UNIVERSITAT DE BARCELONA

FUNDAMENTAL PRINCIPLES OF DATA SCIENCE MASTER'S THESIS

Analysis of Content Diversity in News Recommendation Systems

Author:
Rubén BLANCO BORRÁS

Supervisor:
Oriol Pujol

A thesis submitted in partial fulfillment of the requirements for the degree of MSc in Fundamental Principles of Data Science

in the

Facultat de Matemàtiques i Informàtica

January 17, 2026

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

Master in Fundamental Principles of Data Science

Analysis of Content Diversity in News Recommendation Systems

by Rubén BLANCO BORRÁS

This Master's Thesis studies informational diversity in news recommendation systems from a dual perspective: producer diversity and consumer-perceived diversity. The main objective is to analyze which diversity metrics are most suitable for evaluating journalistic content and how they can be applied to different datasets.

In a first phase, a study is conducted on various diversity metrics used in recommendation systems, taking as reference those proposed in the literature associated with Microsoft and using the *MIND Small* dataset. This analysis allows the evaluation of classical diversity and coverage metrics applied to news recommendation, as well as an understanding of their advantages and limitations in controlled environments.

In a second phase, these metrics are adapted and applied to a news dataset from the media outlet *3Cat*, aiming to evaluate the diversity of political topics present in the published content. In this context, a distinction is made between producer diversity, related to the ideological and thematic variety of the generated content, and consumer diversity, modeled through an activation metric that approximates the diversity effectively perceived by the reader.

The results allow for a comparison of both diversity perspectives and the analysis of potential mismatches between the informational supply and the diversity experienced by the user. The code developed for data processing and metric calculation can be found in the project repository (Blanco Borrás, 2026).

Acknowledgements

I would like to express my gratitude to my supervisors, Oriol Pujol and Jordi Vitrià, for their guidance, availability, and valuable contributions throughout the development of this project. Their expertise and judgment have been fundamental for the completion of this work.

I would also like to thank my mother for her constant support and encouragement during the Master's program, without which it would not have been possible to complete this academic stage.

Chapter 1

Introduction

Diversity is a central concept in the study of news recommendation systems, but also one of the most difficult to define precisely and consistently. In the fields of media and data science, the term *diversity* is used to refer to a wide variety of distinct phenomena, ranging from the plurality of topics and viewpoints to the distribution of attention among content producers. This lack of a single definition has historically hindered both its measurement and its systematic incorporation into algorithmic systems.

Recommendation systems today play a fundamental role in information access, as they mediate which content is shown to users on digital platforms. One of the classic problems associated with these systems arises when they present a limited range of content, either by prioritizing only the most popular items or those similar to what the user has previously consumed. This behavior can result in impoverished informational environments, commonly described as filter bubbles or echo chambers, and has motivated growing interest in the study of algorithmic diversification mechanisms.

From academic literature and real system design, diversity can be approached from multiple perspectives. On the one hand, it can be understood as diversity from the consumer's point of view, that is, the variety of content to which a user is exposed. This conception includes aspects such as topical diversity, exposure to different viewpoints, novelty, or serendipity, aiming to avoid monotony and overfitting to past preferences. On the other hand, diversity can also be analyzed from the producer's perspective, considering how visibility is distributed among different creators or content sources, and preventing attention from concentrating only on a small set of dominant actors.

The relevance of diversity in media goes beyond technical considerations. From a social and democratic perspective, informational diversity is associated with forming informed citizens, fostering plural public debate, and preventing media concentration dynamics. However, different normative approaches to the functioning of democracy lead to different interpretations of what type of diversity is desirable and how it should be evaluated. This plurality of objectives explains the coexistence of multiple metrics and approaches for its analysis.

Digital platforms, in turn, incorporate diversification mechanisms not only for social reasons but also for practical motivations. Exposure to varied content can improve long-term metrics such as user retention, reduce fatigue associated with repetitive recommendations, and support the sustainability of the creator ecosystem. In practice, most recommendation systems implement diversification algorithms at re-ranking stages, balancing diversity with other objectives such as accuracy or engagement.

In this context, there is no single optimal diversity metric nor a universally accepted standard for its evaluation. Available metrics depend on the notion of similarity between items, the level of aggregation considered, and whether the analysis is conducted from the consumer's or producer's perspective. This multiplicity of approaches makes a careful analysis necessary to determine which metrics are appropriate in each scenario and what type of diversity they are actually capturing.

The objective of this Master's Thesis is to analyze and evaluate different diversity metrics applied to news recommendation systems, explicitly distinguishing between producer diversity and consumer-perceived diversity. In a first phase, widely used diversity metrics in the recommendation systems literature are studied, using the *MIND Small* dataset proposed by Microsoft as a reference environment. In a second phase, these metrics are adapted and applied to a subset defined by a temporal range of the *3Cat* news dataset, in order to evaluate the diversity of political topics in the published content along with an activation metric that approximates the diversity experienced by the reader, ultimately combining them into a single objective value that can be used to evaluate diversity in a less biased manner by taking into account both the producer and the consumer.

