Predicting Encyclopedia Hyperlinks: A Story of Directed Hypergraphs and Machine Learning

Moses Boudourides
Graduate Program on Data Science
School of Professional Studies
Northwestern University

October 2025

Abstract

This document presents a comprehensive narrative of research on predicting hyperlinks in encyclopedia networks using directed hypergraphs and graph neural networks. We explore how the CHESHIRE algorithm, using only simple degree features, achieves strong performance in predicting editorial cross-reference decisions across six encyclopedias covering topics from hate groups to social movements. Through detailed analysis of network structure, degree distributions, and topological properties, we uncover surprising insights about what makes hyperlink prediction successful.

1 Introduction: The Puzzle of Encyclopedia Cross-References

1.1 Why Do Editors Link Entries?

Imagine you are reading an encyclopedia entry about the Civil Rights Movement. At the end of the entry, you find a "See Also" section that directs you to related entries: Martin Luther King Jr., Rosa Parks, the Montgomery Bus Boycott, and the Voting Rights Act. These cross-references are not random—they represent deliberate decisions by encyclopedia editors and entry authors about which topics are conceptually related, historically connected, or thematically coherent.

But what guides these decisions? Is there a pattern to how encyclopedia editors choose to link entries? More intriguingly, could we *predict* these editorial choices using computational methods?

This is the central question that motivates our research. We investigate whether machine learning algorithms can learn the implicit rules that govern cross-referencing in encyclopedias. If successful, such algorithms could help editors identify missing links, suggest new connections, or even reveal hidden patterns in how knowledge is organized.

1.2 The Challenge: From Text to Networks

Encyclopedias are naturally structured as networks. Each entry is a node, and each cross-reference is a directed edge pointing from one entry to another. However, traditional network analysis tools are designed for *simple graphs*, where each edge connects exactly two nodes. Encyclopedia cross-references are more complex: a single entry might reference multiple other entries simultaneously, creating a *many-to-many* relationship.

To capture this complexity, we model encyclopedias as **directed hypergraphs**—mathematical structures where edges (called hyperedges) can connect multiple nodes and have direction. Each directed hyperedge consists of a **tail** (source nodes) and a **head** (target nodes), representing the many-to-many nature of cross-references. This representation preserves the rich structure of encyclopedia cross-references while enabling sophisticated computational analysis.

1.3 Our Approach: The CHESHIRE Algorithm

We employ a graph neural network called CHESHIRE (CHEbyshev Spectral HyperlInk pREdictor) to predict missing or future hyperlinks. The algorithm learns patterns from existing cross-references and uses them to predict which entries should be linked but currently are not.

Remarkably, we find that using only the *simplest possible features*—how many incoming and outgoing links each entry has—the algorithm achieves strong predictive performance. This suggests that the structure of the network itself, rather than the semantic content of entries, contains powerful signals about editorial linking decisions.

1.4 Six Encyclopedias, Six Experiments

We test our approach on six encyclopedias covering diverse topics:

- **Balleck**: Hate groups and extremist organizations in America
- **Knight**: Conspiracy theories in American history
- **Ness**: American social movements
- **Powers**: Protest, power, and change
- **Snow**: Social and political movements
- **Thompson**: Diversity and social justice

These encyclopedias vary in size (185 to 434 entries), connectivity (830 to 2860 links), and editorial philosophy. By analyzing all six, we can identify universal patterns that transcend specific topics or editorial styles.

2 Background: Understanding Graphs, Hypergraphs, and Neural Networks

Before diving into our analysis, we need to establish foundational concepts that non-experts may not be familiar with. This section provides accessible explanations of the mathematical and computational tools we use.

2.1 What Is a Graph?

A **graph** is a mathematical structure consisting of:

- **Nodes** (also called vertices): The entities being studied (in our case, encyclopedia entries)
- **Edges** (also called links): Connections between nodes (in our case, cross-references)

Graphs can be **undirected** (edges have no direction, like friendships on social media) or **directed** (edges point from one node to another, like citations in academic papers or cross-references in encyclopedias).

