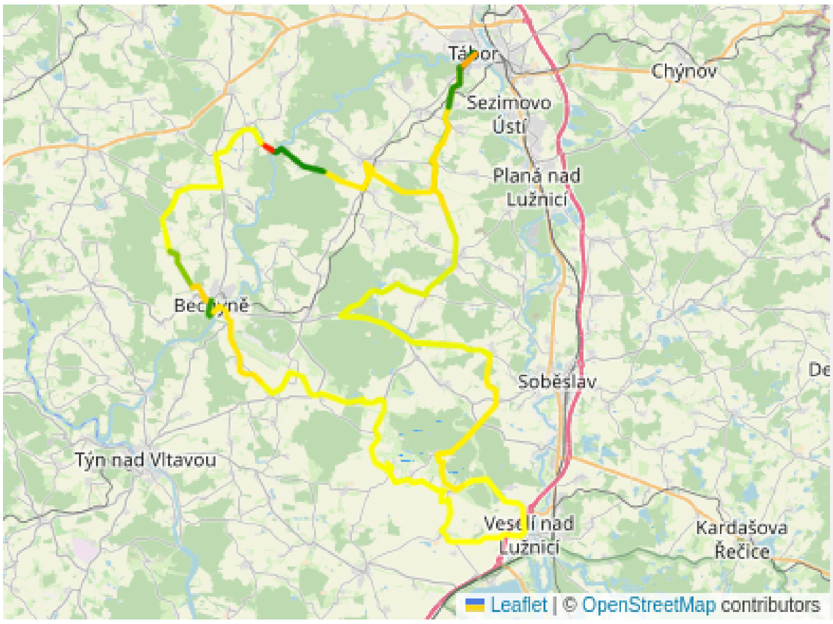

Figure 5.1. Example trip 1 visualized.

5.1.2 Example trip 2

The second example is a 56 km long trip in Orlické hory (Eagle Mountains) in northeastern Bohemia, with a total elevation gain of 1274 m and estimated difficulty of 5/10.

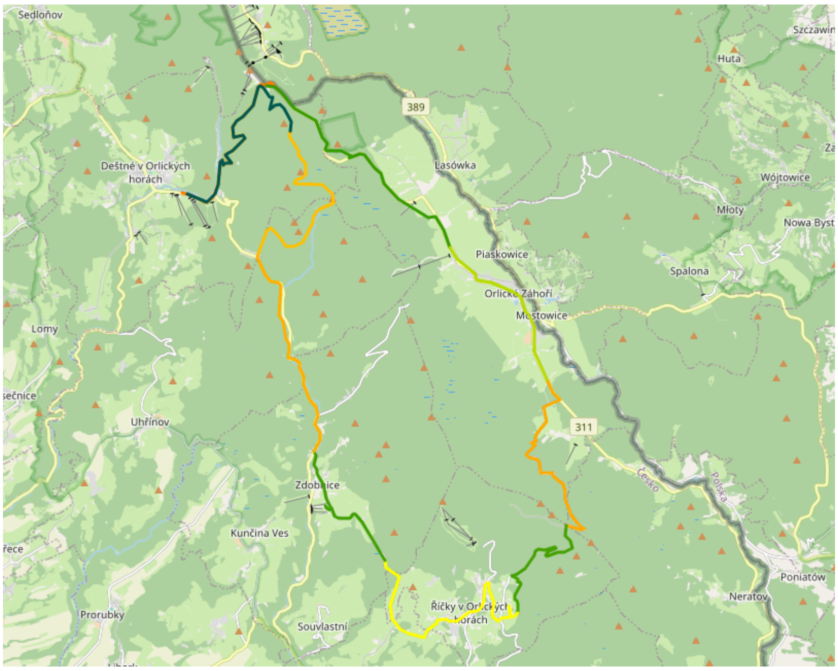

Figure 5.2. Example trip 2 visualized.

5.1.3 Example trip 3

The third example is a trip near Most in Western Bohemia spanning 90 km, with an uphill climb of 665 m and an estimated difficulty of 3/10.

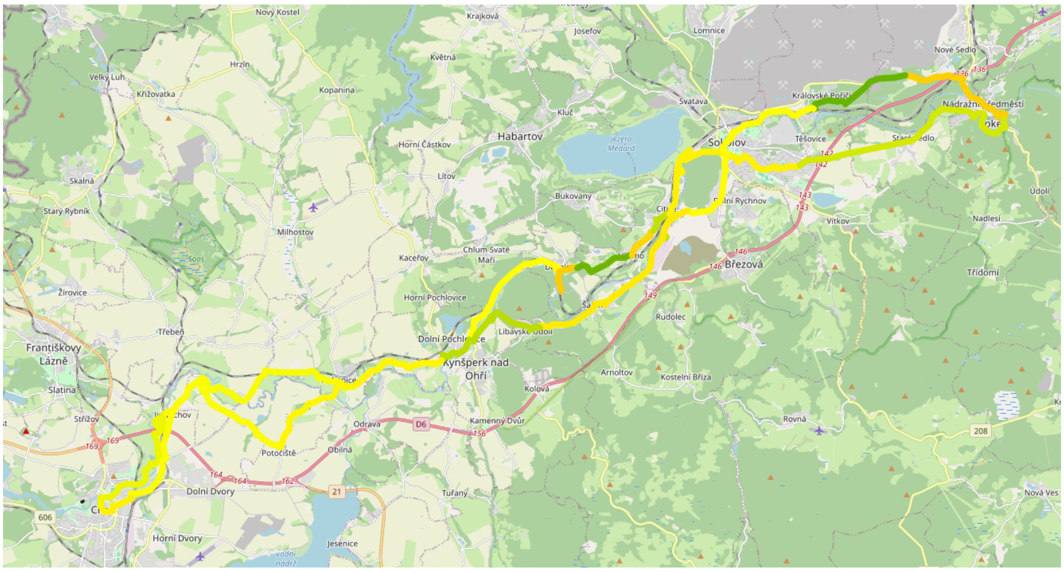

Figure 5.3. Example trip 3 visualized.

5.1.4 Example preferences

We further test the annotator's ability to work with user preferences. It is evaluated using the following example preferences:

Number	Preference
1	`I like beer and good food` 5.6.1
2	`I like to learn about local traditions` 5.6.2
3	`military, aviation` 5.6.3

Table 5.2. Example user preferences.

5.2 Evaluation tools

The following methods simplify the evaluation.

5.2.1 Visualisation

Visualization is used to inspect the trip route manually and to accompany the description in the interactive mode.

We use the Leaflet library to interactively visualize various aspects of the trip in layers over the OpenStreetMap map base. The following are especially useful:

- **Segments**, colorized according to their steepness (from steep uphill in red, through flat terrain in yellow, to steep downhill in green), with associated information in pop-up windows
- **Features** around the path, with arbitrary associated information in pop-up windows

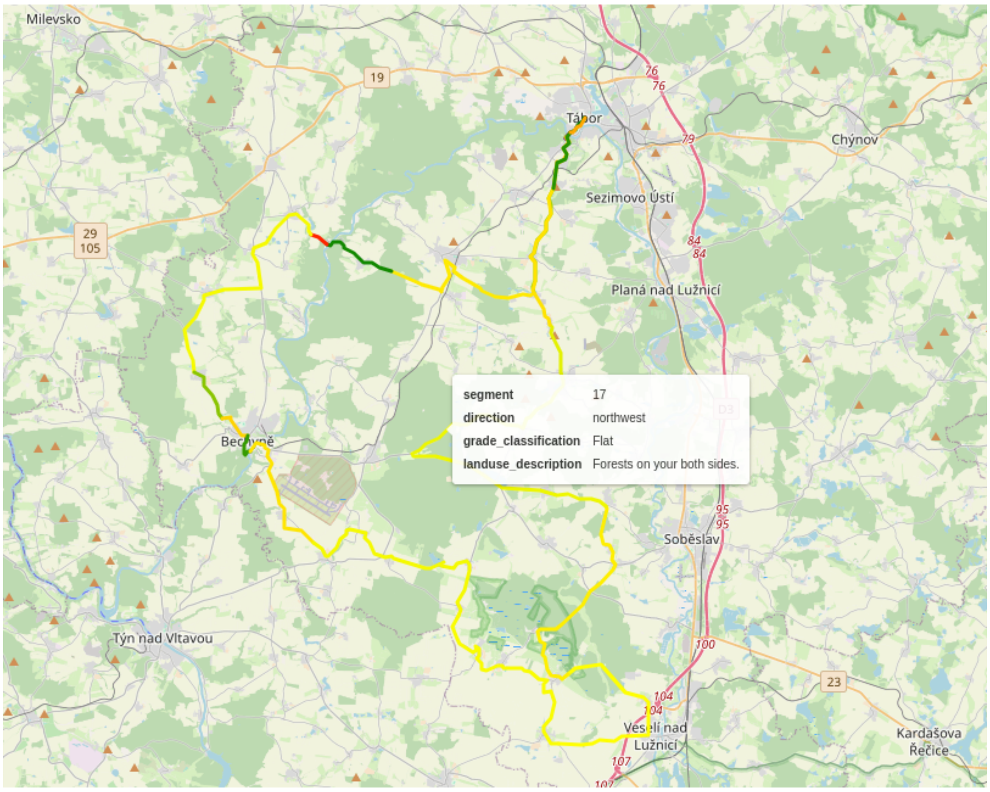

Figure 5.4. Example of segment visualization.

