

ORACLE: Time-Dependent Recursive Summary Graphs for Foresight on News Data Using LLMs

Lev Kharlashkin, Eiaki Morooka, Yehor Tereshchenko and Mika Hämäläinen

Metropolia University of Applied Sciences

Helsinki, Finland

first.last@metropolia.fi

Abstract

ORACLE turns daily news into week-over-week, decision-ready insights for one of the Finnish University of Applied Sciences. The platform crawls and versions news, applies University-specific relevance filtering, embeds content, classifies items into PESTEL dimensions and builds a concise *Time-Dependent Recursive Summary Graph* (TRSG): two clustering layers summarized by an LLM and recomputed weekly. A lightweight change detector highlights what is *new*, *removed* or *changed*, then groups differences into themes for PESTEL-aware analysis. We detail the pipeline, discuss concrete design choices that make the system stable in production and present a curriculum-intelligence use case with an evaluation plan.

1 Introduction

Foresight is the systematic detection, interpretation, and anticipation of signals of change to inform long-term decision-making (Chandrasekaran et al., 2022; Yüksel, 2012). For our setting, this means turning large, unstructured text streams (news, reports, social media) into structured representations of emerging trends and weak signals that remain interpretable to human analysts. Public institutions and universities in particular need early indicators of technological, policy, or societal change to align curricula, partnerships, and research priorities (Hämäläinen et al., 2024). Traditional approaches (expert workshops and static reports) cannot match the velocity and volume of modern information flows, motivating continuously operating, evidence-traceable foresight systems.

ORACLE addresses this need for a University of Applied Sciences by transforming Finnish news into a weekly *recursive summary graph* that reveals how narratives emerge, merge, and fade over time, grounded in PESTEL dimensions (see Yusop 2018). Unlike generic dashboards, ORACLE is designed

for ongoing operation, traceability, and institutional relevance.

Our contributions are:

- A practical pipeline for daily Finnish news ingestion with hashing-based versioning and vectorized storage.
- A two-level Time-Dependent Recursive Summary Graph (TRSG) that hierarchically organizes narratives and updates weekly while accumulating long-term knowledge.
- A week-to-week change-detection mechanism that groups themes and analyzes institutional relevance through PESTEL.
- A real-world use case—curriculum intelligence—demonstrating decision support for academic stakeholders.

2 Related Work

Semantic representations and retrieval. Dense sentence embeddings (e.g., SBERT, SimCSE) enable semantic clustering and temporal tracking over large text streams (Reimers and Gurevych, 2019; Gao et al., 2021). For scalable retrieval, approximate nearest-neighbor search and vector databases such as FAISS and Milvus support efficient similarity queries over continuously ingested corpora (Johnson et al., 2021; Wang et al., 2021).

Clustering and community detection. Graph-based community detection (Louvain, Leiden) is widely used to expose latent structure in text-derived graphs (Blondel et al., 2008; Traag et al., 2019). Hybrid topic models like BERTopic combine transformer embeddings with clustering to yield interpretable topics, though they are typically applied to static snapshots rather than rolling streams (Grootendorst, 2022).

Summarization and abstraction. Neural abstractive models (BART, PEGASUS) produce flu-

ent summaries but face challenges in factual grounding at multi-document scale (Lewis et al., 2020; Zhang et al., 2020). Extractive methods (LexRank, TextRank) remain strong for faithfulness and traceability (Erkan and Radev, 2004; Mihalcea and Tarau, 2004). Streaming and dynamic summarization explores hierarchical, time-aware abstractions over evolving corpora (Huang et al., 2024).

Temporal modeling and change detection. Modeling evolving topics spans burst detection and dynamic topic models, capturing intensity and drift over time (Kleinberg, 2003; Blei and Lafferty, 2006). These lines inform explicit week-over-week change labeling and rolling updates for interpretable monitoring.