Ultimately, this work aims to contribute to a more nuanced understanding of diversity in news recommendation systems, highlighting the differences between the informational supply and the diversity effectively perceived, and emphasizing the importance of selecting metrics that are coherent with the social and technical objectives of the analyzed system.

Chapter 2

Train Models

This chapter describes the training process of the recommendation models used in this work. The main objective is to generate trained models that allow replicating and understanding the diversity metrics proposed by the authors of the *RADio paper* (Vrijenhoek et al., 2024). Through this process, we aim to analyze how these metrics behave in a controlled news recommendation environment and establish a solid basis for their subsequent application and adaptation to other datasets.

2.1 Results and Comparative Analysis

This section presents a detailed analysis of the results obtained in recommendation by applying different recommendation algorithms on the 'small MIND dataset' (MSNews, 2020) in this work compared to the values published in *RADio*. Two training configurations were considered: one with two *epochs* and another with four *epochs*. For each model, the highest performance between both configurations is selected and used as the final reference. For comparison purposes, a table is included that summarizes the RADio values along with those obtained in both training configurations, highlighting the best result for each architecture.

2.1.1 Direct Comparison between RADio and this Work

Table 2.1 shows the NDCG@10 values reported by RADio and those obtained in this project after training the models for two and four *epochs*. Using a single comparative table allows for quick visualization of differences, the evolution with additional training, and the proximity of the resulting values to those published by RADio. Each row highlights in color the best result among the two training configurations applied in this work.

Algorithm	**RADio**	**2 epochs**	**4 epochs**
LSTUR	0.4134	0.4074	0.3943
NAML	0.4091	0.4019	0.4080
NPA	0.4068	0.3561	0.3306
NRMS	0.4163	0.3967	0.4006
Most Popular	0.2750	0.2882	0.2882
Random	0.2949	0.3263	0.3263

TABLE 2.1: Comparison of NDCG@10 between RADio and the models trained in this work. The best value obtained between 2 and 4 *epochs* is highlighted.

2.1.2 Selection of Optimal Training

Since the models were trained with two different configurations (2 and 4 *epochs*), the criterion adopted for final evaluation is to select, for each model, the highest value among both. This allows a fair assessment of which architecture converges faster and which gains real benefit from extended training.

The observed patterns are as follows:

- **LSTUR:** slightly worsens from 0.4074 to 0.3943, being similar to the reference target set by RADio.
- **NRMS:** shows a clear trend of benefiting from additional training, achieving its best performance with four epochs: 0.4006.
- **NAML:** achieves its best value with four epochs (0.4080), almost identical to that published by RADio.
- **NPA:** is the model most affected by epoch variation: it drops from 0.3561 to 0.3306, indicating no benefit from additional training.

Consequently, each model is analyzed based on its best result, as shown in the table above, and for subsequent predictions, the model with the most optimal number of epochs will be used.

2.1.3 Comparative Analysis

Although the models were trained only for two and four *epochs*, the results obtained are remarkably close to the values published by RADio, especially for the more robust models. This suggests that part of the representational capacity of these architectures is acquired quickly, even with limited training, which is consistent with previous studies on attention-based architectures.

2.1.4 NAML: Practically Identical Results

The NAML model achieves a value of **0.4080** with 4 *epochs* in this work, practically identical to 0.4091 reported by RADio and to the 2 *epochs* value (0.4019). This may be due to:

- **Architectural robustness:** the hierarchical attention structure and use of multiple embeddings allow capturing useful patterns from early training phases.
- **Fast convergence:** word- and news-level attention models typically learn relevant relationships in a few iterations.
- **Broad semantic information representation:** combining title, body, category, and subcategory provides greater contextual diversity.

This behavior indicates that NAML is particularly efficient and stable, even with limited training.

2.1.5 NRMS and LSTUR: Moderate Differences

The values obtained for NRMS and LSTUR show that each model reaches its best performance under different configurations:

- NRMS: improves to **0.4006** with 4 *epochs* (vs. 0.3967 with 2 *epochs* and 0.4163 in RADio).
- LSTUR: achieves its best value with 2 *epochs*, **0.4074** (vs. 0.3943 with 4 *epochs* and 0.4134 in RADio).

The differences can be explained by:

1. NRMS may require more epochs to stabilize parameters.
2. LSTUR depends on the combination of long- and short-term user representations, making performance decrease if training is prolonged too much.
3. NRMS, based purely on self-attention, improves with additional training, though it does not reach RADio's reference value.

2.1.6 NPA: The Most Epoch-Sensitive Model

NPA shows the largest relative difference between training configurations: **0.3561** with 2 *epochs* vs. 0.3306 with 4 *epochs* (RADio: 0.4068). This reflects its internal properties:

- Uses personalized word- and news-level attention.
- Requires a significant volume of user interactions to stabilize personalized embeddings.
- Is more susceptible to premature overfitting when trained without sufficient regularization or data.

Therefore, increasing to four epochs not only does not improve performance but may even degrade it.