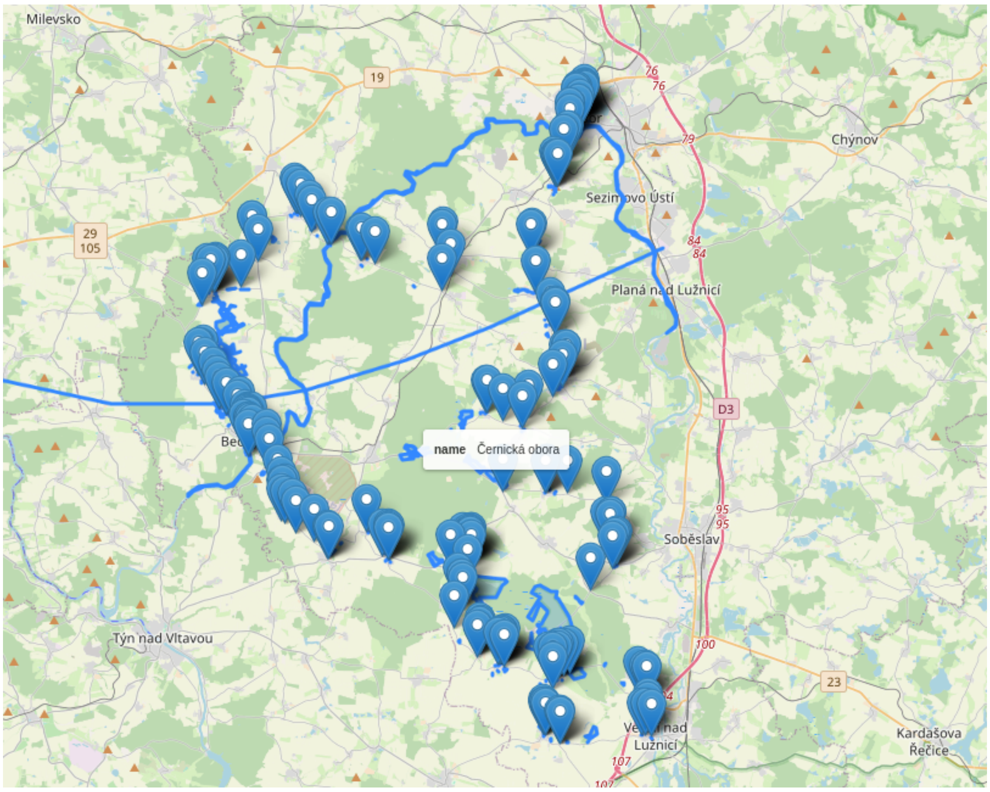

Figure 5.5. Example of trip feature visualization.

5.2.2 Text formatting

To simplify orientation in the long description, we can format the output as HTML and attach the available Wikipedia links to simplify access to additional information.

To format the article as HTML, we can include this requirement in the prompt for the language model when generating the final annotation. We specifically instruct the language model to:

```
Format the article as HTML -
<h1> for the name of the article
and <p> for each paragraph.
```

Then, if available, we replace the names in the text with their respective links to Wikipedia to have direct access to more information.

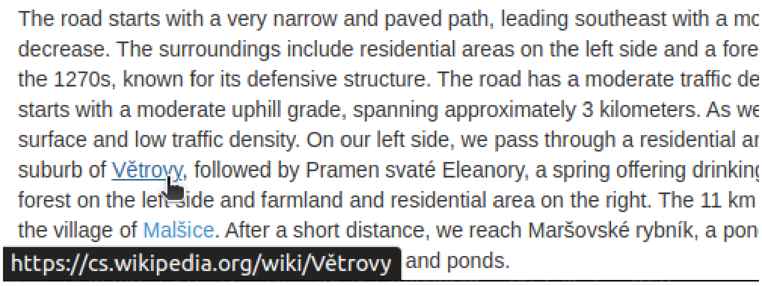

Figure 5.6. Direct link to Wikipedia.

5.3 Accuracy evaluation

For accuracy evaluation, we compute annotation quality scores manually based on defined criteria regarding terrain, settlements, and points of interest that the cycling trip itinerary should meet.

5.3.1 Accuracy evaluation criteria

The criteria have two levels:

- **MUST**
  - Mention all uphills and downhills with an elevation change of more than 100 meters.
  - Mention all settlements with a population of more than 200 inhabitants.
  - Mention all points of interest along the path that have more than 500 reviews and more than a 4.0 average rating on Google Maps.
- **SHOULD** (generally weaker **MUST** criteria):
  - Mention all uphills and downhills with an elevation change of more than 50 meters.
  - Mention all settlements.
  - Mention all points of interest along the path that have more than 50 reviews and more than a 3.5 average rating on Google Maps.

5.3.2 Accuracy evaluation for example trips

We manually inspect each of the example trip routes against the map and create reference tables. The tables containing every aspect of each trip can be found in Appendix A. Then, we generate an annotation for each of the example trip (without any personalization) and evaluate the annotations using the reference tables.

Example trip 1 annotation

Trip around Tábor: Exploring the Scenic Southwest Region of Czechia

The trip around the Czech town of Tábor spans roughly 113 km with a moderate elevation change, weaving through picturesque countryside landscapes and rich cultural heritage sites. This moderately challenging route is perfect for cyclists eager to explore the scenic Southwest of Czechia. Be sure to carry ample water and snacks as you pedal through this beautiful region.

The journey begins with a gentle downhill stretch, paving through a narrow road amidst residential views. Starting at 455 meters, the path descends to 405 meters, bringing you close to Kotnov Castle - a historic fortress dating back to the 1270s. This scenic descent is followed by a sharp uphill climb on a mix of terrains, presenting a chance to visit the natural spring, Pramen svaté Eleanory, ideal for a refreshing pit stop. The ascent reaches an elevation of 454 meters, offering splendid views of the surrounding areas. The road then continues to climb to 522 meters, passing through charming villages such as Větrovy and Horky, with the peak of Hýlačka visible en route.

The next segment brings a gentle decline over 4.7 km, passing through tranquil forests and farmlands, and by several bodies of water including Lom Reservoir and various ponds like Maršovské rybník. The village of Lom greets travelers on this southbound path. A subsequent 8 km stretch with a mild ascent goes through Malšice, featuring local landmarks such as the Boží muka wayside shrine and several ponds. The road then descends over 3 km through dense forests, leading past the Dobřejice village and the notable Stádlecký řetězový most, a picturesque suspension bridge.

Next, a steep 768-meter climb takes riders to about 450 meters elevation through serene forests. This short stretch is followed by an 11 km descent through a blend of landscapes, including the historic settlements of Stádlec and Staré Sedlo, and the expansive Závist pond amongst others. The route then transitions into a gentle downhill across 2.4 km, moving through the lush Táborská vrchovina mountain range and past the vibrant community of Radětice.

Progressing further, a steady climb over 1.5 km leads past the Koupaliště Bechyně swimming pool and towards Bechyně, where the grand Zámek Bechyně castle awaits. The road then takes a sharp rise over 6 km, providing views of local sites like the Kopanina and the scenic Radostný pond, leading towards the town of Bechyně, enriched with cultural heritage and natural beauty.

Following this, a flat 11 km stretch passes war memorials and quaint villages such as Klečaty and Hodětín, enriched with local ponds like Nadějka and Rožberk. The journey continues over a similarly flat 10 km, adorned with historic wayside crosses and vibrant communities like Zálší and Sviny, before reaching the serene surroundings near Žíšov, highlighted by the Přírodní památka Veselská blata nature reserve.

As the route nears its end, a 10 km stretch with a mild ascent goes past the village of Vesce, reaching the natural reserve Přírodní rezervace Kozohlůdky. The segment concludes past the historic Borkovice and the picturesque Pískovna Jitra pond. A slight descent over 10.8 km then takes cyclists through lush forests and by the Golfový Klub Panství Bechyně, with the journey winding down near the historic Zámek Černice castle.

The final legs of the trip include a charming 6.3 km climb passing Maršov and its surrounding water features, a refreshing downhill to 405 meters through a beauteous park and reservoir area, and lastly a moderate uphill that introduces the pristine přírodní památka Granátová skála nature reserve, concluding this exhilarating cycling adventure around Tábor.

MUST hills correctly mentioned	3
MUST hills not mentioned	0
MUST hills success rate	100 %
SHOULD hills correctly mentioned	3
SHOULD hills not mentioned	2
SHOULD hills success rate	60 %
MUST settlements correctly mentioned	9
MUST settlements not mentioned	4
MUST settlements success rate	70 %
SHOULD settlements correctly mentioned	9
SHOULD settlements not mentioned	10
SHOULD settlements success rate	47 %
MUST tourist attractions correctly mentioned	3
MUST tourist attractions not mentioned	1
MUST settlements success rate	75 %
SHOULD tourist attractions correctly mentioned	2
SHOULD tourist attractions not mentioned	1
SHOULD tourist attractions success rate	67 %

Table 5.3. Accuracy evaluation of Example trip 1, based on tables in A.1.

Example trip 2 annotation

Circular Route around Jedlová v Orlických horách: A Scenic Adventure in the Czech Countryside

The circular route around Jedlová v Orlických horách spans approximately 54 kilometers, featuring a total ascent and descent of around 1300 meters each. This journey presents cyclists with a challenging and rewarding experience through the scenic landscapes of okres Rychnov nad Kněžnou in the Královéhradecký kraj region of Northeast Czechia. Cyclists are advised to prepare for varied terrains and breathtaking views.

