Segment-Aware Analytics for Real-Time Editorial Support in Media Groups

Lessons from The Telegraph

Martin Spišák[1,*], Rodrigo Alves[1], Thomas Kelleher[2], Jason Sheppard[2], Ondřej Fiedler[1], Ergys Kosovrasti[1], Vojtěch Vančura[1], Petr Kasalický[1] and Pavel Kordík[1]

[1]*Recombee, Prague, Czech Republic*
[2]*The Telegraph, London, UK*

Abstract

We present a segment-aware analytics pipeline designed to support real-time editorial decision-making in digital media platforms. The core of our method combines large language model (LLM) embeddings with sparse autoencoders to extract interpretable, up-to-date segments from news articles. These segments are continuously refreshed and integrated into the recommendation platform, providing the foundation for analytics dashboards aligned with editorial needs. This demo paper describes our experience deploying the pipeline at *The Telegraph* and illustrates how advanced representation learning can bridge recommendation systems and editorial workflows in fast-paced news environments.

Keywords

Editorial Support, Media Groups, Item Segmentation, Sparse Autoencoders

1. Introduction

With millions of daily impressions, major digital media groups (like *The Telegraph*) depend on robust analytics to drive editorial decisions, optimize engagement, and power personalized recommendations at scale [1]. A key area of focus is online editorial news support [2, 3], where data-driven insights inform content curation, headline optimization, and article placement to better align with readers' interests and evolving consumption patterns. An important opportunity lies in effectively leveraging behavioral data to identify emerging trends – insights that translate into actionable strategies to boost user engagement while maintaining editorial integrity. This analytical foundation is highly useful for enabling adaptive, timely, and responsive personalization at scale [4].

However, identifying trends in real time presents significant challenges. Detecting temporal dynamics (such as sudden shifts in reader interest or rapidly evolving news segments) requires models that can process and interpret high-velocity data streams with minimal latency [5]. Additionally, making sense of a constantly growing corpus of text demands systems capable of understanding context, disambiguating meaning, and detecting subtle patterns across diverse topics and writing styles [6, 7]. These tasks are further complicated by the need for scalability, where algorithms must deliver accurate, timely insights across vast corpora of articles and impressions without compromising performance or reliability.

To address these challenges and enhance its personalization capabilities, *The Telegraph* is collaborating with *Recombee*, a leading provider of recommender as a service. *Recombee* offers advanced tools for real-time personalization, including support for **segments** – a flexible mechanism for partitioning items into meaningful, possibly overlapping clusters. More formally, segments represent dynamic

INRA 2025: 13th International Workshop on News Recommendation and Analytics, September 22–26, 2025, Prague, Czech Republic
*Corresponding author.
martin.spisak@recombee.com (M. Spišák); rodrigo.alves@recombee.com (R. Alves); thomas.kelleher@telegraph.co.uk (T. Kelleher); jason.sheppard@telegraph.co.uk (J. Sheppard); ondrej.fiedler@recombee.com (O. Fiedler); ergys.kosovrasti@recombee.com (E. Kosovrasti); vojtech.vancura@recombee.com (V. Vančura); petr.kasalicky@recombee.com (P. Kasalický); pavel.kordik@recombee.com (P. Kordík)
0009-0006-7763-5575 (M. Spišák); 0000-0001-7458-5281 (R. Alves); 0000-0003-2638-9969 (V. Vančura); 0000-0001-6438-366X (P. Kasalický); 0000-0003-1433-0089 (P. Kordík)

groupings of items based on, for instance, shared attributes such as topic, publication time, or editorial tags. Item segments can be defined manually by editors to reflect strategic content categories, or generated automatically through data-driven analysis. This flexibility makes segments valuable for enabling precise, faceted recommendations, delivering relevant experiences to diverse audiences, and maintaining editorial control with robust analytics.

In this paper, we demonstrate a newly developed analytical pipeline designed to (1) **detect trending segments** with high efficiency and scalability, thereby (2) **supporting editorial decision-making** with responsive insights. At its core, our pipeline integrates recent large language models (LLMs) with sparse autoencoders (SAEs) to disentangle polysemantic news embeddings into interpretable features that are used to group items into segments. By incorporating interaction data in the training process, the system is trained specifically to extract trending item segments, unlocking structured analytics that surface timely editorial insights.