2.1.7 General Trend Analysis

The joint analysis of RADio and the results of this work allows observing different patterns:

- **Models with greater multi-scale or hierarchical attention capacity stabilize earlier:** LSTUR and NRMS show relatively consistent values close to the best performance among configurations (LSTUR: 0.4074; NRMS: 0.4006).
- **Models dependent on personalized representations require more iterations:** NAML and NPA show larger variations between 2 and 4 *epochs* (NAML: 0.4019 $\rightarrow$ 0.4080; NPA: 0.3561 $\rightarrow$ 0.3306), reflecting sensitivity to the number of epochs.
- **Performance between 2 and 4 *epochs* does not always increase; in some models, it even decreases:** clearly observable in NPA and partially in LSTUR, where values decline if training exceeds the optimal duration.

2.1.8 Explanation of the Models

Below is a description of the models used in this news recommendation work:

- **Random:** generates a random order of recommendations. This model acts as a baseline reference value that other models must exceed.
- **Most Popular:** ranks news according to total number of views, prioritizing the most popular items.
- **LSTUR (Neural News Recommendation with Long- and Short-term User Representations):** Captures both long-term user preferences and short-term interests. It:
  - Uses user ID embeddings to learn long-term representations.
  - Processes recently read news through a GRU to learn short-term representations.
- **NRMS (Neural News Recommendation with Multi-Head Self-Attention):** A content-based news recommendation approach. Its main functionality includes:
  - Using multi-head self-attention to learn news representations, modeling interactions between words.
  - Learning user representations by capturing relationships among the news they have read.
  - Applying additive attention to select the most important words and news for informative representations.
- **NAML (Neural News Recommendation with Attentive Multi-View Learning):** A multi-view neural approach integrating diverse news and user information:
  - Uses title, body, category, and subcategory to obtain news representation.
  - Uses user behavior history to learn their representation.
  - Applies additive attention to learn informative representations, selecting the most relevant words and news.
  - Due to legal restrictions on the *MIND dataset*, summaries are used instead of full news bodies.
- **NPA (Neural News Recommendation with Personalized Attention):** A content-based method using personalized attention to improve news and user representations:
  - Learns news representations through a CNN.
  - Learns user representations from the news they clicked.
  - Applies personalized word-level attention to highlight important words for each user.
  - Applies personalized news-level attention to emphasize historically relevant news for each user.

Chapter 3

Diversity Metrics

Diversity in recommendation systems is a complex and multifaceted concept that can be approached from different perspectives, including consumer-perceived diversity and producer diversity. To evaluate and compare the effectiveness of recommendation models, it is necessary to use metrics that quantify these different dimensions of diversity. This section presents the metrics selected for this work, describing their theoretical foundation, practical application, and how they allow measuring both the thematic and viewpoint variety to which a user is exposed, as well as the distribution of attention among different content producers. The analysis of these metrics provides the basis for understanding how recommendations can promote a more diverse and balanced informational environment.

A few modifications were made compared to the base code in the `WikidataHandler` class and in `representation.py` to prevent some errors. The rest is identical to the base repository, except for the prior file generation part, as these files are no longer available online and had to be generated from scratch.

3.1 Explanation of the Metrics

This work uses several diversity metrics inspired by *RADio*, which attempts to capture five normative concepts in news recommendation: *Calibration* (*topics* and *complexity*), *Fragmentation*, *Activation*, *Representation*, and *Alternative voices*.

To measure diversity, article distributions are compared using two classical divergence metrics: KL-divergence and Jensen-Shannon Divergence (JSD). KL-divergence measures how much one distribution differs from another but can be unstable if there are zeros in the reference distribution. Therefore, JSD is used, which is bounded between 0 and 1 and does not explode when some probabilities are zero.

- JSD $= 0$ indicates that the compared distributions are identical.

- JSD $\rightarrow 1$ indicates that the distributions are completely different.

For all metrics, the interpretation of high or low values depends on the context:

- For metrics comparing with the impression of articles available to the user for clicking (*Activation*, *Representation*, *Alternative voices*), low values are better, as they indicate that the recommendation does not amplify biases and reflects the available content in a balanced way.

- For *Fragmentation*, a high value is positive, indicating that different users receive different recommendations, increasing per-user diversity.

- For *Calibration* (*topics* and *complexity*), low values are desired, indicating that the recommendation faithfully reflects the user's interests and reading complexity without relevant variations or biases.

3.1.1 Calibration (topics)

This metric evaluates the topical calibration of recommendations by calculating the frequency of each article's *category* in the recommended list and comparing this distribution with the user's *reading history*. Its goal is to measure whether the topic distribution in recommendations adequately reflects the user's interests.

The associated code simply counts the proportion of articles in each *category* and optionally weights these values according to the ranking position (higher-ranked articles have more weight). This aligns with the theoretical intent of the metric: ensuring that the topics most important to the user are proportionally represented in the most visible positions.

3.1.2 Calibration (complexity)

This metric measures the diversity in reading complexity of recommended articles, evaluated using the Flesch-Kincaid Reading Ease index:

$$\text{Reading Ease} = 206.835 - 1.015 \cdot \frac{\text{total words}}{\text{total sentences}} - 84.6 \cdot \frac{\text{total syllables}}{\text{total words}}$$

Where:

- Long sentences and words with many syllables decrease the score (more difficult text).

- Short sentences and simple words increase the score (easier text).