The route begins with a demanding 5.4 km uphill stretch, leading cyclists from an elevation of 650 meters to 990 meters. The road, bordered by lush forests, alternates between paved and unpaved surfaces and has very low to medium traffic density. En route, cyclists can explore notable sites including the historic X/26/B2-90 Z bunker and the protected areas of Kačenčina zahrádka and CHKO Orlické hory. The segment proceeds with a 6.5 km downhill, passing through the beautiful wetlands of Zelenka and the nature reserves of Trčkov. Further along, a 4.4 km gentle descent guides cyclists past the quaint village of Orlické Záhoří and historic landmarks such as the Ignác Preissler memorial.

The journey continues with a moderate 5.5 km uphill ride, where cyclists will navigate through more captivating forest landscapes and past several historical military bunkers, including Izolovaný pěchotní srub R-S 89 U silnice. The subsequent 4 km downhill stretch offers a scenic descent, leading cyclists through the PR Komáří vrch nature reserve, renowned for its diverse ecosystem. A flat 7 km segment follows, taking cyclists through the village of Říčky v Orlických horách, surrounded by dense forests and featuring attractions like Adventure Golf Říčky.

The final stretches of the route include a 3.9 km downhill to the village of Zdobnice, a reversed 5.4 km uphill passing through Luisino údolí and Kamenec, and a 5.7 km uphill leading towards the prominent Velká Deštná peak. Cyclists will also pass by the Jelení lázeň nature reserve. Concluding the journey is a steep 6.4 km downhill, where riders will be treated to views of the Malá Deštná peak, completing this adventurous cycling tour around Jedlová v Orlických horách.

MUST hills correctly mentioned	7
MUST hills not mentioned	1
MUST hills success rate	88 %
SHOULD hills not available	
MUST settlements correctly mentioned	2
MUST settlements not mentioned	0
MUST settlements success rate	100 %
SHOULD settlements correctly mentioned	2
SHOULD settlements not mentioned	0
SHOULD settlements success rate	100 %
MUST tourist attractions correctly mentioned	0
MUST tourist attractions not mentioned	1
MUST tourist attractions success rate	0 %
SHOULD tourist attractions correctly mentioned	2
SHOULD tourist attractions not mentioned	1
SHOULD tourist attractions success rate	67 %

Table 5.4. Accuracy evaluation of Example trip 2, based on tables in A.2.

Example trip 3 annotation

Trip around Cheb: Exploring the Rolling Hills of Karlovarský Kraj

The trip around the Czech town of Cheb spans roughly 90 km, with an elevation change of about 600 meters. This route guides cyclists through the scenic landscapes of Northwest Czechia, posing a moderate challenge suitable for riders of various experience levels. Embark on this cycling adventure and enjoy the natural beauty and cultural landmarks of the Cheb District.

The journey begins with a tranquil 6 km stretch, descending gently from 450 meters to 430 meters. The path, a mix of unpaved and paved surfaces, meanders through meadows and residential zones with a backdrop of a railway and farmlands. Notable spots along this segment include Jánské náměstí square, Kostelní náměstí square, and a vibrant Graffiti wall, all within close proximity. The presence of a nearby river and a variety of natural landscapes makes this an enjoyable introduction to the region.

Progressing into the route, the next 8 km offer a slight descent from 430 meters to 418 meters. As you pass through the hamlet of Chocovice and the village of Nebanice, the surroundings unfold with forests and open fields, presenting an authentic rural Czech experience. This segment connects to a leisurely path through Chotíkov and Dasnice, enhancing the ride with views of local life and natural beauty.

The road then inclines moderately for about 2 km, rising from 420 meters to 471 meters. This section, surrounded by forests and a nearby railway, introduces cyclists to a serene woodland area and a quaint wetland, perfect for nature enthusiasts. Following this, a gentle descent leads to the village of Hlavno, offering a peaceful ride through more wooded landscapes and the charming locality of U lomu.

As the route continues, it gently descends for nearly 3 km from 440 meters to 406 meters, transitioning through an industrial stretch into more natural settings. Key points include the village of Citice and the scenic U hřiště locality. Climbing next from 406 meters to 461 meters over 7.5 km, riders will discover diverse landscapes including the picturesque jezero Medard lake, Husovy sady park, and the historic Sokolovský zámek castle, enriching the cultural tapestry of the journey.

The most challenging segment awaits with a 4 km climb at a 6% grade, where dense forests flank the path leading to the historic town of Loket. Here, riders can explore several historic sites including the Loket Castle or Hrad Loket, offering a glimpse into the region's medieval past. The route then descends towards Královské Poříčí, where local monuments and natural surroundings continue to captivate.

The final leg of the journey is a mix of gentle climbs and descents through lush forests and quaint villages. The road takes cyclists through Libavské Údolí and the picturesque village of Šabina, among others, culminating in a stretch that leads back to Cheb. This last segment is relatively flat, easing riders back into urban surroundings while still offering views of natural beauty and local life.

Concluding in Cheb, cyclists are greeted by the historical Městský dům, marking the end of a memorable cycling adventure through the Karlovarský kraj. This route not only challenges the body but also enriches the soul with its natural landscapes and cultural encounters.

MUST hills not available	
SHOULD hills correctly mentioned	4
SHOULD hills not mentioned	3
SHOULD hills success rate	57 %
MUST settlements correctly mentioned	8
MUST settlements not mentioned	3
MUST settlements success rate	73 %
SHOULD settlements correctly mentioned	3
SHOULD settlements not mentioned	6
SHOULD settlements success rate	33 %
MUST tourist attractions correctly mentioned	2
MUST tourist attractions not mentioned	1
MUST tourist attractions success rate	67 %
SHOULD tourist attractions correctly mentioned	1
SHOULD tourist attractions not mentioned	2
SHOULD tourist attractions success rate	33 %

Table 5.5. Accuracy evaluation of Example trip 3, based on tables in A.3.

5.3.3 Accuracy evaluation results

We summarized the results from the example trip evaluations. The annotator performed well when mentioning significant hills, with an accuracy of 94 %. The annotator also mentioned more than 80 % of significant settlements. Interestingly, the annotator was less successful in identifying significant tourist attractions (47 %) than the less significant ones (56 %). This occurrence might have been caused by a limited number of test cases.

The complete results can be found in the following table:

Category success rates	Example 1	Example 2	Example 3	Average
MUST hills	∅	88 %	100 %	**94 %**
SHOULD hills	57 %	∅	60 %	**58 %**
MUST settlements	73 %	100 %	70 %	**81 %**
SHOULD settlements	33 %	100 %	47 %	**60 %**
MUST tourist attractions	67 %	0 %	75 %	**47 %**
SHOULD tourist attractions	33 %	67 %	67 %	**56 %**

Table 5.6. Accuracy evaluation results

5.4 Financial requirements evaluation

Some of the services we use, namely OpenAI language models and Google Elevation API, are paid. We compute the approximate price of the annotation generation for each of the evaluated trips using the following pricing policies.

5.4.1 Pricing of used services

Google Elevation API is a pay-as-you-go service, meaning the price is fixed per request. The number of locations per request is limited to 512. The pricing[2] is shown in the following table:

Monthly volume range	Price per request
0–100,000	0.005 USD
100,001–500,000	0.004 USD
500,000+	0.004 USD

Table 5.7. Google Elevation API pricing as of May 2024.

Our project is small-scale and belongs to the first group paying $0.005 per request.

OpenAI models' pricing[3] is based on the choice of a specific model. The price is computed from the input and output length in tokens.

A token is a unit of text, usually a short word or a part of a word, both accepted and generated by a model. While its length is not given and depends on the used tokenizer, it consists of approximately four characters on average. Another approximate is 30 tokens per 1-2 sentences [23].

The following table shows pricing for the OpenAI models we are using.

Model	Price per input	Price per output
gpt-4-turbo-2024-04-09	$10.00 / 1M tokens	$30.00 / 1M tokens
gpt-3.5-turbo-0125	$0.50 / 1M tokens	$1.50 / 1M tokens

Table 5.8. OpenAI language model pricing as of May 2024. Source: [cite].

5.4.2 Pricing calculation

We use the following formula for total price computation.

$$price_{total} = price_{ele} + price_{pre} + price_{seg} + price_{ann}$$

- The price for obtaining elevation data is considered constant for every use of the annotator, as the amount of elevation data should fit into one API call.

$$price_{ele} = \$0.005$$

- The price for using GPT-3.5 to parse user preference is also considered constant, as the user preference is usually expressed by one sentence.

[2] https://developers.google.com/maps/documentation/elevation/usage-and-billing

[3] https://openai.com/api/pricing/

- Input has approximately 200 tokens for prompt and
- Output has approximately 40 tokens.

$$price_{pre} = (230 + 30) \cdot 0.5 \cdot 10^{-6} + 40 \cdot 1.5 \cdot 10^{-6} = 0.000155$$

- Price for using GPT-3.5 to annotate all the segments depends on number of segments n.
  - Input has approximately 300 tokens per segment, including the introduction paragraph, and 250 tokens for the prompt.
  - Output has approximately 150 tokens per segment.

$$price_{seg} = (n + 1) \cdot ((300 + 250) \cdot 0.50 \cdot 10^{-6} + 150 \cdot 1.50 \cdot 10^{-6}) = (n + 1) \cdot 0.0005$$

- The price for using GPT-4 to create the final trip annotation depends on the number of segments n.
  - Input has approximately 150 tokens per segment, including the introduction paragraph, and 300 tokens for the prompt.
  - Output has approximately 100 tokens per segment.

$$price_{ann} = (n + 1) \cdot ((150 + 300) \cdot 10.0 \cdot 10^{-6} + 100 \cdot 30.00 \cdot 10^{-6}) = (n + 1) \cdot 0.0075$$

5.4.3 Financial requirements evaluation results

Prices of annotation of the three example trips, if user preference parsing is also considered:

Task	Example 1	Example 2	Example 3
Elevation data fetch	0.005	0.005	0.005
GPT-3.5 preference parsing	0.000155	0.000155	0.000155
GPT-3.5 segment description	0.0105	0.0055	0.009
GPT-4 final annotation	0.1575	0.0825	0.135
Total price	0.173155 USD	0.093155 USD	0.149155 USD

Table 5.9. Financial requirements of annotating the example trips.