1.1. Related Work

News is among the most extensively studied domains in recommendation systems and exhibits unique characteristics such as fast evolving article lifespan and relevance, global adoption of personalization, and the profound social implications of these systems [8, 9, 10, 11]. A wide range of methods has been proposed specifically for semantic grouping and trend detection in news analytics. Probabilistic models [12, 13] are widely used for uncovering evolving themes but suffer from interpretability issues. Traditional clustering- and graph-based methods [14] offer support for grouping and emerging story detection, but often lack granular semantic understanding. More recently, transformer-based topic modeling approaches [15, 16] leverage contextual embeddings to capture fine-grained topical structure, though their clusters may drift without additional constraints. Commercial platforms such as Chartbeat, NewsWhip, and Event Registry[1] similarly offer real-time topic monitoring by clustering content or assigning persistent tags and entity-based topic identifiers. These systems share our goal of providing interpretable, actionable, and up-to-date analytics for editorial support. In contrast, our approach (that is *integrated into the recommendation ecosystem*) combines LLM-based embeddings with an SAE to *efficiently* map articles to a fixed set of interpretable neurons, enhancing real-time trend detection.

Regarding **explanation and understanding in recommendation systems**, the literature generally presents two main approaches. The first adopts a more technical perspective, focusing on algorithmic explainability and interpretability of the recommendation mechanisms themselves [17, 18, 19]. The second approach is more practical (particularly in the context of news recommendation) and explores the integration of editorial teams into the recommendation loop [20, 21, 22], aiming to bridge the gap between technical infrastructure and editorial requirements. Our work follows this direction by empowering editorial teams to better understand the dynamics of item popularity in news recommendation.

Finally, to generate temporal segments, our pipeline employs **SAEs** [23, 24, 25, 26, 27], a technique that has recently attracted considerable attention in the machine learning community for its ability to produce interpretable and disentangled representations [23, 27]. Despite this growing interest, the use of sparse autoencoders within the domain of recommender systems is still in its early stages [28, 29, 30, 31]. To our knowledge, we are the first to propose a *temporally-aware* SAE training procedure designed specifically to capture emerging clusters from LLM-based content embeddings.

2. Methodology and Tech Stack

Our automatic segmentation pipeline comprises four steps, which we will further explain in the remainder of this section:

1. We train an SAE using article embeddings stored in a vector database. The training procedure employs *interaction-based sampling* to encourage the emergence of up-to-date topic structure within the active neurons of the SAE.

[1]https://chartbeat.com | https://newswhip.com | https://eventregistry.org

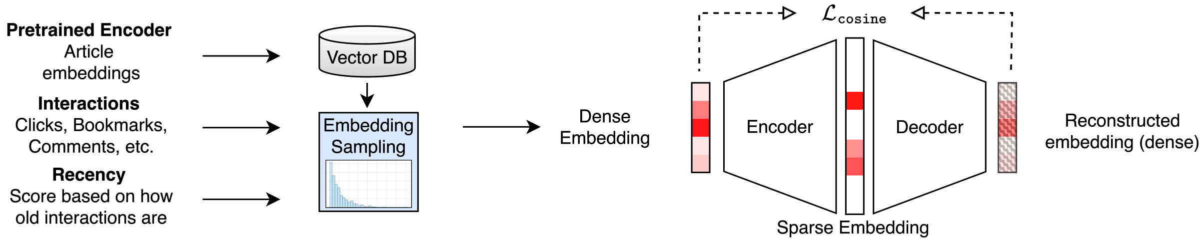

Figure 1: Sparse autoencoder (CompresSAE [31]) training using interaction-based sampling to emphasize recent trends.

2. The SAE encoder transforms article embeddings from dense representations into a small set of active neurons. By inverting this item-to-neuron mapping, we obtain (possibly overlapping) groups of items linked to each neuron: a semantic segmentation of articles.
3. We apply LLM-based post-processing to label each segment using metadata (e.g., headlines and summaries) from the associated articles.
4. Each segment is serialized as a query-language expression in order to enable its integration into analytics dashboards.

2.1. Sparse Autoencoder

Architecture. Our pipeline uses the CompresSAE [31], a sparse autoencoder architecture specifically designed for retrieval tasks with a focus on preserving directional information in sparse embeddings.

