Explainable Mapper: Charting LLM Embedding Spaces using Perturbation-Based Exploration, Explanation, and Verification Agents Supplementary Materials

Xinyuan Yan, Rita Sevastjanova, Sinie van der Ben, Mennatallah El-Assady, and Bei Wang

In this supplement, we first present additional examples of ball mapper (Apendix A). We then provide further use cases of the fine-tuned BERT-base model (Apendix B). Next, we introduce three additional models—BERT-tiny, RoBERTa, and DeBERTa—which, together with BERT-base, are fine-tuned on two linguistic tasks. We analyze and compare their mapper graphs using *Explainable Mapper* (Apendix C).

We also present a multiscale mapper example (Apendix D), followed by an episode table and concrete examples of expert–agent collaboration (Apendix E). Finally, we include a reproducibility checklist for *Explainable Mapper*, covering all prompt templates and prompt engineering examples (Apendix F), data processing details (Apendix G), implementation details, storage, and computation time (Apendix H), model specifications and fine-tuning procedures (Apendix I), and alternative robustness metrics (Apendix J).

A Examples for the Ball Mapper

In this section, we present several examples of *Explainable Mapper* based on the ball mapper. Figure 11 provides an overview of the ball mapper applied to the final layer of a fine-tuned BERT model, where we use the same ε value as in the classical mapper to define the geometric neighborhood.

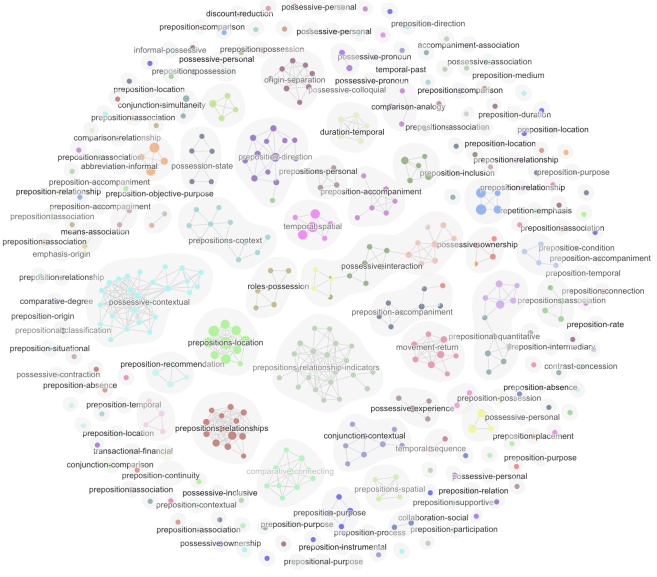

Fig. 11: Ball mapper graph of the final layer of the fine-tuned BERT model, with overlaid component annotations.

- *Xinyuan Yan and Bei Wang are with the University of Utah. E-mails: xinyuan.yan@utah.edu, beiwang@sci.utah.edu*
- *Rita Sevastjanova, Sinie van der Ben, and Mennatallah El-Assady are with ETH Zürich. E-mails: rita.sevastjanova@inf.ethz.ch, vandersi@ethz.ch, menna.elassady@ai.ethz.ch*

Component investigation. In the ball mapper, we observe that tokens are generally grouped into components based on their supersense-role labels, with more complex inner connections compared to the classical mapper. For example, Figure 12 (left) shows a component composed of tokens labeled "Time." The Component Explainer produces a highly consistent explanation, indicating that words such as *'in,' 'after,'* and *'before'* often precede markers of time events to convey temporal information. Similarly, as shown in Figure 12 (right), the words labeled "EndTime" form a single component composed of two nodes. To examine its internal structure, we apply the Edge Explainer to characterize the transition between the nodes. The explanation reveals a lexical shift from *'until'*, which denotes temporal boundaries (*e.g., "until I went home"*), to *'to'*, which indicates a time sequence (*e.g., "8 AM to 10 AM"*).

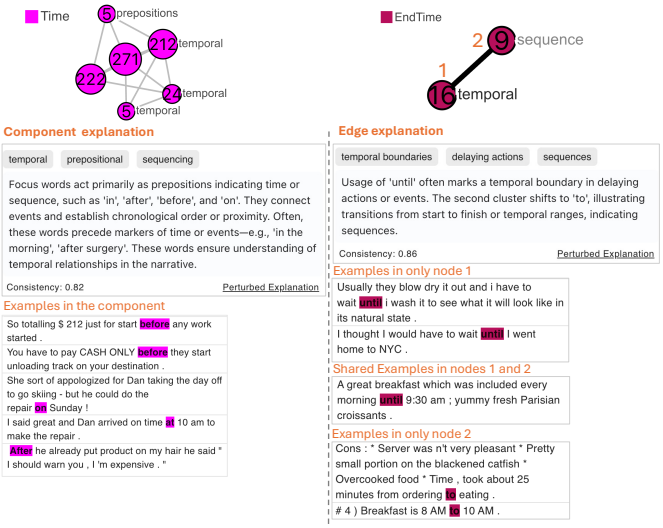

Fig. 12: A component (left) and an edge (right) in the ball mapper graph, with accompanying explanations.

Model confusion. In addition, we identify a component mixed with three labels of "Originator", "Agent", and "SocialRel" as illustrated in Figure 13, suggesting that the model has difficulty in distinguishing among these categories. To investigate this, we apply the Path Explainer to analyze the shortest path connecting nodes on opposite sides of the component. The explanation indicates a progression in word usage: from possessive pronouns like *'their'* and *'my'*, which signal ownership, to prepositions like *'from'* and *'by'*, which suggest interaction, and finally to *'with'*, which conveys collaboration. This linguistic transition aligns with the label shift from "Originator" to "Agent" to "SocialRel" along the path, indicating scenarios where the model exhibits confusion.

B Further Case Studies of the BERT Model

In this section, we present additional use cases of the fine-tuned BERT-base model (hereafter referred to as BERT for brevity) to complement the main paper.

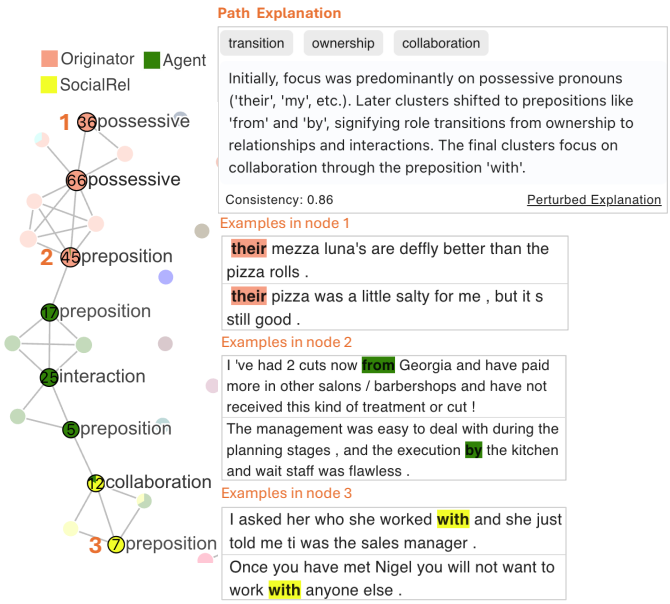

Fig. 13: A path in the ball mapper with a path explanation.

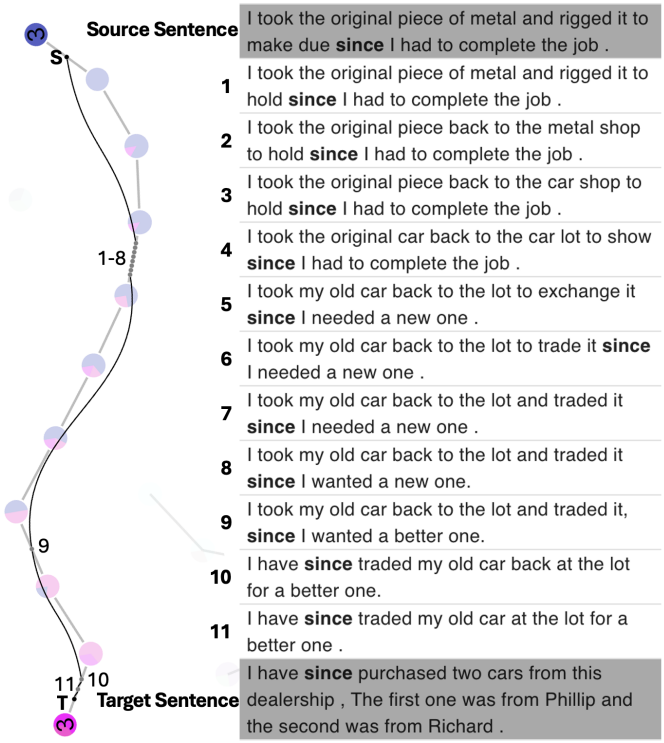

Fig. 14: The trajectory between two end nodes of the *'since'* path formed in layer 6.