LLMs in foresight and strategic analysis. Recent work investigates LLMs for structured foresight, including multi-layer PESTEL-based prompting to scaffold analysis (Alnajjar and Hämäläinen, 2024). Our system operationalizes these ideas as a continuous, interpretable pipeline combining semantic retrieval, dynamic clustering, hierarchical summarization, and explicit change tracking tailored to institutional decision-making.

3 Oracle Platform

3.1 Data Ingestion and Versioning

Sources. Finnish news (e.g., Yle) using open-source RSS feeds are crawled daily. The pipeline extracts canonical URLs and main content (boilerplate-stripped), preserving the original HTML for audit.

Hashing. We compute a stable content hash over normalized HTML. If the hash changes for a known URL, a new version is stored and re-embedded. This suppresses duplicates while tracking edits (e.g., headline updates).

3.2 Relevance Filtering for University

A two-stage filter prioritizes items that matter institutionally:

1. **Lexical stage.** Query expansions over names (*University*, *UAS*, Finnish aliases), domains (education, R&D, local industry) and geography. This fast pass removes obviously unrelated stories.
2. **Semantic stage.** Embedding-based similarity against curated exemplars (e.g., skills funding,

curriculum reform, regional innovation). Borderline items are kept if semantically close.

Non-relevant items are cold-stored for later replays (interests can shift).

3.3 Embeddings, Storage and PESTEL

Documents are embedded (OpenAI TextEmbedding-3) and stored in Milvus with metadata: source, , publication date, PESTEL label and version chain. A compact supervised classifier assigns a single PESTEL label per item (multi-label is feasible but not used here). Cluster-level PESTEL distributions are computed by aggregating item labels.

4 Time-Dependent Recursive Summary Graph (TRSG)

Goal. Replace a flat feed with a weekly, two-level structure that is compact, faithful and easy to compare across weeks.

Cumulative Knowledge Base. While the crawling process runs daily and TRSG graphs are materialized weekly, all ingested data is stored in a single, persistent vector knowledge base. Each new crawl extends this base rather than overwriting it. This design allows the system to reason over historical embeddings when constructing new weekly hierarchies, enabling the detection of longer-term phenomena such as cluster drift, gradual topic convergence or the emergence of entirely new abstract groupings. Over time, these evolving structures provide early hints of potential future trends rather than merely weekly snapshots.

4.1 Construction

L0→L1 (sub-clusters). Build an item-level similarity graph for the week using cosine similarity; run Leiden (Traag et al., 2019). Summarize each community with a *factual* prompt (names, dates, figures, relationships) and embed the resulting text as the L1 node.

L1→L2 (meta-clusters). Cluster L1 summaries and produce an *abstract* summary per meta-cluster (themes, trends, implications, minimal specifics). L2 nodes represent the week’s landscape.

Stability knobs. Small graphs use direct cosine; large graphs use FAISS with range thresholds. Weekly hierarchies are snapshotted for audit and fast reload.

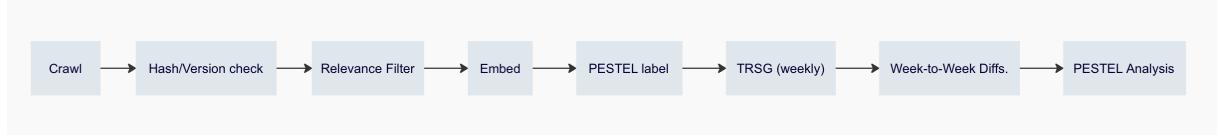


Figure 1: ORACLE workflow overview.

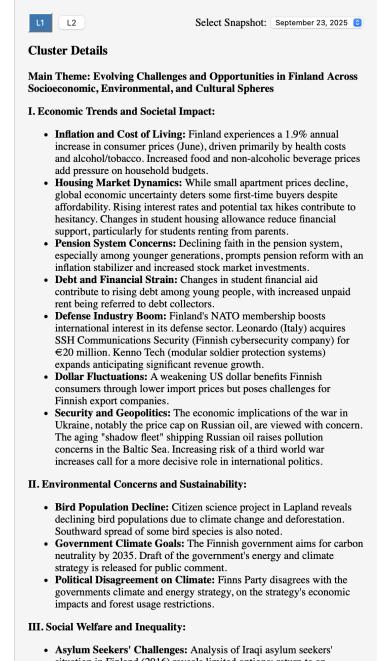


Figure 2: Example of the L1 layer in the TRSG. Each node represents a cluster of semantically related news items and edges indicate similarity links. The summarization step condenses each cluster into a factual thematic report, forming a connected graph of emerging narratives.