Given an input vector $\mathbf{x} \in \mathbb{R}^d$, CompresSAE encodes it into a k-sparse latent representation $\mathbf{s} \in \mathbb{R}^h$ using a non-linear encoder, then reconstructs it using a linear decoder f_{dec} defined as follows:

$$\mathbf{s} = f_{\text{enc}}(\mathbf{x}; \mathbf{W}_{\text{enc}}, \mathbf{b}_{\text{enc}}, k) = \phi(\mathbf{W}_{\text{enc}}\bar{\mathbf{x}} + \mathbf{b}_{\text{enc}}, k)$$
$$\hat{\mathbf{x}} = f_{\text{dec}}(\mathbf{s}; \mathbf{W}_{\text{dec}}) = \mathbf{W}_{\text{dec}}\mathbf{s}$$

Here, $\mathbf{W}_{\text{enc}} \in \mathbb{R}^{h\times d}$, $\mathbf{b}_{\text{enc}} \in \mathbb{R}^h$, and $\mathbf{W}_{\text{dec}} \in \mathbb{R}^{d\times h}$ (denoted jointly by θ) are learnable parameters of f_{enc} and f_{dec}, $\bar{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$ denotes input normalization, and $\phi(\cdot, k)$ is a sparsification function retaining the k entries largest in magnitude (zeroing out the rest) serving as the non-linear activation in the network. The decoder parameter matrix $\mathbf{W}_{\text{dec}}$ is row-normalized to maintain consistent scaling.

Distinct from prior sparse autoencoders trained via ℓ_2 reconstruction loss, CompresSAE minimizes the cosine distance between the input $\mathbf{x}$ and its reconstruction $\hat{\mathbf{x}}$:

$$\mathscr{L}_{\text{cosine}}(\mathbf{x}, f(\mathbf{x}; \theta, k)) = 1 - \frac{\mathbf{x}^\top \hat{\mathbf{x}}}{\|\mathbf{x}\|_2 \|\hat{\mathbf{x}}\|_2}$$

Due to space limitations, we refer to the CompresSAE article [31] for additional architectural details and motivations.

Unlike most SAEs [23, 24, 32, 25, 26] that are designed primarily for interpretability – aiming to learn monosemantic sparse representations – CompresSAE was originally developed for *embedding compression* to improve scalability of similarity search, offering embedding compression quality competitive with state-of-the-art methods [31]. Interestingly, the goal of compression appears to align closely with the monosemantic structure sought for interpretability [33], and the same architecture that enables compression also allows us to identify descriptive semantic features within the dense textual embeddings. This makes CompresSAE well-suited for preparing foundational representations applicable across downstream tasks ranging from retrieval to interpretability and analytics.

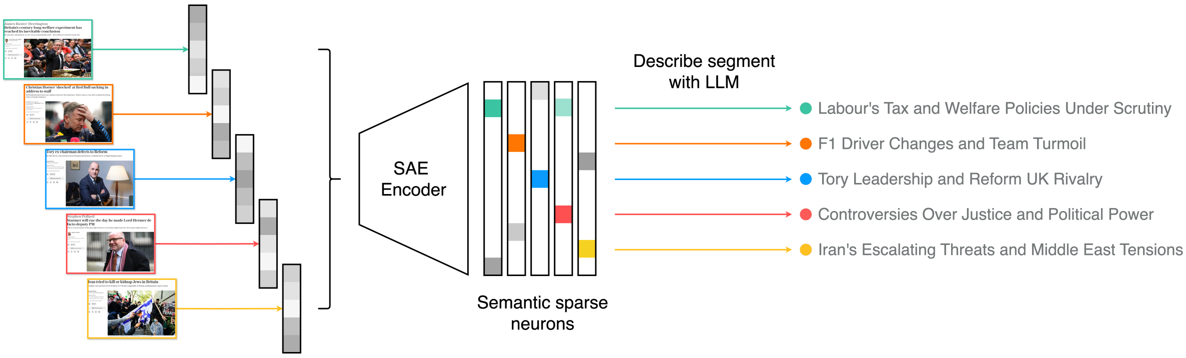

Figure 2: Passing a dense article embedding through a trained CompresSAE encoder reveals a disentangled set of interpretable features. From another perspective, these features correspond to emergent item segments.