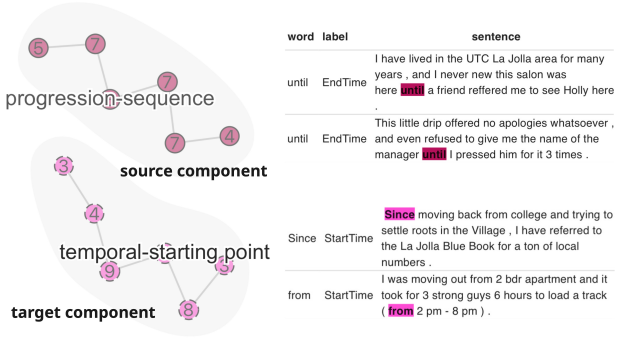

Fig. 15: Both components contain temporal prepositions but differ in focus: the source component emphasizes initiation and continuity, while the target component emphasizes culmination and time limits.

B.1 Trajectory Explanation Example

We present a trajectory explanation corresponding to the path shown in Figure 5 of the main paper. This trajectory also serves as a demonstration example in our expert study. The path is derived from layer 6 of the fine-tuned BERT model and illustrates that *'since'* is initially used in temporal contexts to indicate actions or events following a point in time. As the path progresses, its usage shifts toward causal and conditional meanings.

To further analyze this transition, we apply the Trajectory Explainer by constructing a trajectory between the two end nodes of the path. As shown in Figure 14, the trajectory largely follows the path: *'since'* conveys a causal meaning in sentences 1–9 and shifts to a temporal interpretation in sentences 10 and 11. The attachment positions of sentences 9 and 10 indicate a transition from a mixed mapper node to a node with purely temporal labels, further supporting the hypothesis of path evolution.

B.2 Temporal Preposition

Temporal prepositions (e.g., '*since*', '*from*', '*until*') indicate when an action or event occurs. As shown in Figure 15, in layer 12, tokens marking the start of an action form a distinct component from those marking its end. We use the Component Explainer to compare these components. The summary, validated by the Component Verifier, states that both components highlight temporal sequences but differ in focus: the source component emphasizes initiation and temporal continuity, whereas the target component emphasizes culmination and time limits.

B.3 *'As'* in Conjunction and Preposition

The token *'as'* can function as either a conjunction or a preposition, depending on its usage in a sentence. As a conjunction, it connects clauses or phrases, indicating a relationship between them. As a preposition, it introduces a noun or pronoun, establishing a relationship between that noun or pronoun and another element in the sentence.

In Figure 17, we illustrate how the fine-tuned BERT model captures these syntactic properties in its middle layers. In particular, at layer 5, the tokens separate into two distinct components. We examine their linguistic properties by annotating component nodes with descriptive keywords. These annotations reveal two distinct embedding patterns: one component primarily represents conjunction usage, while the other corresponds to role-identification contexts.

To further investigate this distinction, we apply the Component Explainer to compare the two components. The resulting explanations highlight their key differences. One component predominantly uses *'as'* in comparative constructions to indicate equal degree, frequency, or quantity (e.g., "*twice as much*"), whereas the other uses *'as'* to denote identity or role, often at the beginning of a sentence (e.g., "*As a nurse*"), emphasizing status or function. This distinction aligns with the functional role of prepositions.

The Component Verifier yields a high consistency score, as evidenced by the strong resemblance between the explanations of perturbed components and the original explanations.

B.4 Possessive Pronouns

Using the Node Explainer, we study how fine-tuned BERT learns possessive pronouns. We filter the pronoun '*my*' and show its occurrences in the mapper graph for layers 1 and 12. In layer 1, where embeddings are still noncontextualized, all tokens form a single component (see Figure 16). In layer 12, embeddings are contextualized for the classification task, and tokens split into multiple components grouped with related tokens. Using the Path Explainer on two nodes at opposite ends of a component reveals how encoded properties change. The explanation states: *focus words 'my,' 'your,' 'their,' 'his,' and 'our' consistently appear across clusters, expressing personal possession and*

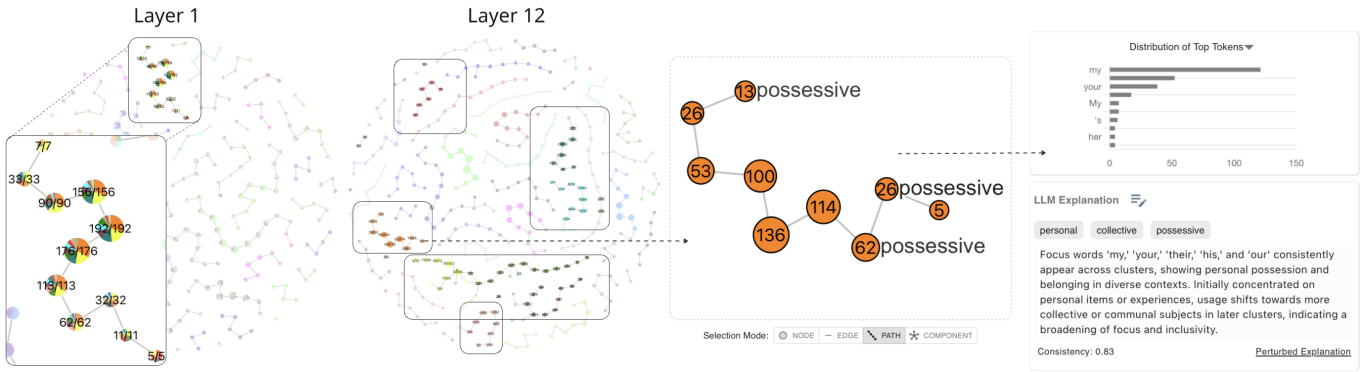

Fig. 16: Exploring occurrences of the token *'my'*. In layer 1, embeddings remain non-contextualized, and all tokens are grouped into a single component. By layer 12, the embeddings become contextualized for the downstream classification task. Node annotations highlight the shared property of the tokens—namely, that they are *possessive* pronouns. The Path Explainer provides additional insight into the semantic transition along the path, showing usage contexts ranging from *personal experience* to a *collective subject*.

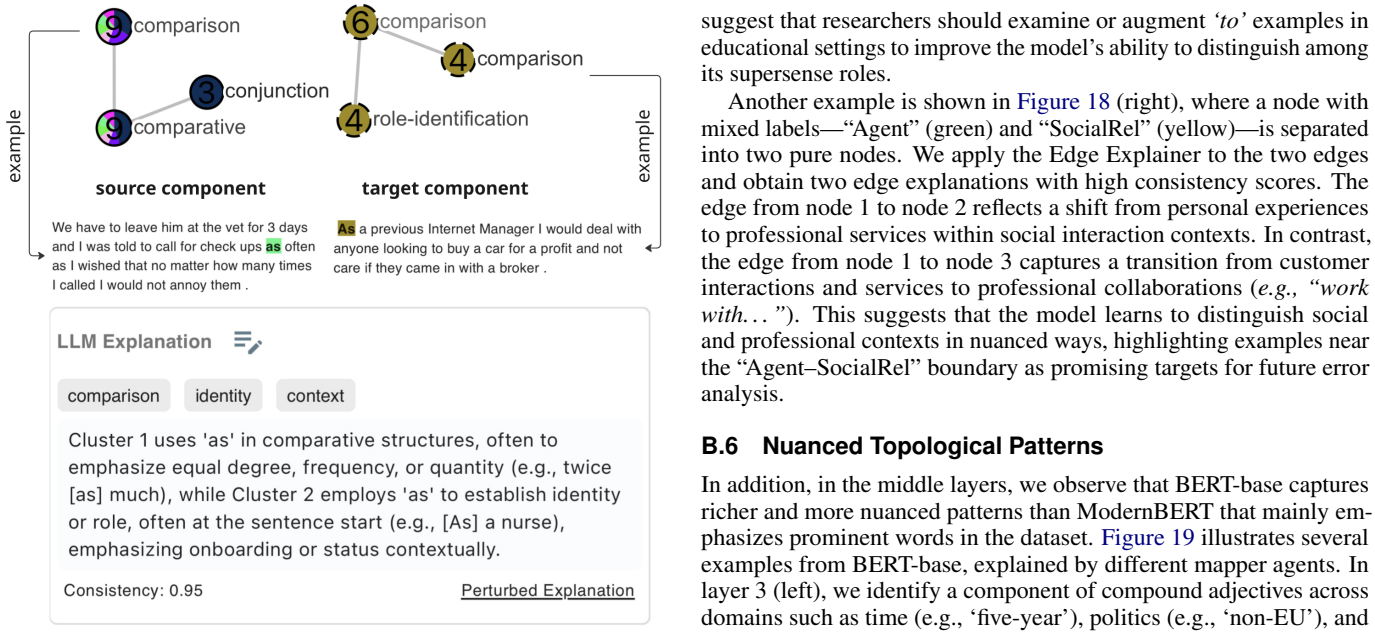

Fig. 17: Layer 5 of the fine-tuned BERT captures the two functions of the token '*as*', i.e., its role of conjunction (connecting clauses or phrases) and preposition (by showing a relationship between that noun/pronoun and another word in the sentence). The two functional roles get separated into two distinct connected components.

belonging in diverse contexts. Initially centered on personal items or experiences, usage shifts toward more collective or communal subjects in later clusters, indicating a broader focus and inclusivity. The Path Verifier yields a high consistency score (0.83), showing that explanations for perturbed examples along the path remain similar to the original explanation.