5.5 Time complexity evaluation

We measure the time of each operation of the annotation process and provide an overview of the time complexity for the individual operations.[4]

[4] The time necessary to execute locally running operations is negligible in comparison to the remote operations and is therefore collectively referred to as Other processes.

Task	Example 1	Example 2	Example 3	Average
Elevation data fetch	3.0 s	2.5 s	2.5 s	2.67 s
Neighborhood data fetching	10.5 s	7.5 s	14.5 s	10.83 s
Feature fetching	16.5 s	16.5 s	28.5 s	20.50 s
Wikipedia data fetching	38.5 s	7.0 s	17.0 s	20.83 s
Segment annotation	28.0 s	40.0 s	52.5 s	40.17 s
Final annotation generation	39.5 s	21.0 s	41.0 s	33.83 s
Other processes	4.0 s	3.0 s	4.0 s	3.67 s
Total time	139.5 s	82.5 s	134.0 s	164.0 s

Table 5.10. Time complexity 5.5 of the example trips.

The time consumed by the particular operations significantly differs each time, even though there is no specific reason for this caused by the implementation besides the trip length. It most probably depends on factors such as the amount of data processed, the Internet connection speed, the current usage of the remote servers, the device's hardware performance, and possibly also the amount of cached information.

Additional time complexity testing would be necessary to identify specific patterns. However, we did not conduct these tests, as the value of such information does not justify the considerable cost.

5.6 Evaluation of user preference setting

The default annotations are generated without stating a specific user preference. In this section, we will instruct the annotator to generate descriptions of the same trip as in Example 1 [5.1.1], but each time with different user preferences taken from table 5.2. We state the preferences, show the generated preference tags, and highligh their impact on the resulting description.

5.6.1 Example preference 1

- User preference: `I like beer and good food.`
- Resulting tags:

```
{
     "amenity": ["pub", "bar", "restaurant", "cafe"],
        "shop": ["brewery", "liquor", "deli", "grocery"],
     "cuisine": "beer"
}
```

- The description:

Tábor Loop: Exploring the Czech Countryside on a 113 km Circular Route

The Tábor Loop is a delightful 113 km cycling journey around Tábor, Czechia, offering a blend of picturesque countryside and historical sites. With a gradual ascent and descent of around 1100 meters, cyclists will enjoy varied landscapes and several cultural points of interest. It's advised to carry ample water and snacks, and to bring along your spirit of adventure.

The ride begins on a narrow, paved path that descends gently for about a kilometer. You'll be flanked by residential areas and a lush forest. The segment is mostly downhill, dropping about 50 meters in elevation. Notably, you'll pass by the **Hotel Nautilus, a great spot for regional cuisine**. The road then transitions into a steep uphill over a short distance, encountering the Eleanořin pramen, a natural spring perfect for a refreshing stop. Further along, the road climbs moderately for 2.6 km, passing through the quaint suburbs of Horky and Větrovy, where you can stop at the **Pintovka restaurant for some authentic Czech dishes**.

The route then descends slightly over 4.7 km, through serene forests and farmlands. Here, you'll encounter the Lom reservoir, and further along, the Maršovské and Horní rybník, offering delightful views and a peaceful cycling environment. Ascending gently for 8 km, the path takes you through more scenic landscapes, including a park and several water bodies. Don't miss the **local bar, Vinárna Pod věží** in Malšice. The path then descends moderately for 3.2 km, passing historic sites like the Stádlecký řetězový most and the Boží muka shrine.

Continuing, the road climbs steeply for a short distance before transitioning into a relaxing 11 km one-way downhill, passing through the charming villages of Staré Sedlo, Rataje, and Stádlec. Here, the **Hospoda U Kyriánů restaurant** near Zámek Stádlec castle offers a quaint stop. The route then gently descends over 2.4 km through more farmland, leading to the village of Radětice and close to the scenic Táborská vrchovina mountain range.

The ride gradually ascends for about 1.5 km, bringing you near the **Na Růžku restaurant** and the locality of Poušť, before transitioning into a moderate downhill stretch where the historic Zámek Bechyně castle awaits. The route climbs slightly over 6 km, passing through military areas and the peaceful village of Hodonice. The segment ends with a flat stretch through forests and farmlands, showcasing local flora and fauna and concluding at the scenic Pískovna Jitra pond.

Finally, the route continues flat for about 10 kilometers, passing through the forested area and farmland with occasional villages like Želeč and Obora, where **local pubs** and historic ponds enrich the journey. The road then ascends slightly, passing through Maršov, where you can enjoy breaks at picnic tables near the Maršovské rybník pond. The journey concludes as you cycle back towards Tábor, passing the **Výčep restaurant, perfect for a hearty meal** after a fulfilling ride.

5.6.2 Example preference 2

- User preference: `I like to learn about local traditions.`
- Resulting tags:

```
{
              "tourism": "museum",
             "historic": true,
    "historic:cultural": true,
              "amenity": "cultural_center",
              "leisure": ["theatre", "dance"],
                "craft": true
}
```

- The description:

Trip around Tábor: A Scenic Adventure in the Czech Countryside

The cycling adventure around the Czech town of Tábor spans approximately 110 km and involves an elevation gain of about 1100 m. The journey offers a balanced mix of uphill and downhill stretches, perfect for cyclists eager to experience the picturesque countryside and local traditions of this region. Make sure to pack sufficient water and snacks for this exhilarating ride.

The journey begins with a gentle downhill slope on a narrow paved road, taking you through a residential area to your left and lush forest to your right. This stretch is moderately trafficked and descends about 50 m. Not far from here, the **Museum of Puppetry Arts beckons with cultural insights**, located just a short detour from the route. The road then shifts to a challenging uphill climb over 330 m, increasing your elevation by about 50 m. This segment, featuring a blend of paved and unpaved surfaces, passes by the refreshing Eleanor's Spring, an ideal spot for a quick water refill, and the serene suburbs of Horky and Větrovy, before reaching the peak of Hýlačka.

Continuing on, the route offers a slight downhill through forests and farmlands, during which you'll ride past the quaint village of Lom and a series of scenic ponds including Dolní rybník and

Maršovské rybník. The road then climbs over 8 km, starting from 430 m and rising to 530 m, through more captivating landscapes. Key sites along this section include the village of Malšice, the historic Boží muka wayside shrine, and the dramatic Stádlecký řetězový suspension bridge, before concluding at the small village of Dobřejice.

As the road steeply ascends, you'll be enveloped by forests on both sides, with a notable stop near the wetlands at 620 square meters. Descending slightly over 11 km, the path takes you through Staré Sedlo and Stádlec, home to the intriguing Zámek Stádlec castle, and on through Rataje and several rustic ponds, concluding with a gentle ride through the rolling farmlands of Radětice, enveloped by the Táborská vrchovina mountains.

The route then transitions into a moderate uphill towards Bechyně, passing the **International Museum of Ceramics** and the historical Zámek Bechyně castle, a majestic site nestled in this charming town. The road ascends gently over 6 km through military zones and pastoral landscapes, highlighting the village of Hodonice and its surrounding natural beauty, before flattening out towards the serene villages of Klečaty and Březnice, adorned with a poignant war memorial.

The final stretches of the trip are characterized by tranquil rides through lush farmlands and forests, dotted with historic wayside crosses and natural reserves like Veselská blata. The route concludes with an engaging ride through Mokrá and Borkovice, passing the Kozohlůdky nature reserve and several idyllic ponds, before the final push back to Tábor, leaving cyclists enriched with the natural and cultural splendor of this Czech landscape.

5.6.3 Example preference 3

- User preference: `military, aviation`
- Resulting tags:


```
{
      "landuse": "military",
     "military": true,
     "building": "hangar",
 "building:use": "hangar",
      "aeroway": ["aerodrome", "helipad", "runway"],
     "man_made": "tower"
}
```


- The description:

Trip around Tábor: A Scenic Adventure in the Southwest Czech Countryside

The trip around the Czech town of Tábor spans 110 km and includes a total ascent of 1100 m, featuring both challenging uphills and thrilling downhills. This route treats cyclists to breathtaking views of Southwest Czechia and is ideal for those seeking a picturesque yet moderately demanding cycling experience.

The journey begins on a narrow, paved stretch with a gentle downhill slope. This 1 km segment has a slight elevation drop and low to medium traffic, making it a smooth start. As you ride, you'll encounter residential areas and a forest, adding to the scenic variety. Noteworthy is the nearby Muzeum loutkařského umění, an intriguing stop for enthusiasts of puppetry arts. The challenging ascent soon follows, stretching over 330 meters with a substantial elevation gain, leading you past the natural spring, Pramen svaté Eleanory. The road then takes you through the suburbs of Horky and Větrovy, up to the peak of Hýlačka, offering splendid views and a rewarding ride.

