Advanced Hypergraph Mining for Web Applications Using Sphere Neural Networks

Zhongtian Sun
zs440@cam.ac.uk
University of Cambridge
Cambridge, UK
University of Oxford
Oxford, UK
University of Kent
Canterbury, UK

Anoushka Harit
University of Cambridge
Cambridge, UK
ah2415@cam.ac.uk

Jongmin Yu
jm.andrew.yu@gmail.com
University of Cambridge
Cambridge, UK

Jingyun Wang
jingyun.wang@durham.ac.uk
Durham University
Durham, UK

Pietro Liò
pl219@cam.ac.uk
University of Cambridge
Cambridge, UK

Abstract

Web-based applications often involve analyzing complex multi-relational data generated by various domains, including social platforms, bibliographic networks, recommendation systems, and e-commerce platforms. Traditional graph-based methods struggle to model interactions beyond simple pairwise relationships, such as higher-order dependencies and the underlying geometric and structural properties of the data. This paper presents a novel application of hyperspherical deep learning to hypergraphs, integrating geometric hypergraph mining with a Sphere Neural Network (SNN) to model and analyze these intricate relationships effectively. Using real-world datasets, including Reddit, DBLP, MovieLens, and Amazon Co-purchase, our framework embeds hypergraphs into hyperspherical spaces, preserving both relational and geometric properties. Experimental results demonstrate that our method significantly improves performance on tasks such as recommendation, co-purchase prediction, and user behavior analysis, outperforming state-of-the-art techniques. This work highlights the potential of integrating geometric hypergraphs and hyperspherical deep learning to advance the analysis of web-based data.

CCS Concepts

• **Computing Methodologies → Graph Neural Networks, Hypergraph, Sphere Neural Network**.

Keywords

Graph Representation Learning, Recommendation System, Hypergraph , Sphere Neural Network

ACM Reference Format:
Zhongtian Sun, Anoushka Harit, Jongmin Yu, Jingyun Wang, and Pietro Liò. 2025. Advanced Hypergraph Mining for Web Applications Using Sphere Neural Networks. In *Companion Proceedings of the ACM Web Conference 2025 (WWW Companion '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3701716.3715577

WWW Companion '25, Sydney, NSW, Australia

ACM ISBN 979-8-4007-1331-6/2025/04
https://doi.org/10.1145/3701716.3715577

1 Introduction

Web-based applications generate vast amounts of complex data, characterized by multi-relational interactions and higher-order structures. Examples include user discussions on Reddit [1] [1], academic collaboration networks on DBLP [17] [2], movie rating patterns on MovieLens[6] [3], and co-purchase behaviors on Amazon [9][4]. Traditional graph-based models [7, 12, 14–16, 18, 20] often fail to capture the nuanced relationships and structural complexity inherent in these datasets, limiting their effectiveness in tasks such as recommendation, behavior prediction, and anomaly detection.

Hypergraphs, which generalize graphs by allowing hyperedges to connect multiple nodes, provide a more expressive framework for modeling such relationships [2, 5, 13]. However, effectively mining insights from hypergraphs remains a challenge due to their high-dimensional nature and lack of appropriate embedding techniques. To address this, we propose a new application of hyperspherical deep learning to hypergraph analysis, integrating geometric hypergraph mining [11] with Sphere Neural Networks (SNNs) [3, 19], a class of neural networks designed for hyperspherical spaces. By embedding hypergraph structures into a hyperspherical space [11], our approach captures both geometric and relational properties, enabling robust analysis of web-based datasets. Our contributions are threefold:

(1) We introduce a geometric hypergraph mining framework tailored to web-based data, leveraging the expressive power of hyperspherical embeddings [10].
(2) We develop a novel integration of Sphere Neural Networks [3] with hypergraph learning, enhancing higher-order relationship modeling and extending SNNs beyond traditional graphs to more complex structures.