The metric calculates the reading complexity of recommended articles and optionally gives more importance to articles appearing in top ranking positions. It compares this distribution with the user's *reading history* to ensure that recommendations are similar to what the user typically reads in terms of complexity.

3.1.3 Fragmentation

The theoretical intent of *Fragmentation* is to measure whether recommended news for a user covers multiple distinct events or if several articles address the same event.

In the implementation, the recommendation stories for a user are compared with 5 *pools* of articles shown to other users using the same recommendation algorithm.

- Low value $\rightarrow$ multiple users receive similar recommendations (low inter-user diversity).

- High value $\rightarrow$ different users receive distinct recommendations (positive diversity).

3.1.4 Activation

Activation quantifies the variety of affective intensity of recommended articles using the absolute value of a sentiment analysis score. Recommended articles are compared with the pool of articles shown to the user.

- Low value $\rightarrow$ the recommendation maintains a similar sentiment distribution relative to the impression shown to the user.

- High value $\rightarrow$ the distributions differ, indicating some bias.

3.1.5 Representation

Representation analyzes the presence of political actors in articles and evaluates fairness in representation. In this work, it focuses on the U.S. with the *MIND Dataset*, considering only two parties: Republican and Democrat, calculating the percentage of mentions of each. Recommended articles are compared with the pool of articles shown to the user. Low JSD values indicate that the recommended content preserves the distribution of the user impression and does not add potential biases.

3.1.6 Alternative voices

Alternative voices aims to measure the presence of minority versus majority voices. In the theoretical implementation, a minority is any person identified by the NLP pipeline who does not have a Wikipedia or Wikidata page.

Recommended articles are compared with the pool of articles shown to the user. Low JSD values indicate that the recommended content preserves the distribution of the user impression and does not add potential biases.

At the implementation level, it is unrelated to the above, as it simply measures how many people have a *givenname* in their Wikidata information, and since the vast majority do (28,590 out of 29,977), the obtained metric values are very low, limiting its practical usefulness due to algorithmic limitations, as the dataset itself already identifies people and their corresponding Wikidata *QID*. For the metric to be useful, an NLP algorithm would need to identify people in the body of each news article (this part is absent in the *MIND Dataset* due to legal restrictions) and determine whether they exist in Wikipedia, creating a new *entities* column instead of using the default one where all have QID, but this is beyond the scope of this work, making the metric of limited practical value.

3.1.7 Potential Improvement: Internal Diversity

Instead of comparing with an external *pool*/history/*impression sample*, internal diversity of the recommendation could be measured using:

- Normalized Shannon entropy, where values close to 1 indicate maximum entropy/diversity and 0 minimum.

- ILS Diversity, based on similarity between articles within the recommendation, also normalized between 0 and 1, where values close to 1 indicate maximum diversity and 0 minimum.

This would allow evaluating the "pure" diversity of the recommended list, independent of the available pool diversity. Current metrics (KL or JSD) measure how much the recommendation preserves or distorts the original content distribution, but do not measure the actual internal diversity of the list. If the *pool*/history/*sample* is not diverse, comparing with it is meaningless. It does not measure pure diversity but rather relative to available content.

3.1.8 Classification of Metrics According to TechPolicy

According to the TechPolicy article (TechPolicy Press, 2023), diversity metrics can be classified into four main categories:

- **Consumer Diversity:** reflects the user experience regarding the variety of content they consume. Example: *Calibration (topics)*, *Calibration (complexity)*, *Activation*.

- **Producer Diversity:** measures whether different producers or sources have equitable presence in recommendations. Example: *Representation*.
- **Content-/item-based Diversity:** evaluates the heterogeneity of recommended items based on attributes such as topic, complexity, or news type. Example: *Calibration (topics)*, *Calibration (complexity)*, *Fragmentation*.
- **Society- or platform-level Diversity:** analyzes the representation of different social, political, or cultural groups within the recommendation set. Example: *Representation*, *Alternative voices*.

The implemented metrics allow evaluating different aspects of diversity in news recommendations: from the user perspective (*Calibration (topics)*, *Calibration (complexity)*, *Activation*), the producer perspective (*Fragmentation*), and society-level (*Representation* and *Alternative voices*). The implementation generally aligns with the theoretical explanation of each metric, except *Alternative voices*, which has practical limitations.

3.2 Comparison of Metrics with RADio

This section compares the results obtained in this work on the *MIND small dataset* with those reported by the RADio repository on GitHub (Vrijenhoek, 2020), not from their paper, as values differ since the *paper* used the *large dataset* with many more iterations, and the repository used the *demo dataset*, even smaller than the *small dataset*. Values calculated in this work correspond to a subset of 10,000 users from the full test set, which contains over 73,000 users. Parts of another repository (Halwesit, 2020) were also used in the implementation. To facilitate comparison and obtain more intuitive values, only *Jensen-Shannon divergence* is used, as it is normalized and allows direct distribution comparison.

3.2.1 Metric Ranges

Metric	RADio (min-max)	This work (min-max)
Calibration Topic	0.000 – 0.994	0.000 – 0.994
Calibration Complexity	0.000 – 0.994	0.000 – 0.994
Fragmentation	0.355 – 0.994	0.440 – 0.994
Activation	0.000 – 0.982	0.000 – 0.982
Representation	0.000 – 0.948	0.000 – 0.914
Alternative Voices	0.000 – 0.608	0.000 – 0.138

TABLE 3.1: Metric value ranges for RADio and this work.

