

Coding Guide

In the attached Excel Document, you will find four separate sheets, corresponding to the four dimensions: **Questions** (143 data rows), **Preferences** (38 data rows), **Cues** (134 data rows) and **Impression Factors** (106 data rows).

In each sheet, please add in the **Category** column, your annotation for the category (only one annotation per data point). Please follow the descriptions in the tables and the instructions to choose the appropriate category.

Question Categories:

Category	Description	Example
HowTo	Questions that ask how to achieve a particular functionality. This also includes how to navigate the development environment, for e.g. to build an application. This does not include how to fix errors (code Debug instead).	<i>How to convert a float to integer in Java?</i>
Debug	Questions related to why an error occurs or how to fix it. This includes trying to understand what an error message means.	<i>What does "InvalidArgumentException" mean?</i>
Conceptual	Related to the conceptual understanding of technology and its components. This includes what is a component is, what is the difference between some components, and how multiple technologies interact.	<i>How is machine learning used in recommender systems?</i>
Resource	Questions related to finding a useful resource for learning or determining what information a particular resource or kind of resource provides. The question includes looking for a particular resource, or a kind of resource. It also includes questions about what information a particular resource provides about the technology.	<i>What kind of information does Wikipedia provide about JavaFX?</i>
Miscellaneous	Questions that do not fit into any other category.	-

Preference Categories:

Category	Description	Example
Resource Type	Specifying the specific resource or kind of resource needed.	<i>Looking for a Stack Overflow post</i>
Information Style	Specifying that the information should be structured or presented in a particular manner. This includes organization, level of granularity, depth, recency of information, and if code or data examples are wanted.	<i>Looking for a short code example</i>

Cue Categories:

Category	Description	Example
Recommended	The resource is chosen because it is among one of the top four search results, is featured by the search engine, has high number of upvotes, claps, likes, etc., is explicitly recommended by users, or mentioned in another resource.	<i>Has a large number of upvotes</i>
Familiarity	The participant has some familiarity with the resource or the content that it contains, or has used the resource in the past and has had a positive impression. This category applies if the participant explicitly mentions that they have used this resource in the past.	<i>Have used this resource before, and it was useful</i>
Noted key-words	The search result title/snippet or resource contains keywords that were present in the query, or are relevant to their question.	<i>Google snippet contains words that describe the exact question being searched</i>
Authoritative	The source of the resource seems to be a credible or reputed authority. In a majority of cases this is because the resource is from the developers of the technology themselves.	<i>Is created by Sun Microsystems</i>
Exact requirements	The resource is the exact resource wanted or generally preferred, or contains the exact characteristics needed. For e.g., if the resource is up to date, is in the correct domain, is the correct level of granularity, or its information is presented, styled, or formatted in a manner that is needed or preferred. If it is mentioned that this resource is preferred <i>because</i> it was useful in the past, code Familiarity instead.	<i>Resource usually contains detailed information</i>
Miscellaneous	Cues that do not fit into any other category.	-

Impression Factor Categories:

Impression Factor refers to *what* the comment is assessing or providing a judgement upon. That is, the Impression Factor is what is *used* to evaluate the resource. Please read through the following completely to decide which category to use.

The impressions in the data set apply to a *learning resource*. In this study, a resource can be one of:

- * Any document (web page, pdf, or other);
- * A thread on a forum (both questions and answers);
- * A search results page (the search result links constitute the "content")

Code each impression as **structure**, **content**, or **pertinence**. The general principles for coding are as follow:

- * If the comment can help a contributor improve the *organization* of the resource, annotate **Structure**.
- * If the comment can help a contributor improve the nature or quality of the information contained, annotate **Content**.
- * If the impression is too personal or vague to help guide improvements to the resource, annotate **Pertinence**.

Use the following coding rules for specific guidance on interpreting the above principles. In case of confusion, please use the following priority and annotate accordingly: (1) Structure, (2) Content, (3) Pertinence.

Structure

The **explicit organization** of a resource.

- Includes the *length, amount of text, inclusion or exclusions of certain features* (code examples, practice exercises, videos, links), how the information is *organized* (e.g., step by step), or other *explicit* decisions on how to structure the document.
- For forums, inclusion or exclusion of answers to questions should be coded as Content, because it was not an explicit resource design decision.
- If the comment is about both the *inclusion* of a type of content and its quality, choose structure (e.g., "Has detailed practice questions", "has recent code examples"). If the comment is **not** about the inclusion of a certain feature, but focuses on its quality *only*, use Content. For example, "has useful code examples", should be coded as "Structure", whereas "the code examples have errors" should be coded as "Content".

- If the resource is a list of search results, use Content or Pertinence as applicable (not Structure).

Content

The **quality of the information** contained in the resource, and clearly a function of the creation of the resource (explicitly or implicitly) or the search engine algorithms (in cases where the resource is a list of search results).

- Includes *amount of information, presence of errors or other quality problems (e.g., broken links), choice of topics, recency, authoritativeness, nature and quality of search results.*
- Include *specific* comments about the *nature of the information, including level of detail, knowledge required, target audience.*
- Include *general* comments about the nature of the information if they are clearly only about the content (e.g., "good tutorial").
- *Specific* comments about the quality of the content should be coded as Content, even if they are subjective. For example, comments about clarity or interestingness. If any aspect of the impression is concrete enough to improve the content, use Content. Otherwise, use Pertinence. For example, "does not contain enough Windows-specific information" should be coded as "Content", whereas "does not support my OS" should be coded as "Pertinence".
- Presence or absence answers to posts on forums in absolute terms. If the impression is about "relevant" answers, use Pertinence.
- If the comment is just that some aspect of the content (e.g., topic, programming language), is not the one wanted, use Pertinence instead.
- If the comment is about both content and pertinence, choose content (e.g., "the answer was vague and not helpful")

Pertinence

The factor is only about the **general relevance of the resource to the user**, independently of the structure or content of the resource. In other words, impressions that don't say anything reliable about the resource itself, except that it's not a good fit for the user, or other reasons.

- Includes *general* comments such as that the resource *does not include the information needed, or relevant answers*, etc.
- Includes comments about the quality of the content that are too general to be actionable (e.g., "useful", or "not helpful")