B.5 Model Confusion in Supersense Role Labels

In the final layer of a BERT model fine-tuned on the supersense-role task, most components in the mapper graph cluster by supersense-role labels. However, some nodes exhibit a mix of labels, indicating areas where the model is confused or uncertain. For example, along the path shown in Figure 18 (left), words labeled as "Goal" (node a) gradually transition to words labeled as "Organization" (node d). We utilize the Path Explainer and obtain an explanation with a high consistency score of 0.86. The explanation reveals that the path focuses on the token *'to'*, which initially appears in diverse contexts, then narrows to educational settings, and eventually stabilizes around academic institutions. This may suggest that the model struggles to differentiate *'to'* between labels "Goal" and "Organization" in educational contexts. These insights suggest that researchers should examine or augment *'to'* examples in educational settings to improve the model's ability to distinguish among its supersense roles.

Another example is shown in Figure 18 (right), where a node with mixed labels—"Agent" (green) and "SocialRel" (yellow)—is separated into two pure nodes. We apply the Edge Explainer to the two edges and obtain two edge explanations with high consistency scores. The edge from node 1 to node 2 reflects a shift from personal experiences to professional services within social interaction contexts. In contrast, the edge from node 1 to node 3 captures a transition from customer interactions and services to professional collaborations (*e.g., "work with…"*). This suggests that the model learns to distinguish social and professional contexts in nuanced ways, highlighting examples near the "Agent–SocialRel" boundary as promising targets for future error analysis.

B.6 Nuanced Topological Patterns

In addition, in the middle layers, we observe that BERT-base captures richer and more nuanced patterns than ModernBERT that mainly emphasizes prominent words in the dataset. Figure 19 illustrates several examples from BERT-base, explained by different mapper agents. In layer 3 (left), we identify a component of compound adjectives across domains such as time (e.g., 'five-year'), politics (e.g., 'non-EU'), and fractions (e.g., 'three-fifths'). In layer 6 (middle), month words (e.g., 'January', 'March') form a component, often appearing after the preposition 'in'. In layer 11, we find a subgraph of African countries. At a bifurcation within this subgraph, we compare two end nodes: one emphasizes financial contexts, while the other emphasizes political contexts. Such structured patterns are largely absent in ModernBERT or only appear in isolated nodes, potentially implying that ModernBERT encodes information in a more dispersed way with fewer locally coherent clusters.

B.7 Editing 'until' Trajectory

As shown in Figure 20, tokens labeled "Endtime" form two disconnected components, one of which is a single node (B). To investigate the transition between them, we construct a perturbation trajectory between nodes A and B using representative sentences containing the focus word *'until'*. The LLM generates eight intermediate sentences, whose embeddings are projected onto the mapper graph to form a trajectory. The trajectory switches between the two components between sentences 3 and 4, where the newly introduced phrase *"have to"* coincides with the transition. However, the change between sentences 2 and 3 is relatively large, replacing *"We waited…"* with *"We decided we would wait…"*, involving multiple edits (e.g., changes to the main clause and verb tense). To obtain a smoother transition, users can insert additional intermediate sentences between these two steps using the same prompt as trajectory generation. In this example, the inserted sentences reveal a more gradual transition from *"We waited…"* to *"We decided to wait…"*, resulting in a smoother edge transition to occur between the

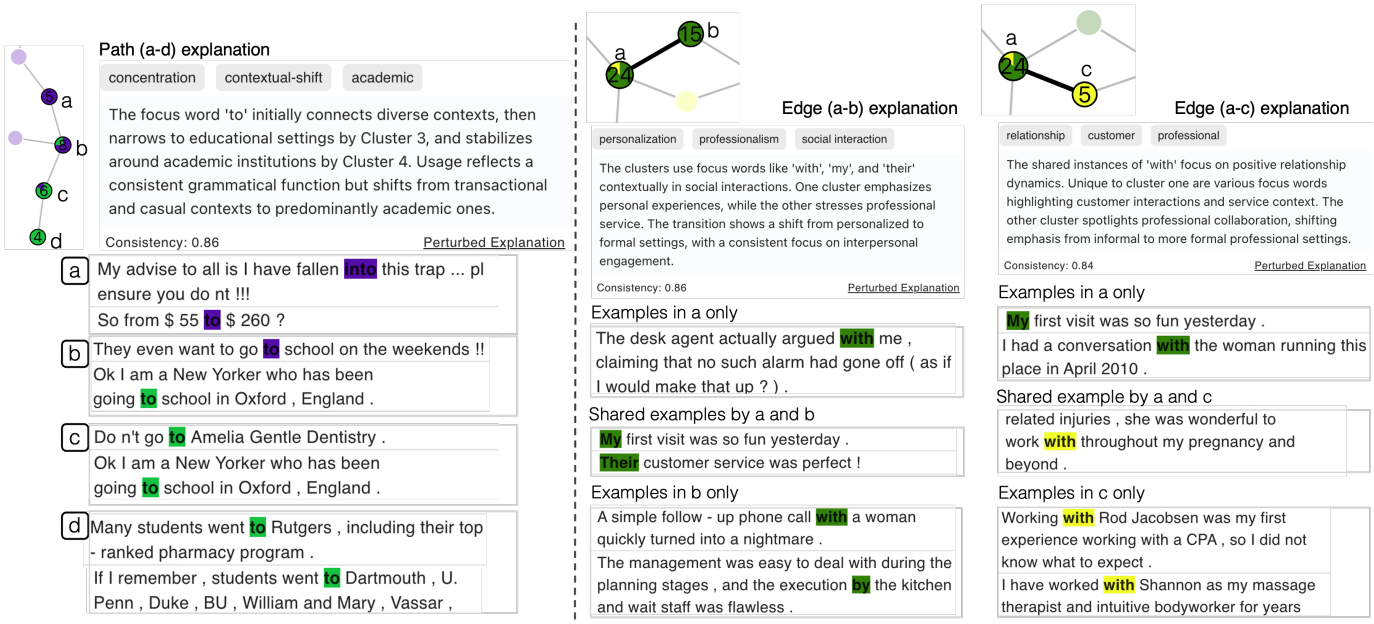

Fig. 18: Model behaviors in the final layer of the fine-tuned BERT. Left: a path illustrates supersense-role labels evolving from "Goal" (purple) to "Organization" (green). Right: two edges emerging from the same node differentiate between "Agent" (green) and "SocialRel" (yellow).

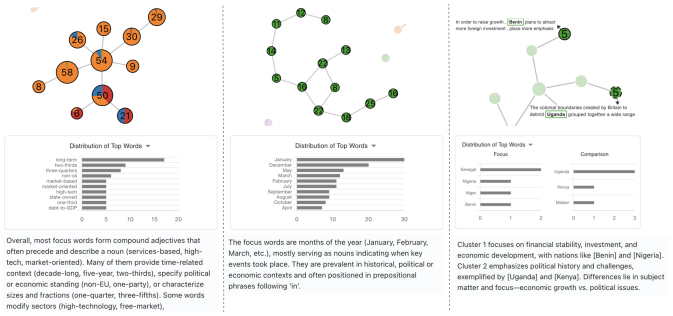

Fig. 19: Nuanced topological patterns in BERT-base. Left: compound adjectives in layer 3; middle: month names in layer 6; right: a subgraph of African countries with node-level comparison.

new sentences 1 and 2. Additionally, users may also delete sentences that contribute little to the trajectory, such as the original sentence 1, which only changes the time expression without affecting the mapper assignment. These editing operations allow users to iteratively refine trajectories and better inspect semantic transitions in the embedding space.

C Comparative Analysis across Models and Tasks

Building on the BERT-base supersense role classification setup (Section 7.2), we additionally fine-tune BERT-Tiny [1, 5], RoBERTa [2], and DeBERTa [4]. The dataset also includes supersense function labels capturing lexical semantic functions, and we fine-tune all four models on this classification task. For all models, we fix the mapper parameter to 50 intervals to maintain a consistent resolution across comparisons.

C.1 Model Behavior Comparison

Figure 21 shows the mapper graphs of the final layers of three BERT variants fine-tuned on the supersense role task. Together with the BERT-base result (Figure 4), we observe that most models separate labels into relatively pure components, whereas BERT-Tiny frequently mixes labels, suggesting that its reduced model capacity limits its ability to distinguish fine-grained semantic roles.

As shown in Figure 21, BERT-Tiny (left) produces a component in which *'from'* is mixed across the "Originator" and "Source" labels, and the component explanation reflects this blend (e.g., "a point of origin or source," "coming from a place"). RoBERTa (middle), by contrast, separates the two uses of *'from'* into distinct components, with explanations distinguishing "physical movement or origin" from "source of interaction." For DeBERTa (right), the *'from'* component labeled "Source" remains, while "Originator" uses of *'from'* merge with other tokens (e.g., *'my'*, *'their'*, *'your'*) into a larger component emphasizing "possession or origin." BERT-base exhibits a similar pattern to DeBERTa.

This example highlights that models vary in their ability to disentangle semantic roles. *Explainable Mapper* makes these differences visible through graph structures and explanations.

C.2 Data Representation Comparison

The fine-tuned models for supersense function form components with relatively coherent labels. Figure 22 illustrates the final-layer mapper graph of BERT-base fine-tuned on supersense functions.