As you descend over 4.7 km from 520 meters to 430 meters, the terrain shifts between forest and farmland, enriching your ride with diverse landscapes. Key landmarks include Dolní rybník and Horní rybník reservoirs, and the quaint village of Lom. The subsequent 8 km stretch features a mild uphill, paving the way through Malšice and past historical sites like Boží muka. The segment ends with a descent into Dobřejice, passing the notable Stádlecký řetězový most, a picturesque suspension bridge.

Continuing the ascent, the road climbs steeply for 770 meters, surrounded by dense forests, leading to the serene villages of Staré Sedlo and Stádlec, home to the historic Zámek Stádlec castle. The road then gently descends through Radětice, offering views of the Táborská vrchovina mountain range. This serene stretch is perfect for reflecting on the natural beauty of the region.

The final legs of the journey include a 1.5 km uphill ride, passing close to the Mezinárodní muzeum keramiky and the locality of Poušť. A gentle downhill leads you towards Bechyně, where the majestic Zámek Bechyně castle stands. **The route then takes a historical turn through WWII era landscapes, including a pass by the Vojenské letiště Bechyně, a disused military airfield, enhancing the trip for military history buffs.**

The concluding stretches maintain a flat terrain, ideal for leisurely cycling. You'll pass through the villages of Klečaty, Březnice, and Hodětín, each offering unique local charm and **historical sites, including a war memorial** and several scenic ponds. The route ends with a reflective ride through Želeč and Obora, passing by natural reserves and historical landmarks, culminating in a tranquil finale to the scenic adventure around Tábor.

We can see that the annotator correctly parses the user preference to tags. The tags provide a tool to fetch relevant features, which are then used in the descriptions.

5.7 Parameter modification evaluation

The annotator has many parameters that can be adjusted, potentially affecting performance and output quality. However, it is not possible to evaluate changes for all parameters. Therefore, we focus on modifying the parameters that were not specifically explained previously, to observe their effects. We demonstrate these changes by annotating the same trip as in Example 1 [5.1.1], allowing direct comparison of the outputs.

5.7.1 Different language model choice

The language models used in our annotator were chosen based on the estimated difficulty of the tasks. For more routine tasks, such as single segment description or user preference parsing, a less powerful model (GPT-3.5) was used, whereas for the final annotation generation, which is a more complex task, we used a state-of-the-art model (GPT-4).

Here, we try to use the less powerful model (GPT-3.5) for the final annotation.

Trip around Tábor: A Scenic Adventure in the Czech Countryside

The cycling journey around the Czech town of Tábor spans a total distance of 113 km, offering cyclists a delightful experience through the Southwest region of Czechia. With picturesque landscapes and some challenging hills, this route promises an exciting exploration of the countryside.

The road leads you to the village of Kotnov, where you can visit the historic Kotnov Castle dating back to the 1270s. As you venture southwest, you'll encounter the natural spring, Pramen svaté Eleanory, providing a refreshing stop along your journey. Pedal through residential areas, farmland, and forests, passing by quaint villages like Větrovy and Horky. Don't miss the opportunity to admire the natural peak of Hýlačka along the way.

Continuing your adventure, the route guides you through diverse landscapes, including forests, farmlands, and water reservoirs like Maršovské rybník and Dolní rybník. Explore the village of Malšice with its rich history and charming population. As you cycle, immerse yourself in the serene surroundings, characterized by rolling hills and picturesque views.

As you journey through the scenic countryside, make sure to stop by notable landmarks such as Boží muka, the village of Dobřejice, and the impressive Stádlecký řetězový most suspension bridge. Enjoy the challenging uphill climbs and rewarding descents while taking in the beauty of the Czech countryside.

Discover the rich history and natural beauty of the region as you pedal through forests, farmlands, and villages. Pass by significant points of interest like the historic Stádlec Castle, the village of Staré Sedlo, and the picturesque water bodies dotting the landscape. The route offers a perfect blend of natural wonders and cultural heritage to enhance your cycling experience.

Continue your journey through the enchanting landscape, passing by charming villages, historic landmarks, and peaceful water bodies. Explore the diverse points of interest along the way, from war memorials to wetland areas, providing a glimpse into the region's unique heritage. Pedal through forests, farmlands, and quaint villages, enjoying the tranquil surroundings and leisurely pace of your cycling adventure.

As you approach the final stretch of your cycling trip, savor the scenic views of the countryside, passing through forests, farmlands, and villages. Explore historic sites, natural reserves, and water bodies, adding depth to your cycling experience. Take in the beauty of the surroundings as you pedal through this picturesque landscape, discovering the hidden gems of the Czech countryside.

We can see that the GPT-3.5 generated a similar beginning of the text, probably because it kept the input text's formulations and level of detail. The quality of the initial sections can be compared to the GPT-4 description. However, as the description progresses, the informativeness decreases, even though the model was instructed to be consistent, and the description gets vague. The second half of the trip is summarized into one very general paragraph, lacking precise names and references of specific places, which is not desired. The problem with generalization is not as prevalent when using GPT-4.

5.7.2 Language model temperature

Language model temperature [24] (on scale between 0.0 - 1.0) is a parameter controlling the randomness of the generated text. Lowering the temperature lowers the randomness and the output of the model is therefore more deterministic and obeying the prompt. With increasing temperature, the model might do less predictable choices leading to higher creativity of the output, but sometimes with decreased relevance or even causing hallucinations[5].

Our annotator uses the temperature of 0.7 for creating the final annotation and the temperature of 0.4 for the single segment annotation. These values should help the annotator describe the segments using the provided information, and then introduce some creativity to generate and appealing text.

We try to lower both temperatures to 0.1 and inspect the changes to the output.

Tábor Loop: Exploring the Southwest Countryside

The Tábor Loop offers a scenic 113 km journey through the Czech countryside, starting and ending at Žižkovo náměstí in Tábor. This moderately challenging route features a total ascent of about 1100 m and a similar descent, making it ideal for cyclists who enjoy beautiful landscapes and a bit of adventure.

The journey begins with a narrow, paved path leading southwest, featuring a moderate downhill grade. You'll experience a slight descent, dropping about 50 meters in elevation. The road, with low to medium traffic, passes by residential areas and the historic Kotnov Castle, a defensive structure from the 1270s. After a steep uphill climb of about 330 meters, you'll encounter the Pramen svaté Eleanory, a natural spring ideal for a refreshing stop. The road then continues uphill for 2.6 km, passing through the suburbs of Větrovy and Horky, and near the peak of Hýlačka.

Next, enjoy a gentle 4.7 km downhill ride through forests and farmland, passing by the Lom reservoir and several ponds including Maršovské rybník, Dolní rybník, and Horní rybník. The route then shifts to a slight uphill for about 8 km, featuring low to medium traffic and scenic views of Malšice village and several natural landmarks. A 3 km moderate downhill stretch follows, leading

[5] In simple terms, a language model is hallucinating when it generates unrelated or made-up content.

you past the Boží muka wayside shrine and the village of Dobřejice, before reaching the Stádlecký řetězový most, a notable suspension bridge.

The road continues with a steep 768-meter uphill segment through forests, featuring a mix of paved and unpaved surfaces. This leads into an 11 km stretch through the village of Staré Sedlo and past the historic Zámek Stádlec castle. The route then descends slightly for 2.4 km through farmland, passing by the village of Radětice and offering views of the Táborská vrchovina mountain range.

As you head southeast for about 1.5 km, the road climbs gently, passing by the Koupaliště Bechyně swimming pool and the locality of Poušť. A moderate downhill stretch of 1.3 km follows, leading you past Zámek Bechyně castle and into the town of Bechyně. The route then climbs slightly for 6.2 km, passing through the village of Hodonice and near several natural and historic points of interest.

The final segments of the loop take you through a peaceful 11 km stretch with slight inclines, passing through the villages of Klečaty, Blatec, and Hodětín, and near several ponds and a war memorial. A 10 km flat stretch follows, featuring low traffic and scenic farmland, leading past the village of Zálší and several natural landmarks. The route concludes with a flat, 8 km stretch through forests, passing by the village of Želeč and several water features, before returning to the starting point in Tábor.

Throughout the Tábor Loop, cyclists can enjoy a variety of landscapes, historic sites, and quaint villages, making it a memorable and enjoyable cycling experience in the heart of the Czech countryside.

Using lower temperature results in a more technical description, which, however, sometimes sounds more like a machine-generated listing distance and location. This does not necessarily have to be considered negative and could be even appreciated by some users, but our goal is to create an appealing description, and this text lacks emotion.

5.7.3 Distance around each segment

We set the distance around each segment to a default value of 200 m. The features within this distance should be observable, or in close proximity. In this section, we try to increase the radius to 1 km and observe if the features used for description are still relevant.

Trip around Tábor: A Scenic Route through Southwestern Czechia

The trip around the Czech town of Tábor spans a total of about 110 km, featuring an ascent of around 1120 m and a descent of about 1120 m. This cycling route is ideal for those seeking a moderately challenging ride amid picturesque landscapes.

The journey begins on a narrow paved path that slopes gently downhill for about 940 meters, dropping about 50 meters in elevation. The path, surrounded by residential areas and a forest, offers medium to low traffic, making it pleasant for cycling. Notably, you will pass by **Holečkovy sady**, a large nearby park, and encounter Pramen svaté Eleanory, a spring where cyclists can refresh themselves with drinking water. The route also includes a steep uphill of about 330 meters leading up to a hilltop view.

This segment continues with a steep descent over 95 km, dropping 20 meters through a scenic forest and residential backdrop. Points of interest include a World War I memorial. The path then leads into a 2.6 km uphill climb through serene forested areas and past the suburbs of Horky and Větrovy. Descending slightly through farmlands and quaint villages like Lom and **Libějice**, the route offers views of local reservoirs before climbing gently through more picturesque rural settings.