Training. The training dataset consists of over 100,000 high-dimensional embeddings of complete news articles, computed using the Qwen3-Embedding-8B model [34]. The core methodological distinction from prior work [31, 23, 24, 32, 33] lies in our training procedure, illustrated in Figure 1. Rather than uniformly sampling item embeddings, we construct batches using **interaction-based sampling** that incorporates a time-decay function using interaction age. Before training, each article is assigned a real-valued score that combines two factors: the number of interactions it has received (more interactions yield higher scores) and the recency of those interactions (newer ones weigh more heavily). These scores are transformed into a categorical distribution via the softmax function. The dataloader then samples embeddings from this distribution to form training batches, instead of relying on uniform sampling. This strategy prioritizes the reconstruction quality of *recently trending* articles, thereby encouraging the activation of neurons associated with current topics.

Neuron Concept Naming. For each item, we select the top five strongest activating neurons, with each neuron representing a (possibly overlapping) cluster. To label a cluster, we gather all items for which this neuron ranks among the top five most active. We then use an LLM to generate a descriptive and characterizing summary or title for the cluster, based on the metadata of its associated items (e.g., article headlines). Figure 2 illustrates how CompresSAE transforms dense article embeddings into a sparse set of semantic neurons. In the center of the figure, each column represents one set, with two (of the five most active) neurons highlighted. By reading across rows, one can see which articles share a given neuron, effectively revealing item segments characterized by common themes. For example, a commentary piece on the Attorney General challenging the Labour Government (left-hand side of Figure 2, fourth row, red framed)[2] primarily activates a neuron linked to *Controversies Over Justice and Political Power* (fourth column, red), while simultaneously engaging a neuron associated with *Labour's Tax and Welfare Policies Under Scrutiny* (same column, green), which demonstrate the ability of our model to capture intersecting semantic dimensions within a single article, enabling fine-grained categorization that supports editorial insights.

2.2. Defining Item Segments

Once the SAE-identified item segments are labeled, we store them in the Item Segmentations format: a structure representing dynamic, potentially overlapping groups of items characterized by shared properties[3]. This structure allows for meaningful grouping of content, supporting interpretation, querying, and operational use across the recommendation and analytics ecosystem – including editorial dashboards. In practice, Segmentations make discovered topics directly usable across the recommendation stack, enhancing personalization capabilities while also supporting transparency and interpretability.

[2]https://www.telegraph.co.uk/news/2025/07/10/starmer-lord-hermer-veto-rule-by-lawyers/
[3]https://docs.recombee.com/segmentations

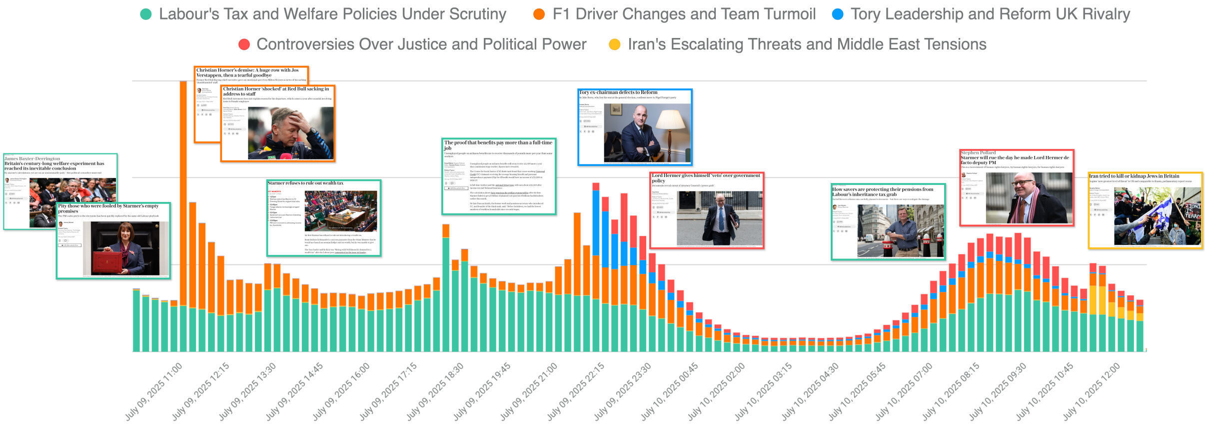

Figure 3: An Insights dashboard highlights diverse relevance dynamics of semantic segments. A blockbuster news event such as *F1 Driver Changes and Team Turmoil* triggers the rapid emergence of a segment that remains highly popular for several hours. In contrast, enduring storylines like *Labour's Tax and Welfare Policies Under Scrutiny* form segments that remain stable over the longer term, sustained by a continuous influx of related news stories.