Supersense functions express general prepositional meaning, whereas supersense roles offer finer context-dependent detail. As shown in Figure 16, when we search for the token *'my'* in final layer of BERT-base fine-tuned on supersense roles, *'my'* appears in a variety of components labeled as "SocialRel" (e.g., "*Both [my] grandparents looked ...*"), "Originator" (e.g., "*[My] only complaint is...*"), "OrgMember" (e.g., "*I LOVE [MY] GYM!*") , "Experiencer" (e.g., "*That's [my] favorite show ...*"), and others. In the supersense function labelling, they are broadly annotated as either "Possessor" or "Gestalt" (a generic relational meaning). Accordingly, in Figure 21, when we search for *'my'* in BERT-base fine-tuned on supersense functions, all occurrences fall into the two components labeled "Possessor" or "Gestalt", together with other related instances. The explanation indicates that the first component is mainly associated with personal goods, while the second component appears to capture border-related context. This shows that supersense role labels provide finer-grained distinctions than supersense function labels, offering computational linguists a new perspective on how different annotation schemes shape the learned representational structure.

D Multiscale Mapper Graphs

The visual complexity of mapper graphs can be controlled by tuning parameters such as the number of intervals: increasing this value reveals finer-grained structure, while decreasing it yields a coarser representation. Figure 23 presents mapper graphs of the final layer of BERT-base fine-tuned on the supersense role classification task under

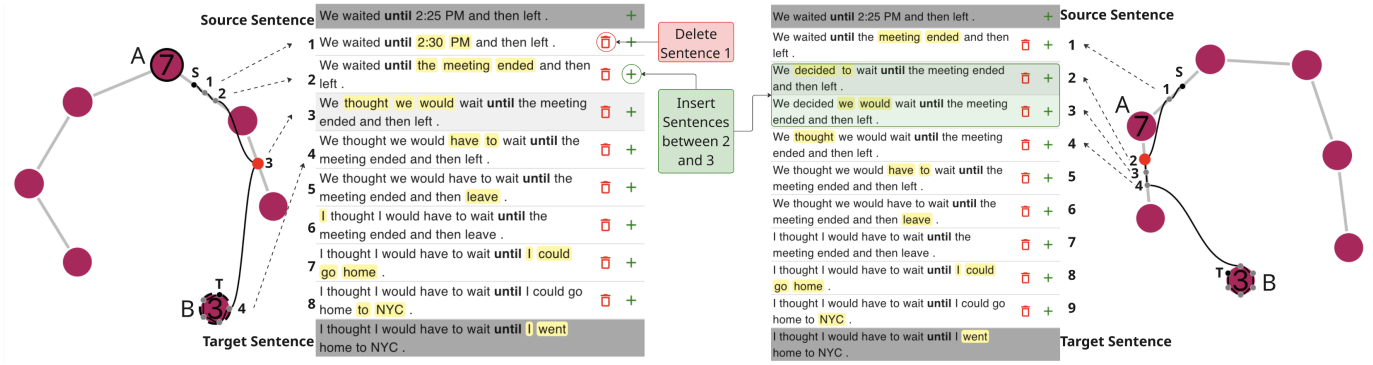

Fig. 20: Interactive refinement of a trajectory. Left: The initial trajectory between nodes A and B. Users may delete intermediate sentences (e.g., sentence 1) or insert new ones between consecutive steps with large semantic changes (e.g., between sentences 2 and 3) via LLM. Right: The trajectory after editing. LLM-identified key changes are highlighted in yellow, and clicking a sentence highlights its corresponding trajectory point.

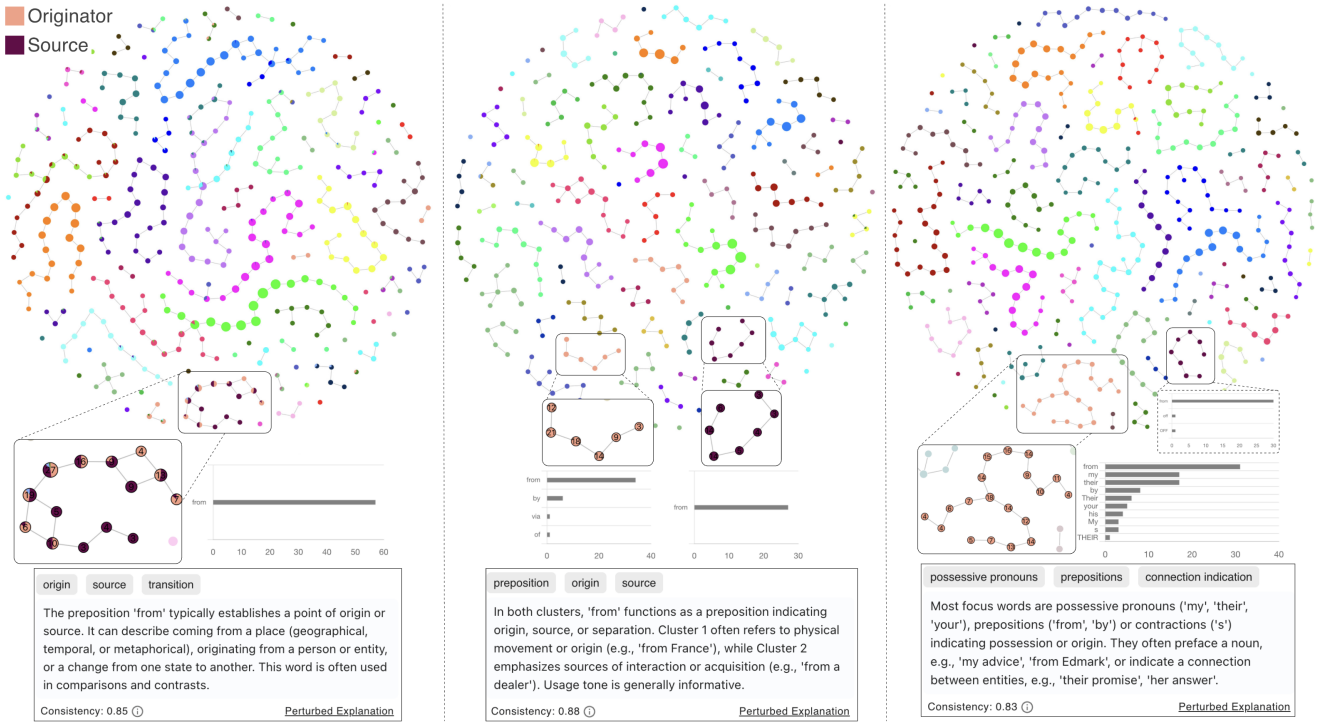

Fig. 21: Mapper graphs of the final layers of three BERT variants fine-tuned on the supersense role classification task: BERT-Tiny (left), RoBERTa (middle), and DeBERTa (right). Nodes are colored by supersense role labels. Components containing the word *'from'* with labels "Originator" and "Source" are highlighted, along with their token distributions and corresponding explanations.

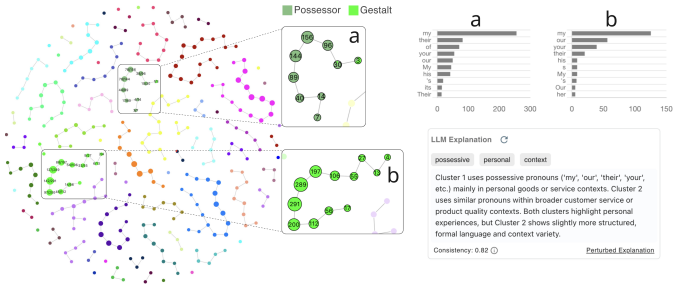

Fig. 22: Mapper graph of the final layer of BERT-base fine-tuned on the supersense function task. Nodes are colored by supersense function labels. Instances of the word '*my*' appear in two components labeled "Possessor" and "Gestalt." A comparison-based explanation of these two components is shown on the right.

three interval settings: 20 (A), 50 (B), and 80 (C). As the number of intervals increases, the graphs become more detailed: paths lengthen and components further separate. In (A), the labels *SocialRel*, *Originator*, and *Agent* appear within a single component; in (B), *Originator* separates from the other two; and in (C), all three labels are nearly fully separated.

We use mapper agents to interpret these components, as shown in Figure 23. The explanation for (A) highlights the distinctive presence of the word "with" and emphasizes the notion of "origin" relative to the other labels. The explanation for (B) uniquely emphasizes "social relations," while (C) primarily reflects "ownership," consistent with their respective labels.

Notably, these component explanations are generated independently. An important direction for future work is to enable LLMs to directly compare mapper elements across scales, thereby providing more explicit and structured characterizations of their differences.

E Expert-Agent Collaboration

From the expert study, we identified 26 episodes in which experts collaborated with the mapper agents to interpret a particular mapper element. We categorized each episode along several dimensions to better understand expert–agent collaboration behaviors.

As shown in Table 1, experts selected target mapper elements motivated by different aspects, including graph encoding (13; e.g., node glyphs, edge thickness, or graph structure), the projection view (6; e.g., lassoing a region or anchoring the layout to locate subgraphs), querying a specific token of interest (5), and using annotations (2). Once selected, experts either summarized or compared the chosen elements. Before opening the LLM explanation, they formed three types of initial hypotheses about the element: concrete hypotheses (7), where they closely inspected the metadata and propose a specific interpretation; general hypotheses (7), where they briefly glanced at the metadata to form a high-level expectation; and weak or no clear hypotheses (13), where they struggled to interpret the element or chose not to make an interpretation. Then, experts interacted with the LLM explanations, such as verifying them against the metadata, referencing their initial hypotheses when relevant, and regenerating explanations when needed. Ultimately, they either confirmed (19), rejected (3), remained indecisive (2), or did not explicitly report (2) a judgment of the LLM interpretation.