Example: If entry A says "See Also: B, C," we draw directed edges from A to B and from A to C.

2.2 What Is a Hypergraph?

A **hypergraph** generalizes the concept of a graph by allowing edges to connect *more than two nodes simultaneously.* In a hypergraph:

- Each **hyperedge** can connect any number of nodes
- Hyperedges can be **directed**, with a tail (source nodes) and a head (target nodes)

Why do we need hypergraphs? Consider an encyclopedia entry x that is referenced by entries a, b, and c, and in turn references entries d, e, and f. In a traditional graph, we would represent this as six separate edges. In a directed hypergraph, we can represent this as a single hyperedge:

$$e(x) = (\{a, b, c\}, \{d, e, f\})$$

where $\{a, b, c\}$ is the tail (predecessors) and $\{d, e, f\}$ is the head (successors). This representation captures the fact that all these references are associated with entry x, preserving important structural information.

2.3 What Is a Graph Neural Network?

A **graph neural network** (GNN) is a type of machine learning model designed to process graph-structured data. Unlike traditional neural networks that work with fixed-size inputs (like images or text), GNNs can handle networks of varying sizes and structures.

The key idea is **message passing**: each node aggregates information from its neighbors, updates its own representation, and passes messages to other nodes. Through multiple rounds of message passing, nodes learn representations that capture both their local neighborhood structure and global network patterns.

Analogy: Imagine a social network where each person (node) learns about their community by talking to friends (neighbors). After several rounds of conversation, each person has a rich understanding not just of their immediate friends, but of the broader social structure.

2.4 The CHESHIRE Algorithm: A Closer Look

CHESHIRE is specifically designed for directed hypergraphs. It works in several stages:

1. **Encoding**: Convert the hypergraph structure into numerical representations (embeddings) that capture each node's position in the network.

2. **Expansion**: Transform hyperedges into ordinary graph edges by creating cliques—fully connected subgraphs where every node in a hyperedge is connected to every other node.

3. **Convolution**: Apply Chebyshev spectral filters to propagate information across the network. This is the "learning" step where the algorithm discovers patterns.

4. **Pooling**: Aggregate node-level information into hyperedge-level predictions.

5. **Classification**: Predict whether a potential hyperedge is real (exists in the encyclopedia) or fake (generated for comparison).

The algorithm is trained using **cross-validation**: we hide some real hyperedges, generate fake ones, and see if the algorithm can distinguish between them. If it can, we know it has learned meaningful patterns about how entries are cross-referenced.

3 Data: Six Encyclopedias as Directed Hypergraphs

3.1 Encyclopedia Statistics

Our six encyclopedias vary considerably in size and structure:

Encyclopedia	Entries	Cross-References
Balleck	185	845
Knight	289	1071
Ness	314	830
Powers	314	830
Snow	434	2860
Thompson	314	830

Snow is the largest and most densely connected encyclopedia, with 434 entries and 2860 cross-references. This means each entry, on average, references about 6.6 other entries—a remarkably high level of interconnection.

Ness, Powers, and Thompson are identical in size (314 entries, 830 links), which is not coincidental: they share a common editorial approach and were constructed using similar methodologies. However, as we will see, their network structures differ in subtle but important ways.

Balleck and Knight occupy a middle ground, with moderate sizes and connectivity levels.

3.2 From "See Also" to Hyperedges

For each entry x, we construct a directed hyperedge $e(x) = (N^-(x), N^+(x))$ where:

- $N^-(x) =$ **predecessors**: entries that reference x (incoming links)
- $N^+(x) =$ **successors**: entries that x references (outgoing links)

Example: Suppose the entry "Civil Rights Movement" is referenced by "Martin Luther King Jr." and "Rosa Parks," and it references "Voting Rights Act" and "Montgomery Bus Boycott." The hyperedge would be:

$$e(\text{Civil Rights Movement}) = (\{\text{MLK}, \text{Rosa Parks}\}, \{\text{Voting Rights Act}, \text{Montgomery Bus Boycott}\})$$

This representation captures the *ego-network* of each entry—the immediate neighborhood of incoming and outgoing connections.