Proceeding with a gentle downhill over 3 km, the path provides a peaceful ride through dense forests. Highlights include the historic Boží muka shrine and the impressive Stádlecký řetězový most, a long suspension bridge. A challenging reversed uphill stretch follows, leading into a flat terrain that passes through the idyllic village of Staré Sedlo and several scenic ponds before reaching the town of Bechyně.

The route continues to ascend slightly, passing military zones and tranquil farmlands. Notable stops include the Hodonice village and several natural attractions like the **PP Židova strouha** nature reserve. A flat road then leads cyclists through forests and past rural communities such as Blatec and Březnice, concluding with a stretch through other serene villages and nature spots.

The final segments of the journey include a flat, slightly descending path through more captivating landscapes, with **Veselí nad Lužnicí** providing cultural enrichment and recreational opportunities. As the route winds down, it passes through the village of Svinky and near the Zámek Černice castle, before concluding with a scenic downhill to the starting point, circling back to the vibrant environments around Tábor.

Throughout the trip, cyclists are treated to a blend of historical sites, natural beauty, and a touch of Czech village charm, making it a fulfilling adventure for cycling enthusiasts looking to explore Southwestern Czechia.

The larger radius causes the annotator to consider some of the features to be present on the way when they are not. We can notice that the Holečkovy sady park, Židova strouha nature reserve, and Libějice village are present in this text, even though they are located further away and the cyclist does not encounter them. However, the description mentions Veselí nad Lužnicí, which is missing in the default description.

Also, as a direct consequence of the enlarged radius, the number of features around each segment increases, which leads to a prolonged time necessary for their fetching.

Chapter 6
User testing

To evaluate user perceptions of our annotator output, we conducted small-scale user testing. This involved gathering feedback from participants to understand their experiences and opinions on the quality and relevance of the annotations.

6.1 Setup and methodics

We collaborated with nine participants (labeled `A - I` to preserve anonymity), who consider themselves occasional cyclists. Each tester described their interests in order to obtain personalized trip descriptions, as shown in Table 6.1. We assigned each participant a trip from our example trips [5.1] and generated a personalized trip description for each participant. Additionally, they were provided the reference book [22] description regarding the same trip, translated into English. Importantly, the participants were not aware of the origin of the obtained texts.

Tester	Trip	Preference
A	1	places with a nice view, especially places to drink with a nice view
B	1	trains, scenic viewpoints, rock formations
C	1	lookout towers, viewpoints, natural wonders
D	2	observation towers, public transport
E	2	I want to hear about refreshment options, especially pubs.
F	2	I would like to grab a beer or a coffee
G	3	archeologic sites, memorials, interesting architecture
H	3	railway history
I	3	I like to admire majestic churches.

Table 6.1. Participant interests and assigned trips

The participants were then asked to fill in two short questionnaires, one for each description.

6.2 Questionnaire

The questionnaire consists of 8 statements and 2 open questions. The statements are rated on a 7-point Likert-like scale [25], ranging from `Strongly disagree` to `Strongly agree`. The statements are structured so that higher scores indicate more favorable results.

The statements *(with abbreviations for identification and clarity)*:

1. The description includes all the essential information needed for the trip. (*essentials*)
2. The description does not contain redundant or irrelevant information. (*redundancy*)
3. I would be interested in visiting all the places mentioned in the description. (*interests*)
4. The description gives a positive impression of the trip. (*positivity*)
5. The author wrote the article based on personal experience on the trip. (*experience*)
6. The description sounds like it is written by a professional cycling journalist. (*professionality*)
7. The language used sounds natural and appropriate for a cycling trip description. (*language*)
8. I like this cycling trip description overall. (*overall*)

The open questions:

- What do you enjoy about the description?
- What could be improved, or is missing in the description?

6.3 Quantitative evaluation

We summarize the obtained statement ratings for both the annotations and the book descriptions in the following tables. Responses are enumerated on a scale from 1 (`Strongly disagree`) to 7 (`Strongly agree`).

Statement	A	B	C	D	E	F	G	H	I	AVG	VAR
essentials	7	7	5	5	6	7	7	7	6	6.33	0.67
redundancy	5	2	5	1	3	6	5	5	4	4.00	2.44
interests	6	2	7	3	4	5	3	6	7	4.78	3.06
positivity	6	6	4	3	7	5	7	6	7	5.67	1.78
experience	4	6	4	2	3	4	1	4	6	3.78	2.40
professionality	7	6	3	1	4	5	7	5	4	4.67	3.33
language	6	5	5	3	5	5	7	7	5	5.33	1.33
overall	7	6	5	3	6	6	7	6	7	5.89	1.43

Table 6.2. Ratings of the descriptions generated by the annotator

Based on the results, the participants think the annotator provides all essential information. They also agree that the annotator gives a positive impression of the trip, and like the description overall. They are also positive about the language used. The participants do not agree upon whether they would like to visit all the places mentioned. Also, there are mixed opinions regarding redundancy of the information. They also could not agree upon the professional sounding of the text.

The participants did not think the description is written based on personal experience.

We also collected the statistics of the book description ratings to compare them to our annotator.

Statement	A	B	C	D	E	F	G	H	I	AVG	VAR
essentials	5	6	5	7	6	6	7	7	2	5.67	2.22
redundancy	5	5	5	7	2	6	2	2	2	4.00	3.56
interests	6	7	7	5	3	3	3	4	5	4.78	2.40
positivity	5	5	7	6	6	7	6	6	7	6.11	0.54
experience	4	6	6	7	4	7	7	6	7	6.00	1.33
professionality	2	5	6	6	5	6	2	6	4	4.67	2.44
language	5	5	6	6	5	4	3	7	3	4.89	1.65
overall	5	5	7	6	6	3	7	5	5	5.44	1.36

Table 6.3. Ratings of the reference book descriptions

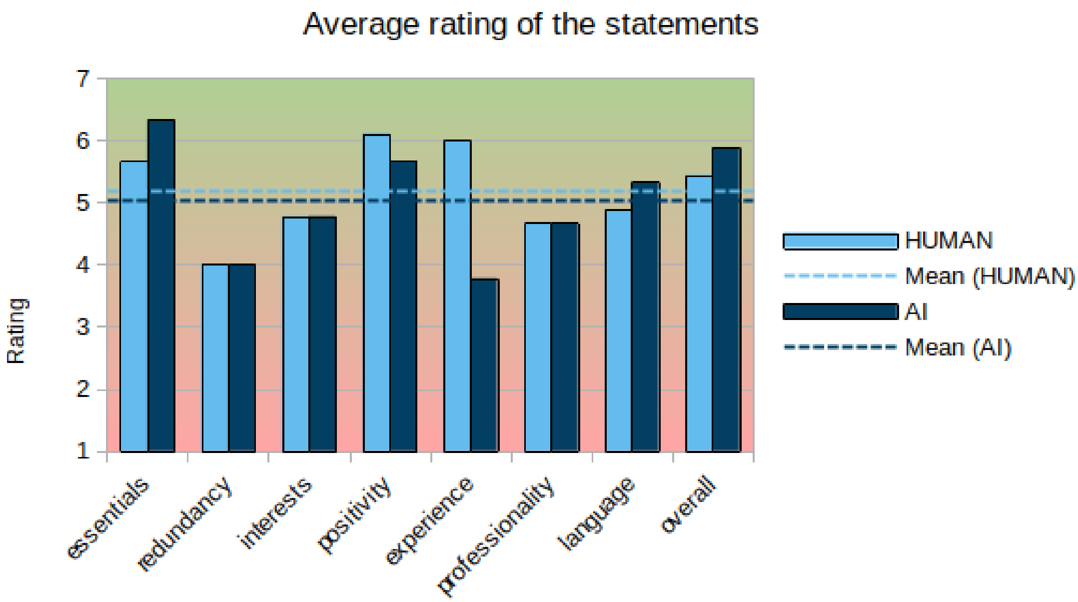

Figure 6.1. Quantitative analysis of user testing results

The graph illustrating the comparison suggests that the descriptions by the annotator can be compared to human-written descriptions. While the mean value is slightly higher for human-written descriptions, the average overall rating slightly favors the annotator. This occurrence is probably due to the fact that every attribute does not contribute to the overall rating equally. We further investigate this phenomenon in Section 6.5.

The only attribute that clearly favored the book descriptions was `experience` - the annotator did not make the participants think the annotations were based on personal experience.

The ratings were surprisingly similar in most of the attributes, however, the relatively high variations in some attributes in both the book and the annotator ratings suggest that the participants are rather biased in their opinions and there is no universal cycling trip description that appeals to all.

6.4 Qualitative evaluation

Open questions can provide valuable insights, although the analysis is more complex. This section summarizes the findings derived from the answers to the open questions. The specific answers can be found in Appendix B.

- **What do you enjoy about the description?**
  The participants often appreciate that the description is detailed and informative, providing essential information about the trip's length, checkpoints, and attractions. They pinpoint that the descriptions match their interests. They find the descriptions of the environment and interesting landmarks along the route appealing.

- **What could be improved, or is missing in the description?**
  Some participants find the description lengthy and tedious to read, almost robotic, wishing for a more concise version. They also question the relevance of certain attractions mentioned along the route, such as golf courses, which may not be interesting to all cyclists. Additionally, some find the precise numerical data excessive and distracting. Some participants were calling for more detailed terrain descriptions.