E.1 Collaboration Examples

Figure 24 shows a case where the mapper agent helped expert E2 recalibrate his interpretation of mapper components, corresponding to case 9 in Table 1. E2 queried the token "resource" in layer 10 and identified two relevant components. After carefully inspecting their metadata, E2 proposed *a concrete hypothesis* of the difference between the components lies in their contexts: the first component was about island countries, whereas the second was about land countries.

E2 then used the mapper agent to generate a comparison explanation, which described the first component as focusing on "abundance or lack" and the second on "scarcity." E2 disagreed with this characterization, commenting that "lack and scarcity are very close; I wouldn't know how to interpret it." E2 also noted that the high consistency score (0.92) did not capture the quality of the explanation and could be misleading. After revisiting the metadata to understand why their original hypothesis was not reflected, E2 found several examples in the second component that were also about island countries. E2 concluded that "I didn't notice them before; the cluster is not super clean, this is a new finding," leading him to reconsider the completeness of his original hypothesis.

Figure 25 shows a case where the mapper agent helped expert E5 confirm and enrich her general interpretation of a mapper node, corresponding to case 20 in Table 1. E5 selected a large node and tried to understand its meaning. After quickly scrolling through the sentence list, she proposed a broad summary that the cluster was about "economic" content and "growth numbers." She then opened the LLM explanation, which characterized the cluster as capturing "financial conditions, growth patterns, and sector contributions." After cross-checking this explanation against the metadata, she judged it to be "a good summary, I can manually see this in the instances." She subsequently regenerated the explanation to seek additional insights; the new explanation further emphasized growth and sectors contributing to GDP. E5 confirmed this explanation and found it more detailed than her initial hypothesis, demonstrating how the mapper agent can efficiently validate and refine expert interpretations.

However, we also observed cases where experts struggled to verify the explanations. In case 23 (Table 1), E5 used the mapper agent to explain a long path but found it difficult to assess its correctness, since she had to visit each node to look for supporting evidence, which was time-consuming. She therefore remained indecisive about this path explanation and suggested an interaction where hovering over phrases in the explanation would directly jump to the relevant sentences to increase the users' trust.

E.2 Expert Questionnaire Rating

The expert questionnaire rating is shown in the Figure 26.

F Prompt Templates and Engineering

Prompt templates. For the Node and Component Explainers, Figure 28 and Figure 29 present the prompt templates used to summarize a node/component and to compare two nodes/components. For the Edge Explainer, Figure 30 shows the prompt template to summarize an edge. For the Path Explainer, Figure 31 show the prompt templates for summarizing a path. For the Trajectory Explainer, Figure 32 shows

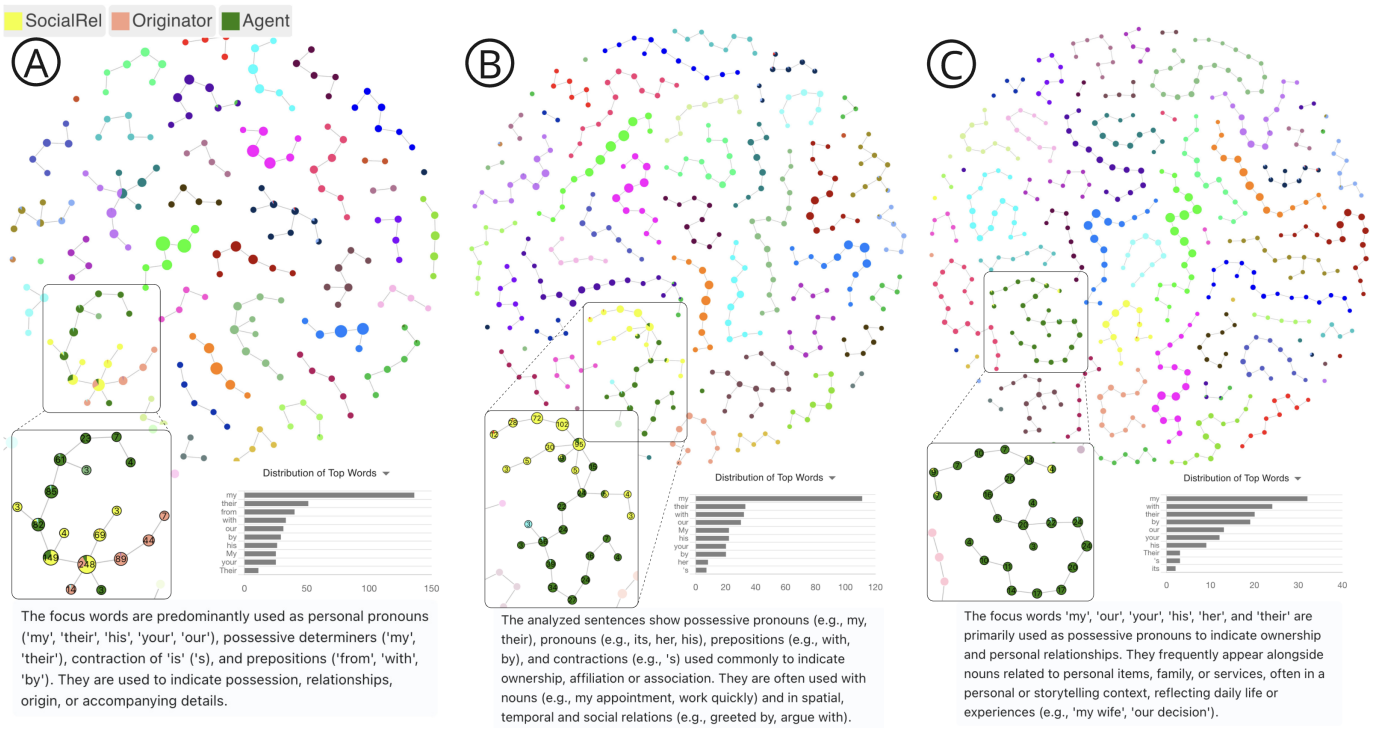

Fig. 23: Mapper graphs of the final layer of BERT-base fine-tuned on the supersense role classification task, using interval settings of 20 (A), 50 (B), and 80 (C). Increasing the number of intervals progressively separates the labels *SocialRel*, *Originator*, and *Agent*. For each highlighted component, we present its token composition and corresponding explanation.

the prompt template used to generate a sequence of minimal (ideally a single token) perturbations between the source and target sentences. Figure 33 shows the prompt template for identifying edits between consecutive intermediate sentences in a perturbation trajectory. In addition, Figure 34 presents the prompt template used to generate perturbed sentences of a given sentence through 1-token replacement and rephrasing. This template is used by the Node, Edge, Path, and Component Verifier.

Prompt engineering. To fully unleash the capabilities of LLMs, we employ established prompt engineering techniques in our initial prompts, such as role-playing and explicit input-output definitions. We then iteratively refine these prompts through qualitative experimentation, initially focusing on the mapper node summary. Once validated, the refined strategies are extended to prompts used in other explanation tasks. As shown in Figure 35, the node summary prompt undergoes three main iterations, each introducing an additional guiding statement (1–3). For example, Statement 1 explicitly highlights linguistic properties, encouraging the LLM to produce more linguistically grounded responses. Statement 2 prompts LLMs to consider a cluster of words as a whole rather than treating them based on lexicon; Statement 3 prompts LLMs to provide concrete examples to support its output, making the explanation more comprehensible and relatable.

G Data Processing

Based on feedback from the pilot study, we focus on words with clear semantic meaning from the Groningen Meaning Bank (GMB) dataset. Specifically, we utilize the *CIA World Factbook* subset, and include words that are nouns, verbs, adjectives, and adverbs, with corresponding part-of-speech (POS) tags: `NN, NNS, NNP, NNPS, VB, VBD, VBG,`

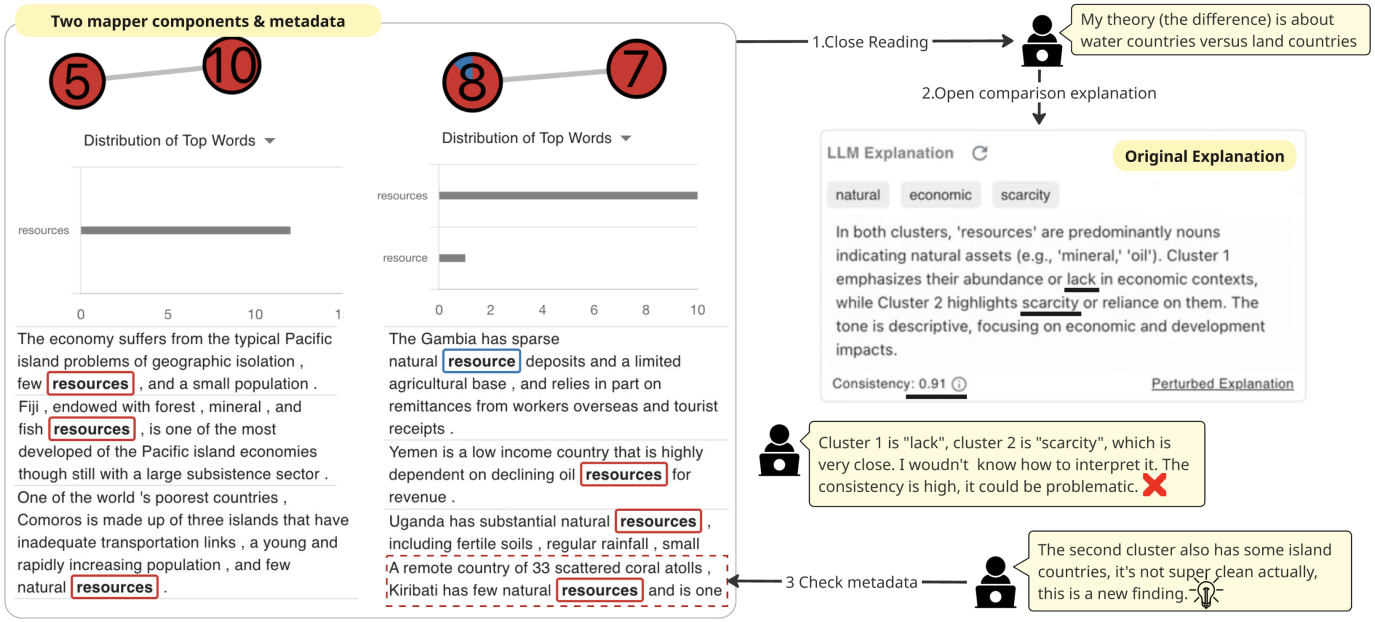

Fig. 24: A case in which the agent helps calibrate E2's hypothesis about component differences. Left: two mapper components related to the word "resources," along with samples of their metadata. Right: E2's interactions with the agent during the interpretation of these components.