To define a Segmentation, we leverage a dedicated query language built into our infrastructure: ReQL[4]. This feature-rich language supports lambda functions, geographical operations, and access to user interaction histories. It is extensively used within the *Recombee* ecosystem for defining filters, boosting specific types of content, and managing logic in recommendations. Item Segmentations can be directly generated from ReQL expressions. For example, a set of item segments automatically generated by our pipeline may yield a Segmentation definition like the following:

```
(if 'itemId' in {"T87VRz", "A4tbf", "A4s4Jd"}
    then {"Royal Weddings and Celebrity Celebrations"}
    else {}) +

(if 'itemId' in {"VwyFV", "aQ3Hyd", "Gsjvy"}
    then {"British Cuisine and Dining Trends"}
    else {}) +

(if 'itemId' in {"TzfX6", "nJLzM", "A4pRr"}
    then {"AI and Social Media Reshape Recruitment"}
    else {}) +

...
```

Currently, our automatic Segmentation pipeline (comprising SAE training, segment generation, and upload to the recommender ecosystem) runs every fifteen minutes, providing an effective balance between segment freshness and infrastructure load.

3. Practical Segment-Level Insights

In this section, we will demonstrate how our automatic semantic item segmentation pipeline is integrated into a real-time analytics tool and discuss how this empowers editorial teams at *The Telegraph* to intuitively analyze content performance, track emerging trends, and make data-informed decisions using interpretable topic clusters rather than abstract metrics.

The *Recombee* platform includes a real-time analytics tool called Insights[5], which allows analysts to build custom, interactive reports (tables, charts, etc.) directly on data stored within the recommender. It supports flexible data slicing, faceted filtering, and user-defined compound metrics, enabling rigorous examination of content performance, recommendation outcomes, and reader behavior without requiring data export to external

[4]https://docs.recombee.com/reql
[5]https://docs.recombee.com/insights

tools. This setup is particularly useful in the news domain, where tracking emerging trends depends heavily on freshness and responsiveness – advantages that are significantly diminished when relying on periodic data exports.

Within Insights, the identified semantic Segments are available as primary analytical dimensions, enabling editorial teams to, e.g.:

- aggregate recommended items by segment;
- compare segment-level readership, click-through, and conversion metrics;
- inspect time-series for individual segments.

A key application is identifying trending segments using an Insights dashboard, which ranks segments by the velocity of recent interactions (shown in Figure 3). This view highlights both explosive, short-lived topics (e.g., breaking sports news) and gradual, evergreen storylines whose incremental gains are easily obscured by headline-driven spikes elsewhere. Other dashboards examine conversion rates for each segment and highlight those performing in the top five percent of all segments. Such anomalies frequently reveal under-exploited themes that merit additional commissioning or promotional investment. This approach fundamentally diverges from traditional article-level performance tracking: while individual article metrics are often noisy and time-sensitive, segment-level insights provide a more stable, interpretable signal. This distinction benefits the *Telegraph* editorial team by enabling higher-level strategic decisions grounded in thematic engagement trends, rather than relying solely on the short-term success of individual pieces.

Lastly, by lowering the cognitive overhead associated with quantitative analysis, automatic semantic segmentation offers a more intuitive approach to trend identification than traditional metrics-based assessment and empowers non-technical newsroom staff with direct access to actionable summaries. Editorial teams can now adjust promotion rules, commissioning priorities, and publication timing on the basis of segment-level evidence – rather than the performance of individual articles, which may be subject to higher variance – thereby grounding editorial decisions in data while preserving the newsroom's strategic autonomy.

4. Key Takeaways

Sparse autoencoders have been established as powerful tools for extracting interpretable patterns from textual data [24, 35]. Recognizing their potential for news platform analytics, we extend this line of work by incorporating interaction data into SAE training, thereby redirecting the model's focus toward features associated with recently trending or popular news articles. The recovered item–feature relationships reveal dynamic semantic segments, unlocking insights that are quick to surface, easy to digest, and actionable.