The opinions on the descriptions vary widely. While some participants find the descriptions comprehensive and engaging, others find them overly verbose and sometimes even lacking in relevance. There is a notable divide regarding the length and the level of detail provided.

6.5 Attribute analysis

To understand the relationships between the opinions on various description attributes, we computed a correlation matrix of their ratings based on both the annotation and the book description ratings.

	essentials	redundancy	interests	positivity	experience	professionality	language	overall
essentials	1.00							
redundancy	0.26	1.00						
interests	-0.24	0.50	1.00					
positivity	0.04	0.12	-0.07	1.00				
experience	-0.13	0.05	0.13	0.34	1.00			
professionality	0.42	0.42	-0.03	0.57	0.12	1.00		
language	0.54	0.44	0.26	0.25	-0.25	0.68	1.00	
overall	0.37	0.13	0.21	0.48	-0.02	0.35	0.43	1.00

Figure 6.2. Correlations of description attribute ratings

The matrix suggests that the `positivity` attribute of the description was the most influential for our participants when rating the overall impression. Other significant attributes included the `language` and `essentials` - the inclusion of all essential information.

Interestingly, whether the trip description was based on personal experience did not impact the overall impression at all.

The fact that the `interests` and `redundancy` factors partially correlate suggests that people are interested in visiting all the places if they are not overwhelmed by abundant tips.

The matrix also shows an association of `language` and `professionality` and `positivity` and `professionality`. That could mean that the participants estimated the author's professionality based on appropriate language and the positive sound of the description.

Mysterious is the increased correlation between `language` and `essentials`, which we cannot explain.

Chapter 7
Discussion

We have evaluated various aspects of our annotator and ran a user testing. This section discusses and explains the annotator's capabilities with the support of our findings.

7.1 Fulfillment of proposed requirements

In Section 3.1, we proposed the desired characteristics of the annotations. Subsequently, we implemented the annotator and thoroughly evaluated it. In this section, we discuss whether the defined requirements have been met.

7.1.1 Technical details

Our annotator correctly provides the overall trip characteristics regarding distance and elevation, with only minor discrepancies caused by the map data imperfections. The accuracy testing results in Section 5.3.3 revealed that the annotator correctly identifies and mentions significant hills on the route (94 %). It also successfully mentions localities and significant settlements (81 %), facilitating the orientation and providing geographic context.

The technical details provided by the annotator are precise and reliable. Additionally, according to the questionnaire responses provided in Section 6.3, the users agreed that the annotator provided all the essential information needed for the trip.

7.1.2 Environmental details and points of interests

The annotator analyzes the surroundings of the route and provides precise descriptions, which are appreciated by the users, as stated in Section 6.4. The success of the annotator is limited when trying to identify the points of interest. According to Section 5.3.3, it identified only 47 %, respectively 56 % of possible points of interest. However, the annotator is able to personalize the choice of points of interest based on the user preference, which is appreciated by the users and partially outweighs this weakness.

Both the environmental details and point of description mentions are considered satisfactory.

7.1.3 Description quality

The users perceive that the annotator generates highly precise and descriptive annotations. But sometimes, the descriptions generated by our annotator sound robotic due to a lack of storytelling elements and occasional unnatural word collocations, as found in Section 6.4.

Several issues have been identified that affect the quality of the descriptions produced by the language model. Firstly, the language model occasionally fails to strictly adhere to the provided instructions, which can lower the overall quality of the output. For instance, despite the prompt with an explicit instruction (mentioned in

Section 4.4.12) to round numbers, the annotator sometimes fails to do so. Moreover, the model has a tendency to summarize the description towards the end, sometimes skipping important information, even though it is instructed to maintain consistency in covering the entire route.

These shortcomings highlight the most complex challenge of this work: balancing the desired tone of voice with strict adherence to the instructions. While the current implementation achieves precision and descriptiveness, there is a compromise between maintaining a natural, engaging tone and strictly adhering to detailed guidelines.

Since the analysis in Section 6.5 suggests that the perceived positivity, followed by language, is the most important aspect of the user rating, further effort should be focused on improving these aspects.

7.1.4 Overall quality

The annotator mostly meets the defined requirements, though there are some exceptions. These exceptions stem from both the nature of the annotator (as discussed in Section 7.3) and the limitations of the language model's capabilities, eventually interconnected with the quality of the prompts.

7.2 Performance limitations

There are still some factors caused by our specific implementation that prevent the annotator from potential commercial usage.

- **Time complexity**

  According to our findings in Section 5.5, the annotator sometimes takes up to three minutes to create the annotation, which is an immense amount of time. It is caused mainly by the usage of multiple remote services, processing a large amount of data several times, and a purely sequential approach.

  To address this problem, we could:

  - introduce parallelization of some tasks, e. g. processing the user preference while fetching the data or describing multiple segments in parallel
  - localize the geographic data
  - use pre-trained language models running locally

- **Financial requirements**

  The financial requirements are also high. The price evaluation in Section 5.4 revealed that the usage of the annotator can cost up to $0.20 per annotation.

  Possible adjustments leading to lower the price include:

  - using free language models, but with a possible decrease in the output quality
  - using free service providing elevation data

7.3 Key differences between the annotator and human-written descriptions

This section explores the distinct nature of trip descriptions created by human authors versus those generated by the AI-powered annotator and the consequences of the differences.

- **Information source**:
  Human-written trip descriptions are based mainly on human experience, as the authors cycled the trip themselves and did not rely on any other source of geographic data. The annotator relies on open geographic data, which can be incomplete, or even incorrect. It can lead to misinforming the user. Also, even though some features are described correctly, slight nuances within the categories (e.g., asphalt surface, but broken) might affect the cyclist greatly but cannot be derived from the data.

- **Currency of information**
  The human-written descriptions remain up-to-date for a limited time only and can get outdated soon, for example, with major road reconstructions, etc. Our annotator generates the description based on the current state of the map data and, therefore, is able to reflect these changes as soon as they are updated in the map source.

- **Perception**
  A person does not choose the points of interest based on the distance but on their visibility. Our annotator is not able to see and, therefore, cannot distinguish between visible and not visible features. A feature might not be visible (covered by another building or even a few streets away) and still be contained in the description. Or, if a feature is clearly visible but outside the given radius (e. g. a mountain on the horizon), the annotator would ignore it.
  The lack of other senses might also have an effect - our annotator cannot hear a noisy road or smell an unpleasant agricultural odor. Increased probabilities of these occurrences could be estimated based on the neighborhood, but the predictions are not always successful.

7.4 Further steps

Some other functionalities might be implemented to enhance the annotations.

- **Advanced user preference settings**
  The testers had different expectations about the level of informativeness they desired from the trip descriptions. Some prioritized precise numerical details, while others considered them unnecessary and sought descriptions that would rather motivate them to undertake the trip. Also, some were satisfied with the length, while others would prefer a shorter description. Considering these preferences, we could adjust the annotations to meet each user's specific needs more effectively.

- **Wikipedia information association**
  We use Wikipedia solely to assess salience and provide links to relevant Wikipedia pages when available. However, we could expand our use of Wikipedia to offer the users additional information and interesting facts, making the annotations more informative.

- **Points of interest rating**
  The annotator lacks access to ratings for points of interest. Incorporating such information could significantly improve its recommendations, make the annotator more trustworthy, and enhance user satisfaction.

- **Visual information incorporation**

  Recently, the integration of visual and textual data has gained significant attention in research, leading to the emergence of advanced language models capable of processing both types of information. By incorporating visual data into our cycling trip annotations, we could enhance their accuracy and appeal, providing users with richer and more attractive descriptions.

Chapter 8
Conclusion

We have demonstrated that large language models, when provided with accurate data, can produce high-quality descriptions of cycling trips. Our development of an annotator capable of generating such descriptions illustrates this potential. Despite the annotator's inability to physically travel the route, user testing suggests that the generated descriptions are not only comparable to but, in some aspects, even surpass those written by humans who have experienced the journey themselves.

Practical applications of our annotator include its possible integration into travel planning websites, interactive tourist guides, and cycling route planning tools, providing automation and leveraging user experience by generating personalized, detailed, and up-to-date content.

References

[1] Markus Drager, and Alexander Koller. Generation of landmark-based navigation instructions from open-source data. 2012.

[2] Petr Michalík. *Generating context-enriched instructions for bicycle navigation*. 2022.
http://hdl.handle.net/10467/101408.

[3] Raphael Schumann, and Stefan Riezler. *Generating Landmark Navigation Instructions from Maps as a Graph-to-Text Problem*. 2021.

[4] Paul Downward, and Les Lumsdon. The development of recreational cycle routes: An evaluation of user needs. *Managing Leisure*. 2001, 6 DOI 10.1080/13606710010026368.

[5] Yechezkel Israeli, and Dalit Gasul. FROM SERIOUS LEISURE TO CYCLING TOURISM – THE CASE OF MOUNTAIN BIKING. *Acta turistica*. 2019, 31 DOI 10.22598/at/2019.31.2.179.

[6] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. *A Comprehensive Overview of Large Language Models*. 2024.

[7] Samuel R. Bowman. *Eight Things to Know about Large Language Models*. 2023.

[8] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, and others. GPT-4 Technical Report. 2024.

[9] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of Artificial General Intelligence: Early experiments with GPT-4. 2023.

[10] Yutaro Yamada, Yihan Bao, Andrew K. Lampinen, Jungo Kasai, and Ilker Yildirim. Evaluating Spatial Understanding of Large Language Models. 2024.

[11] Mohamed Aghzal, Erion Plaku, and Ziyu Yao. Can Large Language Models be Good Path Planners? A Benchmark and Investigation on Spatial-temporal Reasoning. 2024.