3.2.2 Metric Comparison by Model

The average metrics for each model, comparing results obtained in this work with those published by *RADio*, are presented in **Appendix B** (values rounded to 3 decimals). The comparison is performed on a subset of 10,000 users from the test set (over 73,000 *behaviors*).

3.2.3 Comparative Analysis

Comparing each model with *RADio* reveals the following trends:

- **Calibration Topic and Complexity:** In general, values in this work are slightly lower than *RADio* but maintain the same relative trend across models. This indicates that topic distribution and article complexity are comparable.
- **Fragmentation:** In all models, obtained values are higher than those reported by *RADio*, suggesting that recommendations in this work tend to concentrate on a few stories per user, showing lower event diversity within each list.
- **Activation:** Mean values are slightly lower than *RADio*, indicating that the emotional intensity of recommended articles is slightly more moderate.
- **Representation:** This work shows significantly lower values than *RADio*, reflecting reduced presence of political actors in recommendations.
- **Alternative Voices:** Values are very low compared to *RADio*, consistent with implementation limitations: most identified people have Wikidata pages, reducing the proportion of alternative voices. Differences between *RADio* and this work arise from different preprocessing of the *MIND Dataset* in generating the *articles.entities* column in this work compared to RADio.

Overall, although there are differences in *Fragmentation*, *Representation*, and *Alternative Voices*, the general metric trends are maintained and allow coherent comparison of this work's results with RADio's on a representative subset of the dataset.

Chapter 4

3Cat Dataset - Analysis of Political Diversity and Sentiment

This chapter describes the process of analyzing political diversity with its associated sentiment carried out on the *3Cat* dataset. The objective is to quantitatively evaluate the ideological diversity of news published by this outlet and its affective activation in the reader, using a natural language processing approach with two of the diversity metrics discussed earlier, adapted to this case.

4.1 Dataset Selection

For this study, a full month from the 3Cat dataset was selected in order to analyze the political diversity of the news. Due to computational limitations, a sample of 500 news articles was used, which allowed optimization of calculations and reduced processing time. However, with more powerful hardware resources, it would be possible to process the entire month without sampling.

4.2 NLP Processing and Entity Extraction

A natural language processing pipeline was built to identify, within each news article, entities of type *political person* and *political party*. Each entity was assigned a normalized sentiment value in the range $[-1, +1]$, corresponding to the context in which it appears in the news. For this, a local language model, Ollama Qwen3:8b, was used, taking approximately 30 seconds per news article on a mid-range 2025 computer.

Subsequently, the entities were processed to normalize their format, removing capital letters, accents, and other special characters to facilitate subsequent processing and comparison with external lists.

4.3 Obtaining Reference Lists from Wikidata

To correctly classify the detected entities, the following were downloaded from Wikidata:

- The list of political parties in Spain, along with their ideology (left, center, right, far-left/far-right) and official abbreviations.
- The list of political persons in Spain, including ministers, deputies, and other relevant positions.

These lists allowed verification of whether the extracted text entities corresponded to Spanish parties or political persons, and in the latter case, to associate them with their political party whenever possible.

4.4 Ideological Classification and Statistical Aggregation

Each entity classified as a party or political person was assigned to one of five ideological bins: far-left, left, center, right, and far-right. For each bin, the following were calculated:

- Number of mentions.
- Mean sentiment (S_i) in the range $[-1, +1]$.
- Standard deviation of sentiment.

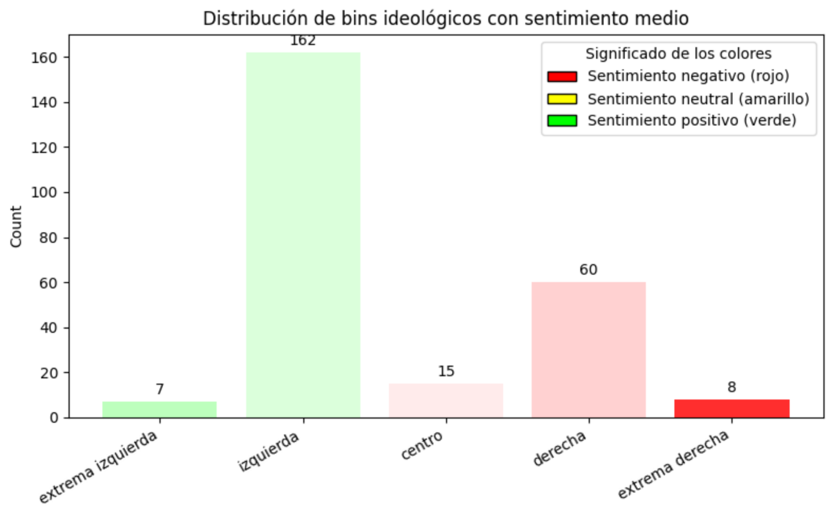

FIGURE 4.1: Distribution of mentions and mean sentiment per ideological bin

To simplify the analysis, the left and far-left bins were merged into a single bin, the two right bins were also merged, and the center bin was removed to avoid bias, obtaining two final bins: *left* and *right*.