The development process has revealed several challenges inherent to the SAE-driven segmentation paradigm – most notably, segment fragmentation and inconsistencies in naming accuracy and coherence. Although the behavior where a larger segment splits into several smaller ones is not necessarily undesirable – and can even shine light on *nuanced commonalities* among trending items – structuring these fragments into a coherent hierarchy would further enhance the usability of the Insights dashboards by supporting exploration across multiple levels of granularity. To address this, we are actively exploring ways to infuse hierarchical structure into the CompresSAE architecture via auxiliary objectives (e.g., [36]), as well as strategies for hierarchy-aware post-processing and labeling.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT and Grammarly in order to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] K. Aishwarya, V. Bhagwat, H. Behra, I. Khurana, G. Jhangiani, G. Bhatia, User preference recommendation system and analytics for news articles, in: International Conference on Information and Communication Technology for Intelligent Systems, Springer, 2023, pp. 183–193.

[2] J. Stray, Editorial values for news recommenders: Translating principles to engineering, in: News quality in the digital age, Routledge, 2023, pp. 151–165.

[3] E. Entrup, R. Ewerth, A. Hoppe, Can editorial decisions impair journal recommendations? analysing the impact of journal characteristics on recommendation systems, in: Proceedings of the 18th ACM Conference on Recommender Systems, 2024, pp. 1062–1066.
[4] Q. Zhang, J. Zhu, J. Sun, G. Cai, R. Yu, B. He, L. Li, Enhancing news recommendation with real-time feedback and generative sequence modeling, in: Proceedings of the Recommender Systems Challenge 2024, RecSysChallenge '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 32–36. URL: https://doi.org/10.1145/3687151.3687158. doi:10.1145/3687151.3687158.
[5] R. Alves, A. Ledent, R. Assunção, P. Vaz-De-Melo, M. Kloft, Unraveling the dynamics of stable and curious audiences in web systems, in: Proceedings of the ACM Web Conference 2024, 2024, pp. 2464–2475.
[6] R. Wang, V. Liesaputra, Z. Huang, A survey on llm-based news recommender systems, arXiv preprint arXiv:2502.09797 (2025).
[7] S. Gao, J. Fang, Q. Tu, Z. Yao, Z. Chen, P. Ren, Z. Ren, Generative news recommendation, in: Proceedings of the ACM Web Conference 2024, 2024, pp. 3444–3453.
[8] S. Raza, C. Ding, News recommender system: a review of recent progress, challenges, and opportunities, Artificial Intelligence Review (2022) 1–52.
[9] N. Helberger, On the democratic role of news recommenders, in: Algorithms, automation, and news, Routledge, 2021, pp. 14–33.
[10] S. Flaxman, S. Goel, J. M. Rao, Filter bubbles, echo chambers, and online news consumption, Public opinion quarterly 80 (2016) 298–320.
[11] N. Muralidhar, H. Rangwala, E.-H. S. Han, Recommending temporally relevant news content from implicit feedback data, in: 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), IEEE, 2015, pp. 689–696.
[12] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[13] D. M. Blei, J. D. Lafferty, Dynamic topic models, in: Proceedings of the 23rd international conference on Machine learning, ACM, 2006, pp. 113–120.
[14] J. Allan, J. Carbonell, G. Doddington, J. Yamron, Y. Yang, Topic detection and tracking pilot study final report, in: Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, 1998, pp. 194–218.
[15] M. Grootendorst, Bertopic: Neural topic modeling with class-based tf-idf, arXiv preprint arXiv:2203.05794 (2022). URL: https://arxiv.org/abs/2203.05794.
[16] Y. Boutaleb, Y. Zhang, W. Lu, M. Zhang, Bertrend: Dynamic topic modeling with weak signals for event detection in news streams, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023, pp. 1618–1622.
[17] Y. Zhang, X. Chen, et al., Explainable recommendation: A survey and new perspectives, Foundations and Trends® in Information Retrieval 14 (2020) 1–101.
[18] A. Vultureanu-Albişi, C. Bădică, Recommender systems: An explainable ai perspective, in: 2021 International conference on innovations in intelligent systems and applications (INISTA), IEEE, 2021, pp. 1–6.
[19] J. Šafařík, V. Vančura, P. Kordík, Repsys: Framework for interactive evaluation of recommender systems, in: Proceedings of the 16th ACM Conference on Recommender Systems, 2022, pp. 636–639.
[20] F. Lu, A. Dumitrache, D. Graus, Beyond optimizing for clicks: Incorporating editorial values in news recommendation, in: Proceedings of the 28th ACM conference on user modeling, adaptation and personalization, 2020, pp. 145–153.
[21] C. Peukert, A. Sen, J. Claussen, The editor and the algorithm: Recommendation technology in online news, Management science 70 (2024) 5816–5831.
[22] B. Mahmood, M. Elahi, S. Touileb, L. Steskal, C. Trattner, Incorporating editorial feedback in the evaluation of news recommender systems, in: Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization, 2024, pp. 148–153.
[23] T. Bricken, A. Templeton, J. Batson, B. Chen, A. Jermyn, T. Conerly, N. Turner, C. Anil, C. Denison, A. Askell, et al., Towards monosemanticity: Decomposing language models with dictionary learning, Transformer Circuits Thread 2 (2023).
[24] L. Gao, T. D. la Tour, H. Tillman, G. Goh, R. Troll, A. Radford, I. Sutskever, J. Leike, J. Wu, Scaling and evaluating sparse autoencoders, 2024. URL: https://arxiv.org/abs/2406.04093. arXiv:2406.04093.
[25] B. Bussmann, P. Leask, N. Nanda, BatchTopK Sparse Autoencoders, 2024. URL: http://arxiv.org/abs/2412.06410. doi:10.48550/arXiv.2412.06410, arXiv:2412.06410 [cs].
[26] S. Rajamanoharan, T. Lieberum, N. Sonnerat, A. Conmy, V. Varma, J. Kramár, N. Nanda, Jumping Ahead: Improving Reconstruction Fidelity with JumpReLU Sparse Autoencoders, 2024. URL: http://arxiv.org/abs/2407.14435. doi:10.48550/arXiv.2407.14435, arXiv:2407.14435 [cs].
[27] H. Cunningham, A. Ewart, L. Riggs, R. Huben, L. Sharkey, Sparse autoencoders find highly interpretable