4.2 Prompt Design

To enforce consistent abstraction, TRSG uses level-specific prompts and recursive summarization when input exceeds model limits.

L1 – Thematic summaries. Summarizes factual content across related news items:

“Create a comprehensive L1 thematic summary in English. Do not evaluate or give suggestions. Identify the main theme, structure logically and include key facts—entities, dates, figures, policy changes and regional details. Write a complete, factual report without introductory remarks.”

This produces grounded, information-rich cluster summaries.

L2 – Strategic synthesis. Aggregates L1 outputs into cross-domain insights:

“Create a unified strategic L2 intelligence briefing in English. Do not evaluate or compare. Extract overarching patterns and systemic trends, emphasizing transformation forces across domains. Present as a coherent intelligence report without meta-commentary.”

L2 abstracts factual clusters into foresight-level narratives.

Recursive summarization. If a cluster’s combined text exceeds the model’s context limit, the content is split into balanced chunks and summarized recursively. Each batch is first summarized individually with the same prompt and those interim summaries are then re-summarized to produce the final L1 or L2 output. This hierarchical compression maintains completeness and coherence even for large clusters.

5 Week-to-Week Change Detection

Snapshots from consecutive weeks are compared at L1 and L2 using cosine similarity.

Matching. For each new summary, find the best old neighbour. Labels: **Stable** ($\text{sim} \geq 0.90$), **Changed** (0.70–0.90), **Added** (< 0.70). Unmatched old summaries are **Removed**. This yields structured deltas rather than vague impressions.

Component	Default / Behavior
Embedding model	text-embedding-3-small
L0→L1 threshold	0.75 (cosine)
L1→L2 threshold	0.55 (cosine)
Small- n fallback	Direct cosine (no FAISS)
Clustering	Leiden (modularity)
Summarization	Gemini 2.0 Flash (L1 factual / L2 abstract)
Persistence	Weekly snapshots (pickle)

Table 1: Key TRSG defaults used in production.

Theme grouping. Added/Removed lists are converted to human-readable themes: short micro-labels per text (LLM), then canonicalization via TF-IDF + agglomerative clustering (cosine distance). The result is a deterministic set of {label, added_texts, removed_texts} per level.

PESTEL analysis. Users select perspectives. For each theme, a schema-constrained analysis returns {title, analysis, level, group, importance [0,1]}. The results are cached in MySQL, keyed by week pair and perspective, to ensure reproducibility and fast retrieval.

6 Use Case: Curriculum Intelligence

An analyst monitoring Political+Technological developments compares weeks 23 and 28. TRSG highlights two new L2 themes: *EU digital skills funding* and *quantum computing policy momentum*. L1 reveals the concrete facts (program names, funding figures, named institutions). The PESTEL analysis recommends actions: (i) align elective modules with EU skill frameworks, (ii) add a quantum fundamentals stack (concepts + labs) and (iii) explore partnerships with local industry labs. The value is not prediction but *traceable synthesis* tailored to University’s remit.

In addition to supporting concrete curriculum adjustments, the TRSG output provides a reusable evidence base for cross-faculty coordination. Because every proposed action links back to the underlying news signals and cluster summaries, departments can justify decisions using a shared source of truth rather than ad-hoc interpretations or anecdotal reports. This auditability also supports institutional learning: when decisions are revisited months later, the exact informational context that motivated them remains inspectable. In practice, the platform enables routine, low-friction foresight workflows (monthly horizon scans, annual strategy cycles or accreditation preparations) without requir-

ing analysts to rebuild situational awareness from scratch.