4.5 Calculation of Diversity Metrics

To evaluate the diversity of the news set, three complementary metrics were calculated:

- **Normalized Shannon Entropy** (H), which measures coverage diversity across ideological bins: $H = -\sum_i p_i \log p_i / \log 2$, where p_i is the proportion of mentions of bin i. The value is normalized so that 0 represents minimum entropy and 1 maximum.
- **Weighted Mean Sentiment** (S), which reflects the overall sentiment of both bins together, to later penalize deviations from balance (0.0):

$$S = \frac{\sum_i (\text{count}_i \cdot \text{mean\_sentiment}_i)}{\text{total\_count}}$$

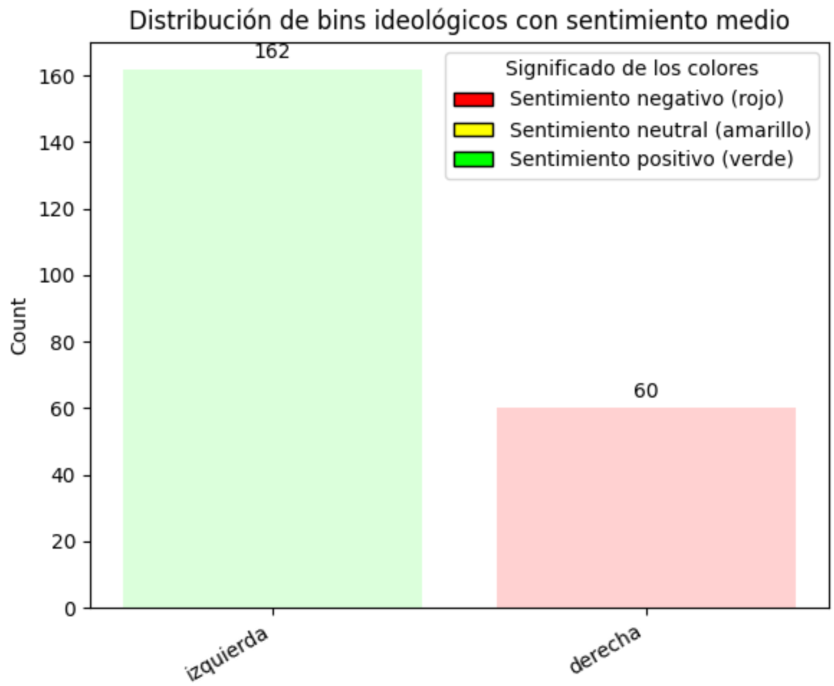

FIGURE 4.2: Distribution of mentions and mean sentiment after merging ideological bins

- **Normalized Polarization** (P), which measures the sentiment difference between the two bins, independent of the number of mentions:

$$P = \frac{|S_{\text{left}} - S_{\text{right}}|}{2}$$

Finally, the three metrics are combined into a global diversity metric D:

$$D = H \cdot (1 - P) \cdot (1 - |S|)$$

where H corresponds to producer diversity, and S and P are variants of the consumer activation metric, capturing respectively the mean bias and polarization. This combination allows identifying hidden manipulations: for example, a positive mean sentiment in both bins could indicate bias, even if apparent polarization is low, and vice versa, if the mean sentiment is zero but both bins have S_i of opposite signs, generating sympathy toward one ideology while creating antipathy toward the other.

4.6 Example Results

For the selected sample of 500 news articles from August 2024, the following results were obtained per ideological bin:

```
Ideological bin: far-left
  Number of mentions : 7
  Mean sentiment     : 0.13
  Standard deviation : 0.53
----------------------------------------
Ideological bin: left
```

```
  Number of mentions : 162
  Mean sentiment     : 0.07
  Standard deviation : 0.53
----------------------------------------
Ideological bin: center
  Number of mentions : 15
  Mean sentiment     : -0.04
  Standard deviation : 0.59
----------------------------------------
Ideological bin: right
  Number of mentions : 60
  Mean sentiment     : -0.09
  Standard deviation : 0.46
----------------------------------------
Ideological bin: far-right
  Number of mentions : 8
  Mean sentiment     : -0.41
  Standard deviation : 0.46
```

After merging the respective bins into *left* and *right* and removing the *center* bin, we obtain:

```
'left': {
    'count': 169,
    'mean_sentiment': 0.07
},
'right': {
    'count': 68,
    'mean_sentiment': -0.13
}
```

Applying the formulas for entropy, mean sentiment, polarization, and diversity gives:

$$H = 0.8647, \quad S = 0.0151, \quad P = 0.1001, \quad D = 0.7665$$

These results indicate that, although the coverage between left and right is relatively balanced (H high), there is slight sentiment polarization (S and P), reflecting the usefulness of combining multiple metrics to accurately assess informational diversity.