3.3 Hypergraph Properties

After converting each encyclopedia to a directed hypergraph, we observe:

- **Number of hyperedges** equals the number of entries (each entry has one hyperedge)
- **Tail entries** (total incoming connections across all hyperedges) range from 477 to 2860
- **Head entries** (total outgoing connections) are identical to the number of cross-references

The hypergraph representation is *lossless*—we can perfectly reconstruct the original encyclopedia network from the hypergraph, and vice versa. However, the hypergraph view makes certain patterns more visible and enables specialized algorithms like CHESHIRE.

4 Methodology: Training and Evaluating CHESHIRE

4.1 The Prediction Task

Our goal is to predict whether a potential hyperedge is *real* (exists in the encyclopedia) or *fake* (does not exist). This is a **binary classification** problem.

Why is this useful? If the algorithm can accurately distinguish real from fake hyperedges, it has learned the underlying patterns of cross-referencing. We can then use it to:

- Identify missing links that editors should consider adding
- Detect anomalous links that might be errors
- Understand what structural features make entries likely to be cross-referenced

4.2 Features: In-Degree and Out-Degree

The algorithm needs numerical **features** to make predictions. In machine learning, features are measurable properties or characteristics of the data that the algorithm uses as input. Think of features as the "clues" the algorithm examines to make decisions—like how a doctor uses symptoms (features) to diagnose illness. For our task, we use the simplest possible features:

- **In-degree**: How many entries reference this entry (number of incoming links)
- **Out-degree**: How many entries this entry references (number of outgoing links)

Why start simple? We want to test whether basic structural information alone is sufficient for prediction. If it works, we know that network topology—not semantic content—drives cross-referencing patterns. This is a strong theoretical claim: it suggests that encyclopedia structure follows mathematical regularities independent of topic.

4.3 Training Procedure: 5-Fold Cross-Validation

To rigorously evaluate the algorithm, we use **5-fold cross-validation**:

1. Divide all hyperedges into 5 equal parts (folds)
2. For each fold:
    - Train on 4 folds (80% of data)
    - Test on the held-out fold (20% of data)
    - Generate negative samples (fake hyperedges) by randomly permuting real ones
3. Aggregate results across all 5 folds

This ensures that every hyperedge is tested exactly once, and the algorithm never sees test data during training. The results are therefore unbiased estimates of real-world performance.

4.4 Hyperparameters: Consistent Across All Encyclopedias

To ensure fair comparison, we use identical hyperparameters for all six encyclopedias:

- **Embedding dimension**: 16 (how many numbers represent each node)
- **Convolution dimension**: 16 (size of hidden layers)
- **Chebyshev filter hops**: 2 (how far information propagates)
- **Dropout**: 0.2 (regularization to prevent overfitting)
- **Epochs**: 50 (number of training iterations)
- **Learning rate**: 0.001 (step size for optimization)

These choices are standard for graph neural networks and were not tuned specifically for our task.

4.5 Evaluation Metrics

We measure performance using eight metrics:

1. **F1 Score**: Harmonic mean of precision and recall (balances false positives and false negatives)
2. **Precision**: Of predicted real hyperedges, what fraction are actually real?
3. **Recall**: Of actually real hyperedges, what fraction did we predict?
4. **Accuracy**: Overall fraction of correct predictions
5. **ROC-AUC**: Area under the receiver operating characteristic curve (measures discriminative ability)
6. **PR-AUC**: Area under the precision-recall curve (better for imbalanced data)
7. **MCC**: Matthews correlation coefficient (robust to class imbalance)
8. **Log-Loss**: Logarithmic loss (penalizes confident wrong predictions)

Each metric captures a different aspect of performance. By reporting all eight, we provide a comprehensive picture of how well the algorithm works.