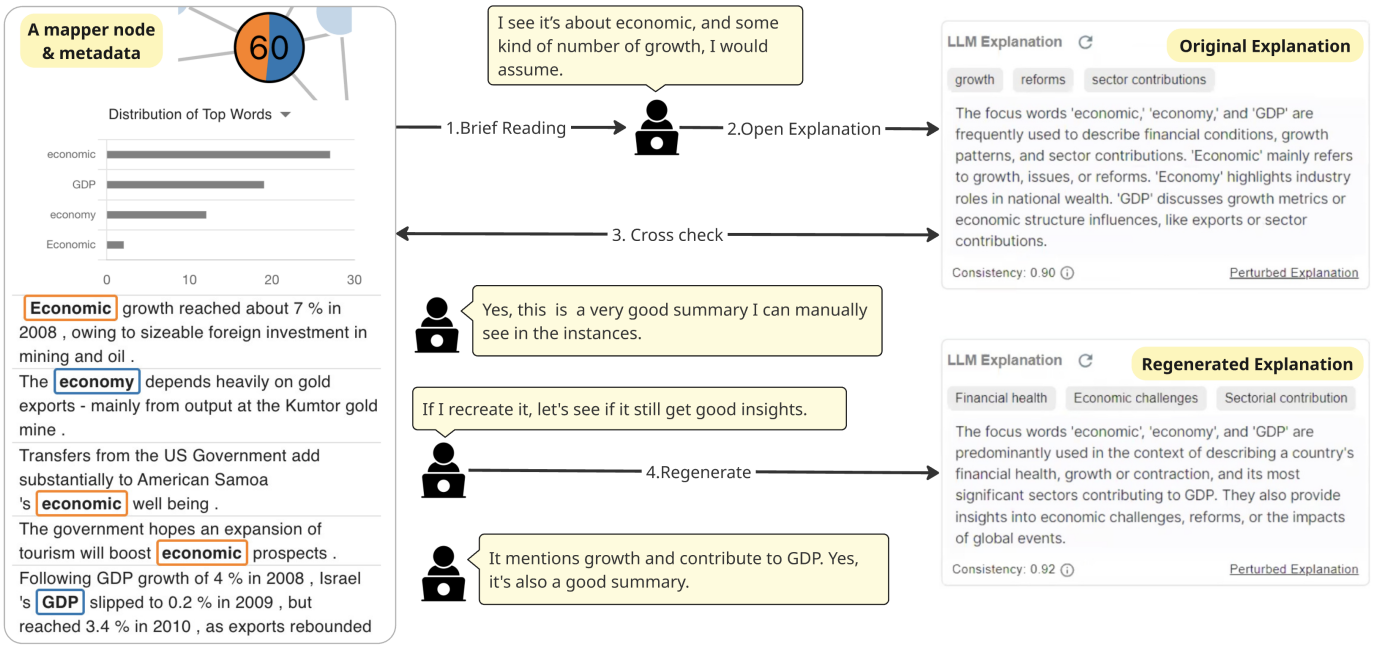

Fig. 25: A case in which the agent helps E5 confirm and enrich her general hypothesis about a node summary. Left: the selected Mapper node and samples of its metadata. Right: E5's interactions with the agent during the interpretation of the node.

Table 1: Expert-agent interaction episodes. *Selection*: how a mapper element was chosen; *Agent Task*: analytic performed on the element; *Initial Hypothesis*: the expert's expectation about the element before reading the LLM explanation, and we coded them based on the specificity (1 = concrete, 2 = general, 3 = none). *Outcome*: expert's final judgment of the LLM explanations.

ID	Expert	Selection	Agent Task	Initial Hypothesis	Outcome
1	E1	Graph encoding	Node summary	1	Confirm
2		Graph encoding	Edge summary	1	Reject
3		Graph encoding	Path summary	2	Unreported
4		Graph encoding	Node comparison	3	Confirm
5		Projection selection	Component comparison	1	Confirm
6		Projection selection	Component comparison	2	Confirm
7	E2	Graph encoding	Node summary	3	Confirm
8		Graph encoding	Edge summary	1	Confirm
9		Token query	Component comparison	1	Reject
10		Annotation	Path summary	2	Confirm
11	E3	Token query	Path summary	3	Confirm
12		Token query	Node comparison	3	Confirm
13		Token query	Edge summary	3	Unreported
14	E4	Projection selection	Node summary	2	Confirm
15		Graph encoding	Component summary	3	Confirm
16		Projection selection	Component summary	2	Confirm
17		Projection selection	Component comparison	3	Confirm
18		Projection selection	Component comparison	3	Confirm
19	E5	Graph encoding	Node summary	3	Reject
20		Graph encoding	Node summary	2	Confirm
21		Graph encoding	Node comparison	1	Confirm
22		Graph encoding	Edge summary	1	Indecisive
23		Graph encoding	Path summary	3	Indecisive
24		Annotation	Component summary	2	Confirm
25	E6	Graph encoding	Component comparison	3	Confirm
26		Token query	Path summary	3	Confirm

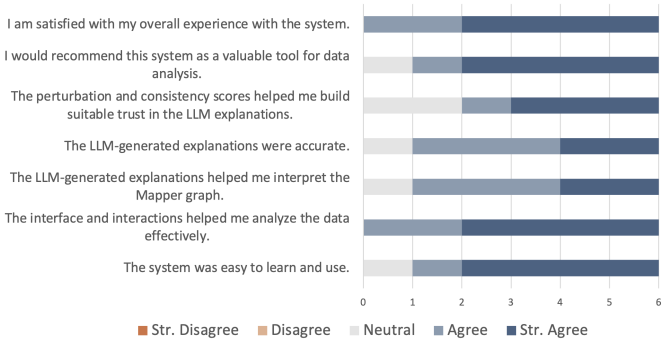

Fig. 26: Expert perceptions of the system measured with Likert-type questions.

`VBN, VBP, VBZ, JJ, JJR, JJS, RB, RBR, RBS`. We further remove words whose lemmas are *'be'*, *'have'*, or *'%'*.

After preprocessing, we obtain 27,504 words across 2,252 sentences. Each word embedding is constructed by averaging the token embeddings that compose the word. The processed dataset is used in case studies comparing BERT-base with ModernBERT, as well as in the expert study.

H Implementation, Storage, and Computation Time

Implementation details. For the workspace implementation, we use JavaScript, D3, Material UI, and the React framework for the frontend, while the backend is built with Python and Flask. The classical mapper is computed using the Python library *KeplerMapper* [6, 7], and the ball mapper is computed with *pyBallMapper* [3]. For dimensionality reduction, we use *scikit-learn* for PCA and t-SNE, and *umap-learn* for UMAP.

The explanation and verification agents are implemented via prompting LLMs, with complete prompt templates provided in Apendix F. We use the *GPT-4o* API for all explainers and verifiers.

Mapper computation time. We report the computation time for the classical mapper. In the first case study, using embeddings from the TREUSLE v4.2 dataset with a fine-tuned BERT model, the dataset contains 4,282 tokens with an embedding dimension of 768. The classical mapper runs in 0.193 seconds on average (SD = 0.020). In the second case study, using embeddings from the GMB dataset with both BERT-base and ModernBERT, the dataset contains 27,504 tokens with a dimension of 768. The mean runtime of the classical mapper is 2.079 seconds (SD = 0.663) for BERT-base and 2.186 seconds (SD = 0.582) for ModernBERT. All computations were performed on a MacBook Pro equipped with an Apple M4 Pro chip and 48 GB of RAM.

Storage overhead of precomputed explanations. To support rapid exploration, *Explainable Mapper* adopts a hybrid execution strategy. For all nodes and connected components in the default mapper graph, it precomputes and caches the original explanation, perturbed explanation, and verification consistency score because these are accessed most frequently. Explanations for other operations are generated on demand after users select the relevant mapper elements, including explanation comparison, edge and path explanations, trajectory generation, and explanation refresh.