features in language models, 2023. URL: https://arxiv.org/abs/2309.08600. arXiv:2309.08600.
[28] M. Ahmadian, M. Ahmadi, S. Ahmadian, S. M. J. Jalali, A. Khosravi, S. Nahavandi, Integration of deep sparse autoencoder and particle swarm optimization to develop a recommender system, in: 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2021, pp. 2524–2530.
[29] M. Spišák, R. Bartyzal, A. Hoskovec, L. Peska, M. Tůma, Scalable approximate nonsymmetric autoencoder for collaborative filtering, in: Proceedings of the 17th ACM Conference on Recommender Systems, RecSys '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 763–770. URL: https://doi.org/10.1145/3604915.3608827. doi:10.1145/3604915.3608827.
[30] J. Wang, X. Zhang, W. Ma, M. Zhang, Interpret the internal states of recommendation model with sparse autoencoder, arXiv preprint arXiv:2411.06112 (2024).
[31] P. Kasalický, M. Spišák, V. Vančura, D. Bohuněk, R. Alves, P. Kordík, The future is sparse: Embedding compression for scalable retrieval in recommender systems, 2025. URL: https://arxiv.org/abs/2505.11388. arXiv:2505.11388.
[32] J. Wang, X. Zhang, W. Ma, M. Zhang, Interpret the Internal States of Recommendation Model with Sparse Autoencoder, 2024. URL: http://arxiv.org/abs/2411.06112. doi:10.48550/arXiv.2411.06112, arXiv:2411.06112 [cs].
[33] T. Wen, Y. Wang, Z. Zeng, Z. Peng, Y. Su, X. Liu, B. Chen, H. Liu, S. Jegelka, C. You, Beyond matryoshka: Revisiting sparse coding for adaptive representation, 2025. URL: https://arxiv.org/abs/2503.01776. arXiv:2503.01776.
[34] Y. Zhang, M. Li, D. Long, X. Zhang, H. Lin, B. Yang, P. Xie, A. Yang, D. Liu, J. Lin, F. Huang, J. Zhou, Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025. URL: https://arxiv.org/abs/2506.05176. arXiv:2506.05176.
[35] A. Templeton, T. Conerly, J. Marcus, J. Lindsey, T. Bricken, B. Chen, A. Pearce, C. Citro, E. Ameisen, A. Jones, et al., Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. transformer circuits thread, 2024.
[36] V. Zaigrajew, H. Baniecki, P. Biecek, Interpreting clip with hierarchical sparse autoencoders, arXiv preprint arXiv:2502.20578 (2025).