5 Results: Performance Across Six Encyclopedias

5.1 Overall Performance

The CHESHIRE algorithm achieves strong performance across all six encyclopedias, despite using only in-degree and out-degree features:

Encyclopedia	F1	Precision	Recall	ROC-AUC	PR-AUC	MCC
Balleck	0.763	0.789	0.747	0.826	0.817	0.548
Knight	0.648	0.639	0.665	0.699	0.699	0.290
Ness	0.574	0.575	0.601	0.612	0.614	0.155
Powers	0.554	0.561	0.581	0.588	0.586	0.127
Snow	0.747	0.722	0.779	0.819	0.806	0.480
Thompson	0.576	0.557	0.610	0.595	0.576	0.131
Mean	**0.644**	**0.641**	**0.664**	**0.690**	**0.683**	**0.289**

5.2 Interpreting the Numbers

What does an F1 score of 0.644 mean? It means that, on average, the algorithm correctly identifies about 64% of real hyperedges while maintaining a similar level of precision. This is substantially better than random guessing (which would achieve F1 ≈ 0.5) and demonstrates that the algorithm has learned meaningful patterns.

Why does performance vary across encyclopedias? Snow and Balleck achieve the highest scores (F1 > 0.74), while Ness, Powers, and Thompson achieve moderate scores (F1 $\approx$ 0.55-0.58). This variation is not due to differences in the algorithm or hyperparameters—those are identical. Instead, it reflects intrinsic differences in how these encyclopedias are structured.

5.3 The Best Performers: Snow and Balleck

Snow achieves F1 = 0.747 and ROC-AUC = 0.819. This encyclopedia has the highest connectivity (average degree = 6.59), meaning entries are densely cross-referenced. Dense networks provide more training signal: the algorithm has more examples to learn from, and structural patterns are more pronounced.

Balleck achieves F1 = 0.763 and ROC-AUC = 0.826, the highest scores overall. Interestingly, Balleck is not the largest encyclopedia, but it has high *degree heterogeneity*—some entries are heavily referenced (hubs) while others have few connections (peripheral nodes). This heterogeneity creates clear structural patterns that the algorithm can exploit.

5.4 The Moderate Performers: Ness, Powers, Thompson

These three encyclopedias are identical in size (314 entries, 830 links) and achieve similar performance (F1 $\approx$ 0.55-0.58). Their lower scores reflect:

- **Lower connectivity**: Average degree = 2.64, less than half of Snow's
- **More uniform degree distributions**: Fewer hubs and peripheral nodes
- **Sparser networks**: Less training signal for the algorithm

However, even these "moderate" scores are meaningful. An F1 of 0.57 means the algorithm is correctly identifying more than half of real hyperedges, far better than random chance.

5.5 The Intermediate Performer: Knight

Knight (F1 = 0.648) falls between the high and moderate performers. It has intermediate size (289 entries) and connectivity (average degree = 3.71). Its performance suggests that network size and density both matter, but neither alone determines predictability.

6 Degree Distributions: The Shape of Encyclopedia Networks

6.1 What Is a Degree Distribution?

The **degree distribution** of a network describes how many nodes have each possible degree. For example:

- How many entries have exactly 1 incoming link? 2? 10?
- How many entries reference exactly 1 other entry? 5? 20?

Degree distributions reveal fundamental properties of networks:

- **Uniform distributions**: Most nodes have similar degrees (egalitarian structure)
- **Skewed distributions**: A few hubs have many connections, most nodes have few (hierarchical structure)
- **Power-law distributions**: Degree follows a power law $P(k) \propto k^{-\gamma}$ (scale-free structure)

6.2 Degree Statistics Across Encyclopedias

We compute comprehensive statistics for in-degree, out-degree, and total-degree:

Mean degree:

- Snow: 6.59 (highest)
- Balleck: 4.57
- Knight: 3.71
- Ness, Powers, Thompson: 2.64 (lowest)

Standard deviation (variability):

- Balleck: 13.87 (highest—extremely heterogeneous)
- Snow: 9.27
- Knight: 3.40
- Ness, Powers, Thompson: ≈ 2.5

Gini coefficient (inequality):

- Balleck: 0.80 (highest inequality)
- Ness, Powers, Thompson: ≈ 0.56
- Snow: 0.52 (most equal among large encyclopedias)

6.3 Interpreting Degree Heterogeneity

Balleck's extreme heterogeneity (std = 13.87, Gini = 0.80) means a few entries are massively over-referenced while most have few connections. The maximum in-degree is 125—one entry is referenced by 125 others! This creates a stark hub-and-spoke structure.