[1] https://zenodo.org/records/3608135
[2] https://www.kaggle.com/datasets/dheerajmpai/dblp2023
[3] https://www.kaggle.com/datasets/grouplens/movielens-20m-dataset
[4] https://snap.stanford.edu/data/amazon-meta.html

(3) We validate our approach on diverse datasets like Reddit [1], DBLP [17], MovieLens [6], and Amazon Co-purchase [9], demonstrating its superiority in key tasks such as recommendation and link prediction.

2 Related Work

The analysis of web-based data has seen extensive exploration through graph-based models. Traditional approaches, such as collaborative filtering and matrix factorization [8], focus on pairwise relationships but fail to capture higher-order interactions. Hypergraphs, which extend graphs by allowing edges to connect multiple nodes, offer a richer representation of complex data. Hypergraph-based models have been applied to tasks like recommendation [4] and community detection [23], yet they often struggle with scalability and the efficient encoding of geometric relationships. Recent advances in geometric deep learning have introduced neural networks for non-Euclidean spaces, such as Sphere Neural Networks (SNNs) [3, 19], which excel at modeling data in hyperspherical spaces. However, these methods have primarily been applied to tasks like image classification and molecular structure prediction, leaving their potential for web-based hypergraph mining under-explored.Our work bridges these gaps by combining hypergraph mining with SNNs[3] to enable efficient modeling of multirelational geometric data in web-based applications. Unlike prior methods, our framework integrates hyperspherical embeddings to preserve both geometric and relational properties, offering enhanced performance on datasets such as Reddit[1], DBLP [17], MovieLens[6], and Amazon Co-purchase[9].

3 Method

Our approach combines geometric hypergraph mining [11] with Sphere Neural Networks (SNNs) [3] to model and analyze web-based data effectively.

3.1 Hypergraph Representation

Let the web-based data be represented as a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where:

(1) $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is the set of N nodes.
(2) $\mathcal{E} = \{e_1, e_2, \ldots, e_M\}$ is the set of M hyperedges.
(3) $\mathbf{X} \in \mathbb{R}^{N \times F}$ is the node feature matrix, where F is the dimensionality of the input features.

Each hyperedge e_k connects a subset of nodes $\mathcal{V}_k \subseteq \mathcal{V}$. The incidence matrix $\mathbf{H} \in \mathbb{R}^{N \times M}$ encodes this relationship:

$$H_{i,k} = \begin{cases} 1 & \text{if } v_i \in \mathcal{V}_k \\ 0 & \text{otherwise} \end{cases}$$

3.2 Hyperspherical Embedding

To map the hypergraph to a geometric space, we embed nodes and hyperedges into a hyperspherical manifold. The embedding function $\phi : \mathcal{V} \cup \mathcal{E} \to \mathbb{S}^d$ projects nodes and hyperedges onto the d-dimensional unit sphere:

$$\phi(v_i) \in \mathbb{S}^d \text{ and } \phi(e_k) \in \mathbb{S}^d, \text{ where } \mathbb{S}^d = \{\mathbf{x} \in \mathbb{R}^{d+1} : \|\mathbf{x}\|_2 = 1\}. \tag{1}$$

The embeddings are optimized to minimize the hyperspherical distortion while preserving relational and geometric properties. This is achieved via:

$$\mathcal{L}_{\text{embed}} = \sum_{(i,j) \in \mathcal{E}} w_{ij} \cdot \left\|\phi(v_i) - \phi(v_j)\right\|_2^2 + \beta \cdot \sum_{i \in \mathcal{V}} \left(\|\phi(v_i)\|_2^2 - 1\right)^2, \tag{2}$$

$(i, j) \in \mathcal{E}$ is a connected node pair with weight $w_{ij}.\phi(v_i)$ is the hyperspherical embedding, $\left\|\phi(v_i) - \phi(v_j)\right\|_2^2$ enforces proximity, β regulates constraints, and $\|\phi(v_i)\|_2^2 - 1$ penalizes deviations from the unit sphere.