For each element, the system stores an explanation, its perturbed explanation, and a verification consistency score. Table 2 summarizes the storage requirements of the cached data for the two datasets used in the main paper across all 12 layers of the BERT embedding space. Even for the larger GMB dataset, the cached annotations require less than 1.5 MB per layer, indicating a modest memory overhead.

Online explanation computation time. To characterize online latency, we measured the time required for node explanation and verification

Table 2: Summary of precomputed explanations across the 12 layers of the BERT embedding space. For each dataset, we report the number of connected components and mapper nodes, along with the per-layer storage required for cached explanations (mean ± standard deviation).

Dataset	Components / Layer	Nodes / Layer	Storage / Layer
STREUSLE v4.2 (Fine-tuned BERT-base)	56 ± 24	360 ± 102	0.42 ± 0.13 MB
GMB CIA World Factbook (BERT-base)	487 ± 164	823 ± 138	1.42 ± 0.26 MB

Table 3: Online latency for node explanation and verification across different node sizes. Values are reported as mean ± standard deviation over ten sampled nodes per size range. Latency remains largely stable across node sizes.

Node Size	1–10	10–20	20–30	30–40	40–50	50–60	60–70	70–80	80–90	90–100
Latency (s)	3.28 ± 0.83	3.07 ± 0.66	3.32 ± 0.69	3.09 ± 0.50	3.46 ± 0.82	3.15 ± 0.99	3.24 ± 0.68	2.98 ± 0.27	3.43 ± 0.54	3.28 ± 0.65

across nodes of varying sizes. Because all explanation tasks use the same LLM-based prompting strategy, node explanation is representative of the online response time. We grouped nodes by the number of contained instances and randomly sampled ten nodes from each group. As shown in Table 3, latency remained stable at approximately 3–3.5 seconds across all examined node sizes (0–100 instances), suggesting that node explanation and verification can be generated within a few seconds regardless of node size.

We evaluated the computation time of trajectory generation using 20 randomly selected mapper-node pairs. For each pair, we generated one trajectory and recorded the computation time and the number of intermediate sentences. On average, a trajectory contained 9.3 ± 3.1 intermediate sentences and required 7.9 ± 2.7 seconds to generate.

I Model and Fine-tuning Details

In this paper, we fine-tune four BERT variants, including BERT-base, RoBERTa, DeBERTa, and BERT-tiny, on the TREUSLE v4.2 dataset for two linguistic classification tasks, based on two sets of word labels: supersense role (53 labels) and supersense function (38 labels).

All models are fine-tuned using the HuggingFace library with the AdamW optimizer and a batch size of 32. We use a linear learning-rate scheduler with 10% warm-up and a fixed learning rate of 3×10^{-4} for all models. For BERT-base, RoBERTa, and DeBERTa, we train for 7 epochs, and for BERT-tiny we train for 10 epochs. We then generate contextualized embeddings of the 4,282 words in the training set. We report the layers, embedding dimensions, model sizes, and HuggingFace Model ID for each pretrained model in Table 4. We show mapper graphs of the final layer of three BERT variants fine-tuned on the supersense function classification task in Figure 27.

J Alternative Robustness Metrics

We implemented BERTScore [8] as an alternative explanation consistency metric to cosine similarity. Specifically, the system reports the BERTScore F1 score computed using the RoBERTa-large model (see Figure 36).

References

[1] P. Bhargava, A. Drozd, and A. Rogers. Generalization in NLI: Ways (not) to go beyond simple heuristics. In J. Sedoc, A. Rogers, A. Rumshisky, and S. Tafreshi, eds., *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pp. 125–135. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021. doi: 10.18653/v1/2021.insights-1.18 5

[2] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán et al. Unsupervised cross-lingual representation learning at scale. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.747 5

[3] P. Dlotko. Ball mapper: a shape summary for topological data analysis. arXiv preprint arXiv:1901.07410, 2019. doi: 10.48550/arXiv.1901.07410 9

[4] P. He, J. Gao, and W. Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. arXiv preprint arXiv: 2111.09543, 2021. doi: 10.48550/arXiv.2111.09543 5

[5] I. Turc, M.-W. Chang, K. Lee, and K. Toutanova. Well-read students learn better: On the importance of pre-training compact models. arXiv preprint arXiv:1908.08962, 2019. doi: 10.48550/arXiv.1908.08962 5

[6] H. J. van Veen, N. Saul, D. Eargle, and S. W. Mangham. Kepler Mapper: A flexible Python implementation of the Mapper algorithm. *Journal of Open Source Software*, 4(42):1315, 2019. doi: 10.21105/joss.01315 9

[7] H. J. van Veen, N. Saul, D. Eargle, and S. W. Mangham. Kepler Mapper: A flexible Python implementation of the Mapper algorithm, Oct. 2020. doi: 10.5281/zenodo.4077395 9

[8] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. BERTScore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019. doi: 10.48550/arXiv.1904.09675 10

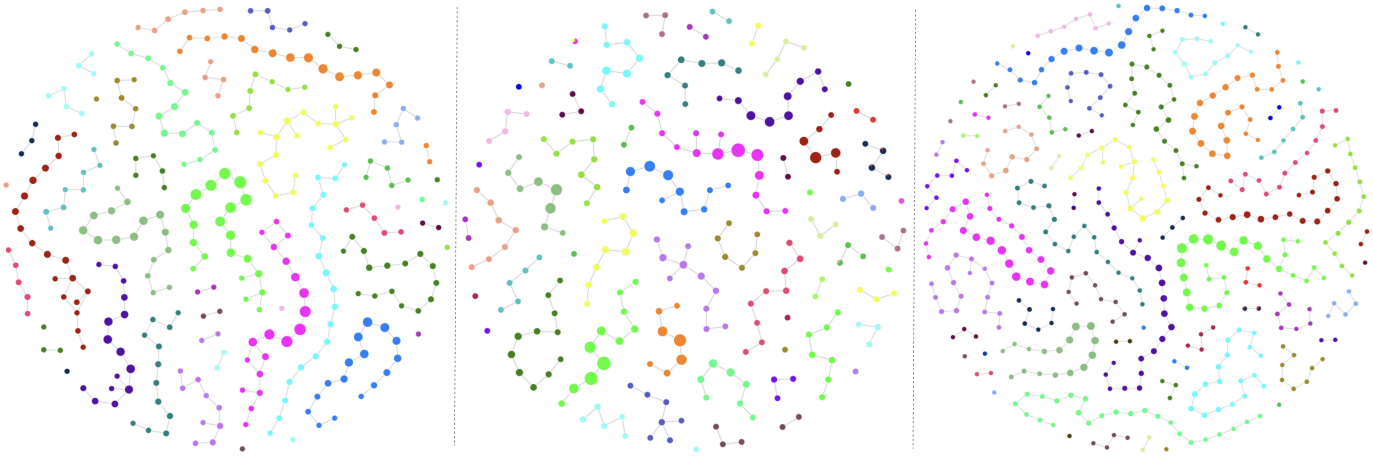

Fig. 27: Mapper graphs of the final layer of three BERT variants fine-tuned on the supersense function classification task: BERT-Tiny (left), RoBERTa (middle), and DeBERTa (right). Nodes are colored by supersense function labels.

Table 4: Summary of models used for fine-tuning tasks.

Model	Layers	Hidden Dim	Params	HuggingFace Model ID
BERT-base	12	768	110M	`google-bert/bert-base-uncased`
BERT-tiny	2	128	4.4M	`prajjwal1/bert-tiny`
RoBERTa-base	12	768	125M	`FacebookAI/roberta-base`
DeBERTa-v3-base	12	768	86M	`microsoft/deberta-v3-base`

INSTRUCTION:
You are a linguist analyzing word usage. Given a set of sentences, each containing a focus word, your task is to analyze these sentences to determine how these focus words are commonly used. Consider the word's part of speech, surrounding words, tone, subject, context, and meaning. Summarize the highly common patterns in 50 words or fewer, then list three key descriptors.

For each sentence, you will receive:
- The focus word.
- The sentence, with the focus word enclosed in [].

Please note that these focus words may differ. Rather than explaining them individually, focus on their common usage.
Where relevant, include concrete examples in your summary to illustrate these patterns.
Provide your response in the following JSON format:
{"summary": "textual summary", "keywords": ["descriptor1", "descriptor2", "descriptor3"]}

INPUT:
[*Instances in this mapper node / Component*]

Fig. 28: Prompt template for summarizing a mapper node/component.

INSTRUCTION:
You are a linguist analyzing word usage. You will be given two clusters of sentences. Each sentence contains a focus word, which is enclosed in brackets []. Your task is to compare how the focus words are used **across the two clusters**.

For each cluster, consider the focus words' part of speech, surrounding words, tone, subject, context, meaning, and position in the sentence. Then, compare the clusters to identify common patterns and notable differences in usage.

Summarize the comparison in 50 words or fewer. Where helpful, include concrete examples to illustrate the contrast or similarity. Provide three keywords that best capture the essence of your comparison.

Format your response in the following JSON format:
{"summary": "brief textual comparison", "keywords": ["descriptor1", "descriptor2", "descriptor3"]}

INPUT:
Cluster 1: [*Instances in the first mapper node / Component*]
Cluster 2: [*Instances in the second mapper node / Component*]

Fig. 29: Prompt template for comparing two mapper nodes/components.