4.7 Conclusions

The analysis carried out on the 3Cat dataset demonstrates that it is possible to quantify the political diversity of a news set through an automated NLP pipeline, entity classification, and combination of diversity and activation metrics. The proposed methodology allows not only measuring producer ideological coverage but also identifying biases and manipulations perceived by consumers, providing a robust tool for comparative studies and temporal monitoring of informational diversity, obtaining a diversity metric with a high degree of objectivity by combining both producer and consumer perspectives.

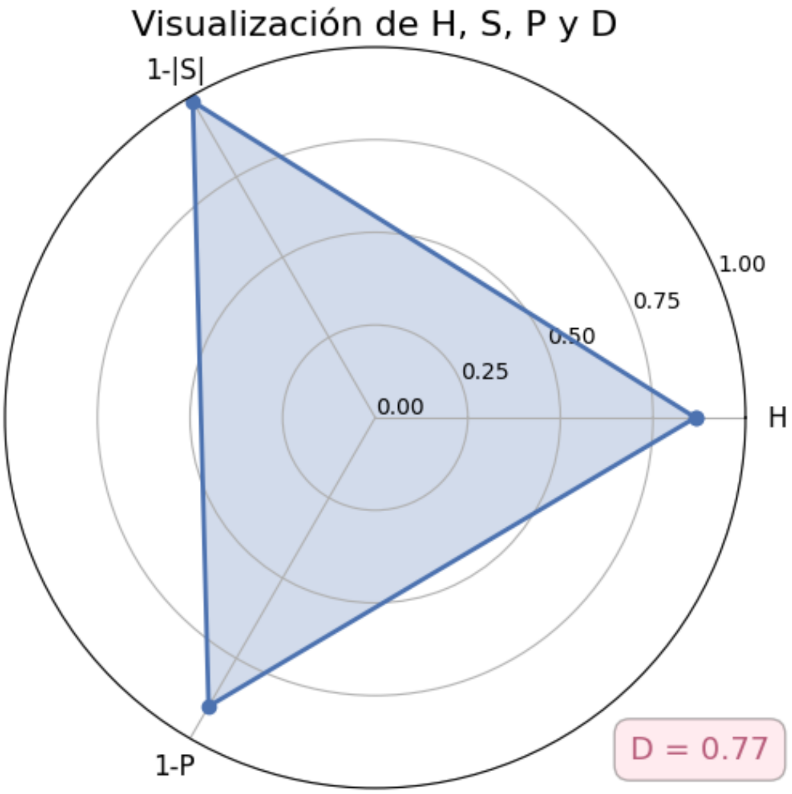

FIGURE 4.3: Combined diversity metric D and components of sentiment and polarization

Appendix A

Technical Requirements

A.1 *1_train_export*

To ensure the correct reproduction of the experiments and avoid incompatibilities between the different libraries used, it was necessary to build a specific execution environment to guarantee the proper functioning of the *Recommenders* package (Recommenders Team, 2020) version 1.2.1, thus avoiding incompatibilities and errors derived from unsupported dependencies. In particular, the recommendation models used (*LSTUR*, *NRMS*, *NAML*, and *NPA*) depend on Microsoft's `recommenders` library, which is strongly tied to specific versions of *TensorFlow*, *CUDA*, and *cuDNN*. Any deviation from these versions causes compilation errors, model loading failures, or incompatibilities with *NumPy*.

A.1.1 Main Compatibility Requirements

- **Python $<$ 3.10**. TensorFlow 2.10, the last version compatible with GPU via *pip* installation, only works correctly with Python 3.9 or lower.
- **TensorFlow 2.10.0**. This is the last version that includes official GPU support with precompiled binaries. Higher versions require custom builds.
- **CUDA 11.2 and cuDNN 8.2.0**. These are the exact versions that TensorFlow 2.10 expects. Using different versions (e.g., CUDA 12 or cuDNN 9) causes failures when initializing the GPU.
- **NumPy $<$ 2.0**. Recent NumPy versions break compatibility with TensorFlow 2.x, so version 1.26.4 is fixed.
- **Binary installation of `blis`, `thinc`, and `spacy`**. These libraries may fail to compile from source on Windows systems, so they are installed exclusively in binary format.

A.1.2 Environment Installation Procedure

The complete process used to create the functional `reco_env` environment is detailed below:

```
--- 1. Creation and Activation of the Environment
>>> conda create -n reco_env python=3.9 -y
>>> conda activate reco_env

--- 2. Installation of C/C++ Packages
>>> pip install blis==0.7.9 --only-binary :all:
```

```
>>> pip install thinc==8.1.10 --only-binary :all:
>>> pip install spacy==3.5.4 --only-binary :all:

--- 3. Installation of 'recommenders' with GPU support
>>> pip install recommenders[gpu]

--- 4. Compatible NumPy Installation (version < 2.0)
>>> pip install numpy==1.26.4

--- 5. Register Environment in Jupyter Notebook
>>> conda install ipykernel -y
>>> python -m ipykernel install --user --name=reco_env
--display-name="Python3.9 (reco_env)"

--- 6. Installation of TensorFlow 2.10 and exact CUDA/cuDNN dependencies
>>> pip install tensorflow==2.10.0
>>> conda install -c conda-forge -c nvidia cudatoolkit=11.2
cudnn=8.2.0 -y
```

This set of versions ensures full compatibility between TensorFlow, CUDA/cuDNN, NumPy, and the `recommenders` library. In this way, the models can be trained and evaluated without errors and with GPU acceleration.