Snow's balanced structure (Gini = 0.52) means connections are more evenly distributed. Even though Snow has high average degree, it avoids extreme concentration. This suggests a more democratic editorial philosophy where many entries are considered important.

Ness, Powers, Thompson's uniformity (low std, moderate Gini) reflects sparse, relatively egalitarian networks. No entry dominates; cross-references are spread more evenly.

6.4 Why Does Degree Distribution Matter for Prediction?

Networks with clear hubs are easier to predict because:

- Hubs create strong structural signals ("this entry is important")
- Degree becomes a powerful predictor ("high-degree entries link to other high-degree entries")
- The algorithm can learn simple rules ("if degree > threshold, predict link")

Conversely, uniform networks are harder to predict because:

- All entries look similar structurally
- Degree provides less discriminative information
- The algorithm must learn subtle patterns

This explains why Balleck (high heterogeneity) and Snow (high density) are easier to predict than Ness/Powers/Thompson (low heterogeneity, low density).

7 Degree-Performance Correlations: What Predicts Success?

7.1 Testing the Hypothesis

We hypothesize that degree distribution characteristics predict CHESHIRE performance. To test this, we compute correlations between degree statistics (mean, median, standard deviation, max, Gini) and performance metrics (F1, precision, recall, etc.).

7.2 Strongest Positive Correlations

Total-Degree Max ↔ Precision: $r = +0.968$ ($p = 0.002$)

This is an extremely strong correlation. Encyclopedias where the most-connected entry has many connections achieve higher precision. Why? High-degree hubs are easy to identify, and the algorithm rarely makes false positive predictions about them.

In-Degree Mean ↔ F1: $r = +0.886$ ($p = 0.019$)

Higher average in-degree predicts better overall performance. Dense networks provide more training examples and clearer structural patterns.

Out-Degree Std ↔ Precision: $r = +0.960$ ($p = 0.002$)

Greater variability in out-degree (how many entries each entry references) predicts higher precision. Heterogeneous networks are more predictable.

7.3 Interpreting Positive Correlations

These correlations confirm our intuition: **denser, more heterogeneous networks are easier to predict**. The algorithm exploits degree information to make accurate predictions. Entries with many connections are structurally distinctive and therefore predictable.

Analogy: Imagine trying to predict friendships in two communities. In one community, everyone has about 5 friends (uniform). In another, some people have 50 friends while others have 2 (heterogeneous). The heterogeneous community is easier to predict because popular people (hubs) are obvious targets for friendship.

7.4 The Absence of Strong Negative Correlations

Interestingly, we find *no* degree statistics that strongly *hurt* performance. Even low-degree, uniform networks achieve moderate success (F1 ≈ 0.55). This suggests that degree information is universally useful, though its predictive power varies.

8 Statistical Comparisons: Are Encyclopedias Really Different?

8.1 The Question of Significance

We observe that Snow has higher average degree than Ness. But is this difference *statistically significant*, or could it arise by chance? To answer this, we perform rigorous statistical tests.

8.2 Kolmogorov-Smirnov Tests

The **Kolmogorov-Smirnov (KS) test** compares two distributions and asks: "Are these drawn from the same underlying distribution?" We apply KS tests to all 15 pairwise comparisons (6 encyclopedias, choose 2).