3.3 Sphere Neural Network (SNN)

The SNN operates on hyperspherical embeddings to propagate and aggregate features. It consists of the following layers:

(1) Spherical Convolution: The spherical convolution layer aggregates features along hyperedges using:

$$\mathbf{h}_v^{(l+1)} = \text{ReLU}\left(\sum_{e_k \in \mathcal{N}(v)} \frac{1}{|\mathcal{V}_k|} \sum_{v_j \in \mathcal{V}_k} \mathbf{W}^{(l)} \mathbf{h}_{v_j}^{(l)}\right), \tag{3}$$

where $\mathbf{h}_v^{(l)} \in \mathbb{R}^d$ is the feature of node v at layer l, $\mathcal{N}(v)$ is the set of hyperedges containing v, and $\mathbf{W}^{(l)}$ is the trainable weight matrix.

(2) Spherical Attention Mechanism: An attention mechanism assigns importance scores to hyperedges

$$\alpha_k = \frac{\exp\left(\mathbf{a}^\top \cdot \text{concat}\left(\mathbf{h}_{v_i}, \mathbf{h}_{e_k}\right)\right)}{\sum_{k'} \exp\left(\mathbf{a}^\top \cdot \text{concat}\left(\mathbf{h}_{v_i}, \mathbf{h}_{e_{k'}}\right)\right)}, \tag{4}$$

where $\mathbf{a}$ is a learnable parameter vector. The attention scores α_k are used to weight the hyperedge contributions.

3.4 Task-Specific Objectives

We formulate task-specific loss functions tailored to web-based applications:

(1) Recommendation: For link prediction (e.g., co-purchase or ratings), we maximize the similarity between connected nodes on the hypersphere:

$$\mathcal{L}_{rec} = -\sum_{(i,j) \in \mathcal{E}^+} \log \sigma\big(\phi(v_i)^\top \phi(v_j)\big) - \sum_{(i,j) \in \mathcal{E}^-} \log\left(1 - \sigma\big(\phi(v_i)^\top \phi(v_j)\big)\right). \tag{5}$$

where $\mathcal{E}^+$and $\mathcal{E}^-$are the positive and negative edges, respectively, and σ is the sigmoid function.

(2) Clustering: For community detection, we minimize the intra-cluster variance while maximizing inter-cluster separation:

$$\mathcal{L}_{\text{clust}} = \sum_C \frac{1}{|C|} \sum_{v_i, v_j \in C} \left\|\phi(v_i) - \phi(v_j)\right\|_2^2 - \lambda \sum_{C, C'} \|\mathbf{c}_C - \mathbf{c}_{C'}\|_2^2, \tag{6}$$

where $\mathbf{c}_C$ is the cluster centroid, and λ balances intra-cluster compactness and inter-cluster separation.

3.5 Overall Optimization

The total loss function combines embedding, task-specific, and regularization terms:

$$\mathcal{L} = \mathcal{L}_{\text{embed}} + \gamma \mathcal{L}_{\text{task}} + \lambda \mathcal{L}_{\text{reg}} \tag{7}$$

where $\mathcal{L}$embed preserves relational and geometric properties by keeping connected nodes close and enforcing the hyperspherical constraint; $\mathcal{L}$task optimizes task-specific objectives like link prediction or clustering; and $\mathcal{L}_{\text{reg}}$ prevents overfitting and improves stability by constraining model parameters. The hyperparameters γ and λ (same as used in equation 6) balance task-specific learning and regularization for optimal generalization.

4 Experiments

We evaluate the proposed HyperSNN framework on four widely used benchmark hypergraph-structured web datasets, including Reddit [1], DBLP [17], MovieLens [6], and Amazon Co-Purchase [9], by comparing it with state-of-the-art baselines and conducting an ablation study to analyze the effectiveness of its components.

4.1 Experiment Setup

4.1.1 Datasets and Baselines. Reddit and DBLP focus on link prediction tasks, while MovieLens and Amazon Co-Purchase target recommendation and co-purchase prediction tasks. We compare HyperSNN with baselines, including Graph Convolutional Networks (GCN) [7], Graph Attention Networks (GAT) [18], Hypergraph GCN [21], and a Multi-Layer Perceptron (MLP) [22].