INSTRUCTION:
You are a linguist analyzing word usage. There are two clusters of sentences, where each sentence contains a focus word. The two clusters have overlapping instances as well as instances unique to each cluster.
Your task is to analyze the transition between the two clusters regarding the usage of focus words, Consider the word's part of speech, surrounding words, position, tone, subject, context, and meaning, by:
1. Identifying how the overlapping instances are used in both clusters.
2. Highlighting how the unique instances in each cluster differ.
3. Summarizing how one cluster conceptually or linguistically shifts into the other.

You will receive:
- A list of overlapping instances.
- A list of instances unique to one cluster.
- A list of instances unique to another cluster.

Please note that these focus words may differ. Rather than explaining them individually, focus on their common usage.
Where relevant, include concrete examples in your summary to illustrate these patterns. Respond in the following JSON format:
{ "summary": "Summary of how the two clusters are related and how they diverge, in 50 words or fewer.", "keywords": ["pattern1", "pattern2", "pattern3"]}.

INPUT:
Overlapping Instances: [*Instances*]
Unique Instances in one cluster: [*Instances*]
Unique Instances in another cluster: [*Instances*]

Fig. 30: Prompt template for summarizing a mapper edge.

INSTRUCTION:
You are a linguist analyzing how word usage evolves across clusters of sentences over time. Each cluster contains a set of sentences, each with a focus word (enclosed in []). Your task is to analyze each cluster in the context of its position in the sequence and identify how the usage patterns of focus words shift, develop, or stabilize across the clusters.

For each cluster, consider the focus words' parts of speech, surrounding words, tone, subject, context, and meaning.

Summarize the **evolution** across clusters in 50 words or fewer, highlighting how usage changes over time. Then list three key descriptors that capture this progression. Concrete examples from different clusters should be included if they help clarify the evolution.

Provide your response in the following JSON format:
{"summary": "textual summary of evolution", "keywords": ["descriptor1", "descriptor2", "descriptor3"]}

INPUT:
Cluster 1: [*Instances in the 1st mapper node*]
Cluster 2: [*Instances in the 2nd mapper node*]
...
Cluster n: [*Instances in the nth mapper node*]

Fig. 31: Prompt template for summarizing a mapper path.

INSTRUCTION:
I have two sentences, each containing a focus word marked with brackets []:
Start: {*Source sentence*}; End: {*Target sentence*}

Your goal is to gradually transform the Start sentence into the End sentence using a sequence of intermediate sentences, where each step differs by only a single-word edit as much as possible.

Guidelines:
- Each consecutive pair should differ by **a single-word edit** (insertion, deletion, or replacement) as much as possible.
- Every intermediate sentence must be grammatically correct and semantically coherent.
- Each sentence must contain exactly one focus word, marked with [].
- Use the same focus word if it appears in both the first and last sentences.

You output the list of sentences in order, including the start and end sentences, provide a brief summary (≤ 50 words) explaining how the usage of the focus word evolved across the perturbation path.
Respond in the following JSON format:
{"sentences": [
"Start Sentence",
"Perturbation sentence 1",
...,
"Perturbation sentence N",
"End Sentence"
], "summary": "Summary of how the usages of focus words evolve along the perturbation path, in 50 words or fewer. }

Fig. 32: Prompt template for creating a perturbation trajectory between two sentences that constrains all intermediate sentences to be grammatically correct and semantically coherent.

INSTRUCTION:
You are given a sequence of sentences. Each sentence is derived from the previous one.
Sentences: {*a list of sentences*}

Compare each sentence with its immediately preceding sentence. For every sentence starting from index 1, return the current sentence with the changed words or spans wrapped in <edit>...</edit> tags.
Mark the changed region as minimally as possible. For added text, wrap only the newly added words, not surrounding unchanged words or phrases.
As much as possible, do not include words enclosed in square brackets [] inside <edit> tags, because those brackets mark the focus word. Only mark the bracketed focus word if the focus word itself changed.
If text was removed and there is no corresponding word in the current sentence, return the current sentence without any <edit> tags.

Return only valid JSON in this exact format:
{"results": [{
"sentence_index": 1,
"marked_sentence": "The <edit>large cat</edit> sat on the mat."
}]}
Keep the square brackets around the focus word unchanged.
Include every sentence index from 1 through {len(sentences) - 1}.

Fig. 33: Prompt template for identifying edits between consecutive sentences in a perturbation trajectory.

INSTRUCTION:
Given a sentence with a focus word enclosed in square brackets [], generate five perturbation examples of the sentence while keeping the focus word unchanged.

Each perturbation should either:
- Rephrase the sentence without altering its original meaning.
- Apply a one-token perturbation (change or substitute a single word other than the focus word).

Do not introduce negation. Respond in the following JSON format: a list of sentences, with the focus word still enclosed in [].

Example input:
Sentence: The [weather] today is perfect for a picnic.

Expected output:
["Today's [weather] is ideal for a picnic.",
"Today's [weather] is perfect for a picnic.",
"The [weather] is wonderful for a picnic today.",
"The [weather] today is excellent for a picnic.",
"The [weather] today is great for a picnic."]

INPUT: [*A sentence*]

Fig. 34: Prompt template for generating the perturbed sentences of a given sentence.

INSTRUCTION:
You are a linguist analyzing word usage. Given a set of sentences, each containing a focus word, your task is to analyze these sentences to determine how these focus words are commonly used. **[1]** Consider the word's part of speech, surrounding words, tone, subject, context, and meaning. Summarize the highly common patterns in 50 words or fewer, then list three key descriptors.

For each sentence, you will receive:
- The focus word.
- The sentence, with the focus word enclosed in [].

Please note that these focus words may differ. **[2]** Rather than explaining them individually, focus on their common usage. **[3]** Where relevant, include concrete examples in your summary to illustrate these patterns.
Provide your response in the following JSON format:
{"summary": "textual summary", "keywords": ["descriptor1", "descriptor2", "descriptor3"]}

INPUT:
[*Instances in this mapper node / Component*]

Examples in a mapper node

word	label	sentence
of	Species	They have a great selection **of** wine from all over the world with all different prices .
of	Species	Even though we were only supposed to have those specific types **of** pizza , when guests asked for a different type , it was brought out with no charge to us !
of	Species	They have all kind **of** coupons available for car washes , oil changes , state inspection , etc .
of	Species	The selection **of** meats , veggies and sauces is awesome too !
of	Species	I admit that I should have paid attention to this kind **of** little things while test drive .

Summary (prompt without statements 1, 2, 3)
The word 'of' is commonly used to denote relationships between parts and wholes, categories, and types of things, often indicating variety or assortment in a context.

Summary (prompt with statement 1)
The word 'of' often follows nouns indicating categories or quantities, such as 'type', 'kind', 'selection', and 'variety'. It highlights specificity within groups in contexts related to products or options. Usage is typically neutral, factual, and descriptive.

Examples in a mapper node

word	label	sentence
on	Locus	Extensive drink list and daily specials but wish they had a bit more **on** their food menu , although popcorn is a nice touch !
in	Locus	A few minutes after I left , I was called and informed that " I " left my wheel lock (which they should have left **in** the car) .
around	Locus	And the salsa , be sure to ask for a jar and have plenty of chips **around** , you will need them .....
in	Locus	There were 3 adults **in** our room but towels for only 2 , no linens for sofa bed .
at	Locus	I typically have work done on my Jeep **at** the dealership , but it is 6 years old now and getting charged dealership prices just did n't seem cost effective anymore .

Summary (prompt with statement 1)
The focus words often indicate location, time, or means. 'In' is used spatially or metaphorically to denote inclusion within. 'On' suggests placement or inclusion, often referring to a surface or list. 'At' typically indicates a specific location or situation. Each word structurally supports a relationship or context within sentences.

Summary (prompt with statements 1, 2)
Focus words like 'in', 'on', and 'at' primarily serve as prepositions, indicating location or context within sentences. They frequently connect subjects with locations or conditions, common in hospitality, personal stories, or service-based contexts. Typically neutral or direct, they set situational tones in everyday narratives.

Summary (prompt with statements 1, 2, 3)
Prepositions like 'in,' 'on,' 'between,' and 'at' commonly denote location, time, or other spatial-temporal relationships. They often follow nouns or verbs, providing details about position or context (e.g., 'on the menu,' 'in one place'). These words convey a straightforward tone and precise relationships.

Fig. 35: An example of prompt engineering for the mapper node summary. Left: the final prompt, with Statements 1–3 added incrementally. Top right: node summaries generated with and without Statement 1. Bottom right: node summaries generated as Statements 2 and 3 are gradually added.

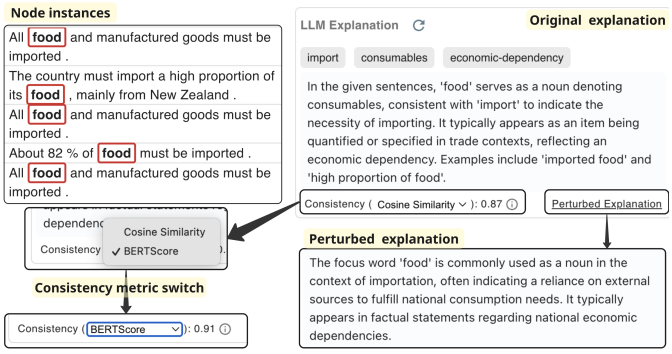

Fig. 36: BERTScore as an alternative explanation consistency metric. The interface displays the instances within a mapper node together with their original and perturbed explanations, and allows users to switch between consistency metrics for comparison.