Result: For in-degree, out-degree, and total-degree, we find:

- **Kruskal-Wallis H-test** (overall difference): $H > 186$, $p < 0.001$ for all degree types

- This means encyclopedias have *highly significantly different* degree distributions

8.3 Effect Sizes: How Big Are the Differences?

Statistical significance tells us differences are real, but not how *large* they are. For this, we compute **Cliff's Delta**, an effect size measure:

- $|\delta| < 0.147$: Negligible
- $0.147 \leq |\delta| < 0.33$: Small
- $0.33 \leq |\delta| < 0.474$: Medium
- $|\delta| \geq 0.474$: Large

Largest effects:

- Snow vs. Ness: $\delta \approx 0.76$ (large)
- Snow vs. Powers: $\delta \approx 0.74$ (large)
- Snow vs. Thompson: $\delta \approx 0.72$ (large)

These large effect sizes confirm that Snow's degree distribution is *fundamentally different* from Ness/Powers/Thompson, not just slightly higher.

8.4 Implications for Generalization

The significant differences across encyclopedias mean we cannot assume results generalize automatically. An algorithm that works well on Snow might struggle on Ness. This motivates our choice to test on six diverse encyclopedias rather than one.

However, the fact that CHESHIRE achieves positive results on *all* six encyclopedias—despite their differences—suggests the underlying principles (degree-based prediction) are robust.

9 Degree-Stratified Performance: The Hub Advantage

9.1 The Hypothesis

We hypothesize that high-degree nodes (hubs) are easier to predict than low-degree nodes (peripheral entries). Intuitively, hubs have more connections, providing more structural information for the algorithm to exploit.

9.2 Stratification Procedure

We divide nodes into three categories based on degree quartiles:

- **Low-degree**: Bottom 33% (few connections)
- **Medium-degree**: Middle 33% (moderate connections)
- **High-degree**: Top 33% (many connections)

We then measure CHESHIRE performance separately for each category.

9.3 Results: A Clear Gradient

Average F1 scores across all encyclopedias:

- Low-degree nodes: F1 = 0.544
- Medium-degree nodes: F1 = 0.644 (+0.10)
- High-degree nodes: F1 = 0.744 (+0.20 from low)

This is a **37% improvement** from low to high degree ($\frac{0.744-0.544}{0.544} \approx 0.37$).

9.4 Interpreting the Hub Advantage

Why are hubs easier to predict?

1. **More training examples**: Hubs appear in many hyperedges, so the algorithm sees them frequently during training.
2. **Stronger structural signals**: High degree is itself a signal ("this entry is important"), making hubs stand out.
3. **Redundant information**: Hubs have many connections, so even if the algorithm misses some, it can infer others from the overall pattern.
4. **Centrality**: Hubs are often topically central, connecting multiple themes. This makes their links more predictable from network structure alone.

Why are peripheral nodes harder to predict?

1. **Sparse connections**: Few links mean less structural information.
2. **Idiosyncratic links**: Peripheral nodes' connections may reflect specific editorial choices rather than general patterns.
3. **Rare occurrences**: The algorithm sees peripheral nodes infrequently, limiting learning.

9.5 Implications for Encyclopedia Editing

The hub advantage suggests that automated link prediction is most useful for:

- Identifying missing links *to* hubs (high recall for important entries)
- Suggesting new peripheral entries that should link to existing hubs

Conversely, predicting links *between* peripheral entries remains challenging and may require semantic information (entry content) rather than just structure.

10 Topology Impact: Surprising Findings About Network Structure

10.1 Beyond Degree: Topological Features

Degree is just one aspect of network structure. We also compute:

- **Density**: Fraction of possible edges that exist
- **Assortativity**: Do high-degree nodes connect to other high-degree nodes?
- **Reciprocity**: Fraction of edges that are bidirectional
- **Clustering**: Do neighbors of a node connect to each other?
- **Transitivity**: Global measure of triangle density
- **Connected components**: Number of disconnected subgraphs

10.2 Expected Correlations

We expect:

- **Density → better performance**: More edges = more training signal
- **Assortativity → better performance**: Assortative networks have clear hierarchies
- **Clustering → better performance**: Clustered networks have predictable local structure

10.3 Actual Results: Surprises!

Confirmed expectations:

- Density ↔ Precision: $r = +0.962$ ($p = 0.002$) ✓
- Avg Hyperedge Size ↔ Recall: $r = +0.950$ ($p = 0.004$) ✓

Surprising findings:

- **Out-Degree Assortativity ↔ Recall:** $r = -0.989$ **(**$p = 0.0002$**)**
  This is the *strongest correlation we found*, and it's *negative*! Assortative mixing (hubs connecting to hubs) *hurts* performance.