4.1.2 Implementation Details. Node features are normalized for numerical stability and faster convergence. Uniform weights are assigned to hyperedges for consistency across datasets, and the hypergraph is stored as a sparse incidence matrix to optimize memory and computation for large-scale data. Models are trained for 100 epochs using the Adam optimizer with a learning rate of 0.01 and a weight decay of 10^{-4} for regularization. Experiments are conducted on an NVIDIA GeForce 2080 Ti GPU and evaluated on Area Under the Curve (AUC), Precision@10, and Mean Reciprocal Rank (overMRR), with results averaged over five runs using different random seeds to ensure statistical robustness.

4.2 Results

We evaluate HyperSNN across four datasets (Reddit, DBLP, MovieLens, Amazon Co-Purchase), selecting metrics suited to each task. For link prediction, AUC measures the ability to distinguish correct links, while MRR evaluates ranking quality. For recommendation, P@10 assesses top-10 relevance, and MRR ensures accurate ranking. These task-specific metrics provide a fair and meaningful evaluation. The following sections analyze performance, compare baselines, and present ablation insights.

4.2.1 Link Prediction Task. We present the results of link prediction experiments on Reddit [1] and DBLP [17] datasets in Table 1. HyperSNN achieves the highest AUC scores, improving by 6.0% and 5.0% on Reddit and DBLP, respectively, compared to the strongest baseline. AUC evaluates the model's ability to rank positive links higher than negative ones, making it a key metric for link prediction. Furthermore, HyperSNN exhibits low standard deviation across runs, highlighting its stability and robustness in capturing complex hypergraph structures.

Table 1: Link Prediction Results (AUC on Reddit and DBLP)

Model	Reddit		DBLP	
	Mean AUC (%)	Std Dev (%)	Mean AUC (%)	Std Dev (%)
GCN	82.0	2.1	80.0	2.3
GAT	84.0	2.0	83.0	2.4
HyperGCN	85.0	1.9	84.0	2.2
MLP	73.0	3.5	72.0	3.6
HyperSNN	**91.0**	**1.7**	**89.0**	**1.8**

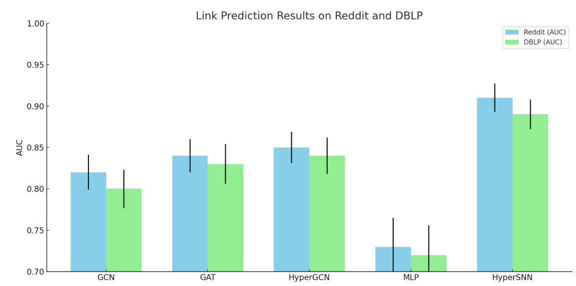

Figure 1: Model Performance for Link Prediction Task

These results highlight HyperSNN's effectiveness in modeling higher-order interactions within hypergraphs. Although traditional models such as GCN [7] and GAT [18] leverage neighborhood aggregation, they struggle to capture the geometric and relational complexity that HyperSNN excels at.

4.2.2 Recommendation and Co-Purchase Prediction Tasks. We evaluate HyperSNN on MovieLens[6] and Amazon Co-Purchase datasets[9] for recommendation and co-purchase prediction tasks. As shown in Table 2, HyperSNN outperforms all baselines in both Precision@10, which measures the relevance of the top 10 recommendations, and MRR, which evaluates the quality of the rankings. This demonstrates HyperSNN's ability to generate and rank relevant recommendations effectively for real-world applications.

Table 2: Recommendation Results (Precision@10 on MovieLens and MRR on Amazon)

Model	MovieLens (Precision@10)		Amazon (MRR)	
	Mean (%)	Std Dev (%)	Mean (%)	Std Dev (%)
GCN	76.0	2.5	70.0	2.7
GAT	78.0	2.3	72.0	2.5
HyperGCN	79.0	2.0	73.0	2.2
MLP	65.0	3.4	60.0	3.5
HyperSNN	**87.0**	**1.8**	**81.0**	**1.9**

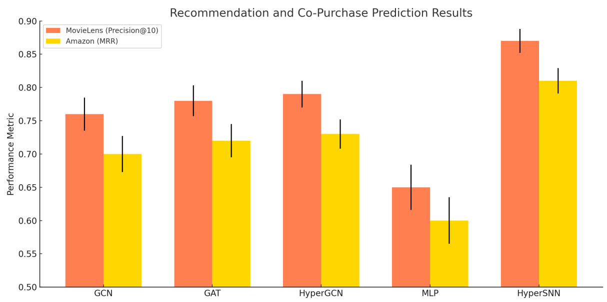

Figure 2: Model Performance for Recommendation and Co-Purchase Prediction Tasks

These results demonstrate that the hyperspherical embeddings learned by HyperSNN provide superior representation power for user-item and product interactions.