[12] Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. PlanBench: An Extensible Benchmark for Evaluating Large Language Models on Planning and Reasoning about Change. 2023.

[13] Rohin Manvi, Samar Khanna, Gengchen Mai, Marshall Burke, David Lobell, and Stefano Ermon. GeoLLM: Extracting Geospatial Knowledge from Large Language Models. 2024.

[14] Ashley Fernandez, and Swaraj Dube. Core Building Blocks: Next Gen Geo Spatial GPT Application. 2023.

[15] Gengchen Mai, Chris Cundy, Kristy Choi, Yingjie Hu, Ni Lao, and Stefano Ermon. Towards a foundation model for geospatial artificial intelligence (vision paper). 2022. DOI 10.1145/3557915.3561043.

[16] Zhenlong Li, and Huan Ning. Autonomous GIS: the next-generation AI-powered GIS. 2023.

[17] the free encyclopedia Wikipedia. *Azimuth*. `https://en.wikipedia.org/wiki/Azimuth`. 2024. Accessed: 2024-05-23.

[18] Geoff Boeing. *Modeling and Analyzing Urban Networks and Amenities with OSMnx*. 2024. `https://geoffboeing.com/publications/osmnx-paper/`.

[19] K. Jordahl, J. Van den Bossche, and others. *geopandas/geopandas: v0.8.1*. 2020. `https://doi.org/10.5281/zenodo.3946761`.

[20] Mordechai Haklay, and Patrick Weber. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing*. 2008, 7 (4), 12-18. DOI 10.1109/MPRV.2008.80.

[21] OpenStreetMap Wiki. *Map features — OpenStreetMap Wiki*. 2023. `https://wiki.openstreetmap.org/w/index.php?title=Map_features&oldid=2543361`. Online; accessed 19-May-2024.

[22] Lukáš Drbohlav. *Srdcem na kole Českem*. 2023. `https://www.cycling-routes.cz/`.

[23] OpenAI FAQ. *What are tokens and how to count them?* `https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them`. 2024. Accessed: 2024-05-21.

[24] Matthew Renze, and Erhan Guven. *The Effect of Sampling Temperature on Problem Solving in Large Language Models*. 2024.

[25] Ankur Joshi, Saket Kale, Satish Chandel, and D. K. Pal. Likert Scale: Explored and Explained. *Current Journal of Applied Science and Technology*. 2015, 7 396–403. DOI 10.9734/BJAST/2015/14975.

Appendix A
Accuracy evaluation tables

This appendix provides complete tables used for manual inspection of example trips, according to the method described in Section 5.3.

A.1 Example trip 1

Hill	Elevation gain	Present in annotation
near Větrovy	100	yes
before Stádlecký most	-120	yes
after Stádlecký most	80	yes
near Hodětín	80	yes
near Nedvědice	80	no
near Obora	-80	no
near Maršov	90	yes
near Větrovy	-100	yes

Table A.1. Example trip 1 - significant hills.

Tourist attraction	Present after pre-processing	Present in annotation
Žižkovo náměstí	no	no
Kotnov castle	yes	yes
Stádlecký most	yes	yes
Zámek Bechyně	yes	yes
Golf Bechyně	yes	yes
Veselská blata	yes	yes
Obora brewery	no	no

Table A.2. Example trip 1 - tourist attractions

Settlement	Present after pre-processing	Present in annotation
Tábor	yes	yes
Horky	yes	yes
Větrovy	yes	yes
Lom	yes	yes
Maršov	yes	yes
Malšice	yes	yes
Dobřejice	yes	yes
Stádlec	yes	yes
Staré Sedlo	yes	yes
Rataje	yes	no
Radětice	yes	yes
Bechyně	yes	yes
Hodonice	yes	no
Březnice	yes	no
Blatec	yes	no
Hodětín	yes	yes
Komárov	yes	no
Klečaty	yes	yes
Zálší	yes	yes
Mažice	yes	no
Sviny	yes	yes
Žíšov	yes	yes
Borkovice	yes	yes
Veselí nad Lužnicí	no	no
Vesce	yes	yes
Mokrá	yes	no
Nedvědice	no	no
Debrník	yes	no
Vyhnanice	yes	no
Hlavatce	no	no
Želeč	yes	no
Obora	yes	no

Table A.3. Example trip 1 - settlements.

A.2 Example trip 2

Hill	Elevation gain	Present in annotation
after Deštné v O. h.	300	yes
before Orlické Záhoří	-300	yes
near Mezivrší	250	yes
before Říčky v O. h.	-300	yes
after Říčky v O. h.	150	no
near Zdobnice	-150	yes
to Velká Deštná	550	yes
near Deštné v O. h.	-550	yes

Table A.4. Example trip 2 - significant hills.

Tourist attraction	Present after pre-processing	Present in annotation
Komáří vrch	yes	yes
Stonehenge miniature	no	no
Velká Deštná lookout tower	no	no
Jelení lázeň	yes	yes

Table A.5. Example trip 2 - tourist attractions.

Settlement	Present after pre-processing	Present in annotation
Jedlová v O. h. (part of Deštné)	yes	yes
Orlické Záhoří	yes	yes
Říčky v O. h.	yes	yes
Zdobnice	yes	yes

Table A.6. Example trip 2 - settlements.

A.3 Example trip 3

Tourist attraction	Present after pre-processing	Present in annotation
Sokolov Chateau	yes	yes
Park Sokolov	no	no
Husovy sady park	yes	yes
Amfiteátr Loket	no	no
Loket Castle	yes	yes
View of Loket	no	no

Table A.7. Example trip 3 - tourist attractions.

Hill	Elevation gain	Present in annotation
after Dasnice	50	yes
before Hlavno	-50	yes
in Sokolov	60	no
before Loket	-80	no
before Královské Poříčí	90	yes
to Těšovice	-60	yes
to Libavské Údolí	50	no

Table A.8. Example trip 3 - significant hills.

Settlement	Present after pre-processing	Present in annotation
Cheb	yes	yes
Jindřichov	yes	no
Chocovice	yes	yes
Nebanice	yes	yes
Mostov	no	no
Kynšperk n. O.	no	no
Dasnice	yes	yes
Hlavno	yes	yes
Citice	yes	yes
Sokolov	yes	no
Staré Sedlo	yes	no
Loket	yes	yes
Královské Poříčí	yes	yes
Tisová	no	no
Šabina	yes	yes
Libavské Údolí	yes	yes
Chotíkov	yes	yes
Loužek	yes	no
Potočiště	yes	no
Chvoječná	yes	no

Table A.9. Example trip 3 - settlements.

Appendix B
User opinions

We provide full reactions to open questions in our questionnaire while keeping the anonymity of the respondents.

What do you enjoy about the description?

- **A**: "This is a thorough and detailed description of a cycling trip, which offers all the needed information, such as elevation changes, distances crossed between checkpoints, and environmental details. The tone is much more direct and professional than in the other trip description, which makes it sound like what one would read in a cycling journal when comparing it to the other description.

  Great for both locals and foreigners alike.

  I like that it describes the environment in a way that highlights all aspects of the trip, e.g. "A steep climb over a narrow, unpaved path follows, surrounded by the tranquility of forested areas and a nearby wetland." Here you get all the necessary details: it'll be a rough climb, but at least it's very beautiful.

  The checkpoints are explained well, which makes it easier to wrap your head around this whole trip.

  Overall, it's a well thought out description of a cycling trip, and it lets everyone know what they're signing up for."

- **B**: "I like mentioning the landscape around the path."

- **C**: "Description contains many interesting places on the route and in the surrounding areas."

- **D**: "It gives me a lot of information about places I normally would not notice along the route."

- **E**: "The spirit of the description is nice."

- **F**: "Specific information and place suggestions, exact kilometer counts."

- **G**: "It is very detailed and thorough; I do not miss any needed information. All the relevant details are mentioned, and I have a good understanding of the trip."

- **H**: "Specific numbers of distance and elevation, railway mentions."

- **I**: "I like that the description often focuses on places that I would really like to visit. I also like that the description contains precise numbers."

What could be improved, or is missing in the description?

- **A**: “Not much to improve, it answers most questions I had.”
- **B**: “I doubt that many cyclists will visit some attractions mentioned during the trip, for example a golf course. I miss more details about the route itself.”
- **C**: “I miss places for refreshments and a mention of larger towns around the route. On the other hand, I wouldn’t need so much precise numerical data.”
- **D**: “While the route includes many ascents, the description features interesting elements such as bunkers, stones, and meadows, which are worth stopping to inspect more closely. I cannot imagine riding this way and stopping by to see everything. Furthermore, the description has a technocratic tone, feeling more like Wikipedia and less like storytelling.”
- **E**: “The length. I don’t want to spend this much time reading.”
- **F**: “Sounds a lot like navigation, could be described in a more exciting manner.”
- **G**: “The amount of detail is huge, maybe there is even more than enough information needed for me to create an opinion on the trip. It sometimes sounded a little bit robotic, with strange word collocations or meanings.”
- **H**: “Mention of Slavkovská dráha seems to be a little out of place (a bit too early in the text). Information about distance beyond the decimal point might be redundant (and making the text a bit harder to read). There is no information about the road quality.”
- **I**: “Even though the text is practical, it is not as captivating as the other one.”

Appendix C
Source code

The source code of the annotator, along with some example files, is located on GitLab:

https://gitlab.fel.cvut.cz/pejsomic/automated-generation-of-textual-annotations-for-bicycle-routes-using-large-scale-language-models.