- **Transitivity ↔ Recall:** $r = -0.949$ **(**$p = 0.004$**)**
  Higher clustering *reduces* performance, contrary to expectations.

10.4 Interpreting the Assortativity Paradox

Why does assortativity hurt performance?

In assortative networks, high-degree nodes preferentially connect to other high-degree nodes. This creates a "rich club" where hubs form a densely connected core. Meanwhile, low-degree nodes connect mostly to each other, forming a sparse periphery.

The problem: This structure makes the network *too predictable* in some ways and *not predictable enough* in others:

- **Within the core**: Links are so dense that predicting any specific link is hard (too many possibilities).

- **Within the periphery**: Links are so sparse that structural patterns are weak.

- **Core-periphery links**: These are rare in assortative networks, but they're the most informative for learning.

Disassortative networks (hubs connecting to low-degree nodes) create more diverse mixing patterns. This diversity provides richer training signal: the algorithm learns that hubs connect to *various* types of nodes, not just other hubs.

10.5 Interpreting the Transitivity Paradox

Why does clustering hurt performance?

High transitivity means many triangles: if A links to B and B links to C, then A likely links to C. This creates tightly-knit communities.

The problem: Within a tight community, *everyone links to everyone*. This makes predicting any specific link difficult—there are too many true positives, and the algorithm struggles to distinguish which links are most important.

Low-transitivity networks have sparser local structure. Links are more selective, creating clearer patterns that the algorithm can learn.

10.6 Practical Implications

These counterintuitive findings suggest that **editorial diversity improves predictability**:

- Encyclopedias where important entries (hubs) reference both other important entries *and* peripheral entries are easier to model.

- Encyclopedias with tight topical clusters are harder to model because links within clusters are too dense.
- Encouraging cross-topic references (reducing clustering) may improve both predictability and knowledge discovery.

11 Synthesis: What Makes Hyperlink Prediction Successful?

11.1 The Four Pillars of Predictability

Our analysis reveals four key factors that determine CHESHIRE performance:

1. **Network Density** ($r = +0.96$ with precision)
   - More connections = more training signal
   - Dense networks have clearer structural patterns
   - Snow (density = 0.015) outperforms Ness (density = 0.008)
2. **Degree Heterogeneity** ($r = +0.97$ for Total-Degree Max)
   - Networks with clear hubs are easier to predict
   - Degree becomes a powerful discriminative feature
   - Balleck (Gini = 0.80) benefits from extreme heterogeneity
3. **Disassortative Mixing** ($r = -0.99$ for Out-Degree Assortativity)
   - Hubs connecting to diverse nodes (not just other hubs) improves prediction
   - Diverse mixing patterns provide richer training signal
   - Balleck (assortativity = -0.20) benefits from disassortativity
4. **Low Clustering** ($r = -0.95$ for Transitivity)
   - Sparse local structure makes links more selective
   - Selective links are easier to predict than dense cliques
   - Networks with clear hierarchies outperform egalitarian ones

11.2 The Paradox of Simplicity

Perhaps the most striking finding is that **simple degree features suffice**. We did not use:

- Semantic information (entry content)
- Topic modeling

- Named entity recognition
- Advanced centrality measures
- Community detection

Yet we achieved F1 scores up to 0.76. This suggests that **network structure alone captures much of what determines cross-referencing**.

Why? Encyclopedia editors, whether consciously or not, follow structural principles:

- Important topics (hubs) are referenced more
- Entries reference topics of similar importance
- Cross-references create navigable paths through knowledge

These principles manifest as degree patterns, which the algorithm exploits.