It will also be necessary to download the files 'glove.6B.300d.txt' (Pennington, Socher, and Manning, 2014) and 'MIND_small' (MSNews, 2020).

A.2 *2_metrics_RADio*

To correctly run the metrics repository, it is necessary to install the following packages using the commands:

```
>>> pip install bs4 community elasticsearch gensim lxml nano
python-louvain stop_words textblob textstat minimock textblob_nl
pathlib
>>> pip install https://github.com/explosion/spacy-models/releases
/download/en_core_web_sm-3.0.0
/en_core_web_sm-3.0.0.tar.gz#egg=en_core_web_sm
```

A.3 *3_3Cat*

To correctly run the *3_3Cat* section, it is necessary to have an environment with support for local language models via `Ollama`, as well as the corresponding orchestration libraries. The installation of requirements is performed using the following commands:

```
>>> pip install langchain langchain_community sparqlwrapper
```

Additionally, it is necessary to install and configure `Ollama` on the system and download, for mid-range computers, the model `Qwen3:8b`. Once Ollama is installed, the model can be obtained by selecting it in the dropdown menu and typing any message in the console, which will trigger automatic download.

After these steps, the environment is ready for executing the *3_3Cat* components based on language models and workflows defined via `LangChain`.

Finally, it will be necessary to download the 3Cat news database file (3Cat, 2024), whose news articles span a temporal range from 01/01/2021 to 30/11/2024.

Appendix B

Model Metrics Comparison

This appendix includes the full tables of diversity metrics for all models analyzed in this work. These tables complement the main text by providing detailed values for comparison with RADio.

Metric	RADio	This work
Calibration Topic	0.665	0.656
Calibration Complexity	0.523	0.497
Fragmentation	0.746	0.933
Activation	0.374	0.298
Representation	0.290	0.048
Alternative Voices	0.096	0.001

TABLE B.1: Metric comparison for the LSTUR model.

Metric	RADio	This work
Calibration Topic	0.642	0.642
Calibration Complexity	0.497	0.497
Fragmentation	0.746	0.933
Activation	0.374	0.299
Representation	0.290	0.034
Alternative Voices	0.096	0.001

TABLE B.2: Metric comparison for the NAML model.

Metric	RADio	This work
Calibration Topic	0.699	0.699
Calibration Complexity	0.494	0.494
Fragmentation	0.746	0.929
Activation	0.374	0.295
Representation	0.290	0.048
Alternative Voices	0.096	0.002

TABLE B.3: Metric comparison for the NPA model.

Metric	**RADio**	**This work**
Calibration Topic	0.647	0.647
Calibration Complexity	0.493	0.493
Fragmentation	0.746	0.933
Activation	0.374	0.291
Representation	0.290	0.037
Alternative Voices	0.096	0.001

TABLE B.4: Metric comparison for the NRMS model.

Metric	**RADio**	**This work**
Calibration Topic	0.774	0.701
Calibration Complexity	0.540	0.498
Fragmentation	0.752	0.945
Activation	0.386	0.300
Representation	0.293	0.058
Alternative Voices	0.083	0.001

TABLE B.5: Metric comparison for the Most Popular model.

Metric	**RADio**	**This work**
Calibration Topic	0.736	0.701
Calibration Complexity	0.552	0.497
Fragmentation	0.810	0.946
Activation	0.421	0.300
Representation	0.339	0.060
Alternative Voices	0.102	0.001

TABLE B.6: Metric comparison for the Random model.

Bibliography

3Cat (2024). *3Cat news database*. Dataset. URL: https://drive.google.com/file/d/1o3pmii7_P0vx-PBHQ0pUHnC7samyXKtG/view.

Blanco Borrás, Rubén (2026). *Analysis of Content Diversity in News Recommendation Systems*. Software repository. URL: https://github.com/Rbb93/TFM.

Halwesit (2020). *News-Recommender-MIND – Base YAML files*. Experimental configuration. URL: https://github.com/halwesit/News-Recommender-MIND/blob/main/results/utils/lstur.yaml.

MSNews (2020). *MIND Small & Large Datasets – Download*. Dataset. URL: https://msnews.github.io/.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014). *GloVe 6B 300d*. Pre-trained word embeddings file. URL: https://www.kaggle.com/datasets/thanakomsn/glove6b300dtxt.

Recommenders Team (2020). *Recommenders*. Software repository. URL: https://github.com/recommenders-team/recommenders.

TechPolicy Press (2023). *What is Media Diversity and Do Recommender Systems Have It?* Popular science article. URL: https://www.techpolicy.press/what-is-media-diversity-and-do-recommender-systems-have-it/.

Vrijenhoek, Sanne (2020). *Metrics calculation (RADio)*. Metrics calculation notebook. URL: https://github.com/svrijenhoek/RADio/blob/main/metrics_calculation.ipynb.

Vrijenhoek, Sanne et al. (2024). *RADio* – An Introduction to Measuring Normative Diversity in News Recommendations*. Popular science article. URL: https://dl.acm.org/doi/10.1145/3636465.