The same machinery also generalizes to other decision layers, including research prioritization, stakeholder engagement and regional partnership planning. Since TRSG captures both stable background narratives and emerging weak signals, it helps distinguish between noise, structural change and transient bursts of attention. For universities operating in rapidly evolving technological and policy environments, this distinction is essential: long-horizon initiatives (e.g., lab infrastructure, degree redesign) demand evidence of durable trends, while quick interventions (e.g., micro-credential pilots) benefit from timely detection of nascent opportunities. By encoding both views into a single recurring graph, ORACLE makes such multiscale reasoning feasible for non-technical users.

7 Conclusion

ORACLE demonstrates that continuous, traceable foresight over fast-moving news streams is operationally achievable using a combination of embeddings, hierarchical clustering, recursive summarization and week-to-week change detection. By structuring evolving narratives into a two-level Time-Dependent Recursive Summary Graph (TRSG) and grounding interpretation in PESTEL, the platform delivers decision-ready intelligence that remains auditable and aligned with institutional needs. The curriculum-intelligence case shows that such a system can inform real strategic choices without relying on opaque prediction. Future work will refine evaluation, explore multi-label PESTEL, and extend the approach to multilingual sources and policy-science-industry link analysis.

8 Limitations

Our system has several inherent limitations:

- **Coverage bias.** The system depends on which news sources are crawled; underrepresented voices or niche media may be missed, skewing the narrative graph.
- **Summary hallucination.** Although prompts are guarded and ground traces are preserved, LLMs may still introduce inaccuracies or omit subtle but relevant facts.
- **Domain specificity.** The PESTEL classifier and thresholds are tuned for one university’s domain;

generalizing to another institution or domain will require re-training or re-tuning.

- **Temporal granularity.** Weekly snapshots may miss rapid developments or sub-week bursts; while daily crawling accumulates data, changes within a week are abstracted.

References

Khalid Alnajjar and Mika Hämäläinen. 2024. Mlpestel: the new era of forecasting change in the operational environment of businesses using llms. *Centria University of Applied Sciences*.

David M Blei and John D Lafferty. 2006. Dynamic topic models. In *Proceedings of ICML*, pages 113–120.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. In *Journal of Statistical Mechanics: Theory and Experiment*, volume 2008, page P10008. IOP Publishing.

Arun Chandrasekaran, George Thomas, and Reshma Nair. 2022. Forecasting and foresight: Systematic review of methods in ai-driven strategic analysis. *Technological Forecasting and Social Change*, 185:122095.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. In *Journal of Artificial Intelligence Research*, volume 22, pages 457–479.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of EMNLP*.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. arXiv preprint arXiv:2203.05794.

Mika Hämäläinen, Marita Huhtaniemi, and Leena Unkari-Virtanen. 2024. Suositus ennakkointitiedon johtamiseen. *Tulevaisuuden kudelmia. Ennakointikyvyyden kehittäminen Metropoliassa*, pages 50–63.

Han Huang, Tao Li, and Wei Xu. 2024. Streamsum: Streaming summarization of evolving news topics with large language models. In *Proceedings of EMNLP*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547. Initial version arXiv:1702.08734.

Jon Kleinberg. 2003. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of ACL*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of EMNLP*, pages 404–411.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of EMNLP-IJCNLP*, pages 3982–3992.

Vincent A Traag, Ludo Waltman, and Nees Jan van Eck. 2019. From louvain to leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1):5233.

Zilliz Wang and 1 others. 2021. Milvus: A purpose-built vector database for scalable similarity search. *arXiv preprint arXiv:2108.04535*.

Tamer Yüksel. 2012. Developing a scale for measuring the pestel dimensions in strategic management. *Procedia - Social and Behavioral Sciences*, 58:1091–1101.

Zaid Yusop. 2018. Pestel analysis. *Paper persented at COMRAP*, pages 34–39.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of ICML*.