11.3 When Does Structure Fail?

Despite strong overall performance, CHESHIRE struggles with:

- **Peripheral nodes** (F1 = 0.54): Sparse connections provide weak signals
- **Uniform networks** (Ness/Powers/Thompson): Lack of hubs reduces discriminative power
- **Idiosyncratic links**: Editorial choices based on specific content rather than general structure

These failures suggest that **semantic information would complement structural features**. A hybrid approach combining degree with topic similarity might achieve even higher performance.

12 Broader Implications: Beyond Encyclopedia Prediction

12.1 Theoretical Contributions

Our work contributes to several areas:

1. **Network Science**: We demonstrate that directed hypergraphs are a natural representation for many-to-many citation networks.
2. **Link Prediction**: We show that simple degree features can achieve strong performance, challenging the assumption that complex features are necessary.

3. **Knowledge Organization**: We reveal structural principles underlying encyclopedia cross-referencing, suggesting that knowledge organization follows mathematical regularities.

4. **Graph Neural Networks**: We validate CHESHIRE's effectiveness on real-world hypergraphs, extending GNN applications beyond ordinary graphs.

12.2 Practical Applications

Our methods could be applied to:

- **Encyclopedia editing**: Automated suggestion of missing cross-references
- **Digital libraries**: Linking related documents in large collections
- **Knowledge graphs**: Predicting missing edges in structured knowledge bases
- **Citation networks**: Recommending relevant papers to cite
- **Hypertext systems**: Optimizing link structure for navigation

12.3 Limitations and Future Work

Our study has limitations:

1. **Small sample size**: Six encyclopedias is a modest dataset. Larger studies across dozens of encyclopedias would strengthen conclusions.

2. **Single domain**: All encyclopedias cover social movements and related topics. Testing on encyclopedias in science, history, or art would assess generalization.

3. **Structural features only**: We deliberately excluded semantic features to test structural sufficiency. Hybrid models could improve performance.

4. **Static networks**: Encyclopedias evolve over time. Temporal models could predict how cross-references change as new entries are added.

Future directions:

- **Semantic-structural fusion**: Combine degree features with topic modeling or word embeddings
- **Temporal prediction**: Model how encyclopedia networks grow and evolve
- **Explanatory models**: Develop interpretable models that explain *why* certain links are predicted
- **Cross-domain transfer**: Train on one encyclopedia, test on another to assess transferability

13 Conclusion: The Story in Retrospect

13.1 The Journey

We began with a simple question: Can we predict encyclopedia cross-references using machine learning? Through a systematic investigation spanning six encyclopedias, we discovered that:

1. **Structure matters more than expected**: Simple degree features achieve F1 scores up to 0.76, demonstrating that network topology alone captures much of what determines cross-referencing.

2. **Density and heterogeneity predict success**: Dense networks with clear hubs (Snow, Balleck) are easier to model than sparse, uniform networks (Ness, Powers, Thompson).

3. **Hubs are predictable**: High-degree nodes are 37% easier to predict than low-degree nodes, confirming the hub advantage hypothesis.

4. **Assortativity and clustering hurt**: Contrary to expectations, assortative mixing and high transitivity reduce performance. Diverse mixing patterns and sparse local structure improve predictability.

13.2 The Bigger Picture

Our findings suggest that encyclopedia cross-referencing is not arbitrary. Editors follow implicit structural principles that manifest as degree patterns, assortativity, and clustering. These principles are learnable by algorithms, opening possibilities for automated assistance in knowledge organization.

More broadly, our work demonstrates the power of **network thinking** in understanding information systems. By representing encyclopedias as directed hypergraphs and analyzing their topology, we uncover patterns invisible to traditional text analysis.

13.3 Final Reflection

The success of CHESHIRE with minimal features is both encouraging and humbling. Encouraging because it shows that sophisticated predictions are possible with simple inputs. Humbling because it reminds us that much of what seems like editorial creativity may actually follow mathematical regularities.

As we build increasingly complex AI systems, it's worth remembering that sometimes the simplest features—in our case, just counting incoming and outgoing links—reveal the deepest truths about how knowledge is structured and connected.