4.3 Ablation Study

We perform an ablation study to assess the contribution of key components in HyperSNN by systematically removing the attention mechanism and hyperspherical embeddings. Table 3 and Fig 3 shows that removing the attention mechanism reduces the AUC, demonstrating its importance in assigning relevance to hyperedges.

Table 3: Ablation Study Results Across All Datasets

Model Variant	Reddit (AUC)	DBLP (AUC)	MovieLens (P@10)	Amazon (MRR)
Full Model	**0.91**	**0.89**	**0.87**	**0.81**
Without Attention	0.88	0.85	0.84	0.78
Without Hyperspherical	0.82	0.80	0.79	0.73
Without Both	0.79	0.77	0.75	0.70

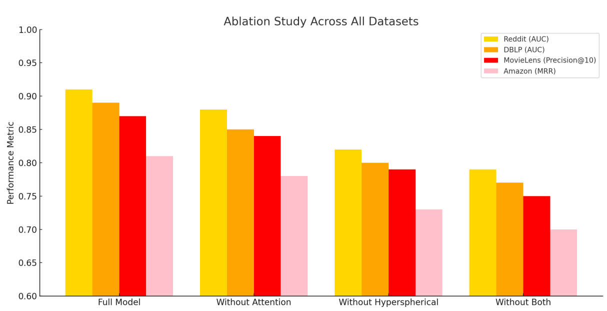

Figure 3: Ablation Study for HyperSNN

Similarly, replacing hyperspherical embeddings with Euclidean embeddings results in a significant drop in performance, highlighting the necessity of geometric representations for capturing relational structures. These results confirm the critical role of both components in HyperSNN's success.

5 Discussion

HyperSNN achieves superior performance across all datasets by leveraging hyperspherical embeddings to capture geometric relationships and an attention mechanism to prioritize critical hyperedges. The ablation study highlights the importance of these components, with significant performance drops when either is removed. The spherical convolution layer ensures effective feature aggregation, preserving the relational and geometric properties of hypergraphs. The model's scalability is evident from its efficient handling of large datasets like Amazon Co-Purchase [9], thanks to sparse representations and batch processing. However, its reliance on hyperspherical embeddings and attention mechanisms increases memory usage. This trade-off is offset by its ability to consistently deliver stable and accurate results, as evidenced by low standard deviations across runs. HyperSNN's success underscores the value of integrating geometric learning with hypergraph structures, paving the way for further advancements in this domain.

6 Conclusion

We introduced HyperSNN, a novel framework that integrates hyperspherical embeddings and attention mechanisms to effectively hypergraph modeling. By leveraging geometric representations and dynamic edge weighting, HyperSNN achieves strong performance in link prediction, recommendation and co-purchase prediction. Experiments on Reddit, DBLP, MovieLens, and Amazon Co-Purchase datasets demonstrate its superiority over state-of-the-art baselines, achieving significant improvements in AUC, Precision@10, and MRR.

The ablation study confirms the importance of hyperspherical embeddings and attention mechanisms, with performance drops observed when either is removed. HyperSNN also scales efficiently on large datasets using sparse representations, though its memory requirements remain a limitation. Future work can focus on improving memory efficiency, enhancing robustness to noisy data, and extending HyperSNN to dynamic hypergraphs, further expanding its applicability in real-world scenarios.

References

[1] Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *Proceedings of the international AAAI conference on web and social media*, Vol. 14. 830–839.
[2] C Berge and E Minieka. 1973. Graphs and Hypergraphs. Graphs and Hypergraphs.
[3] Tiansi Dong, Mateja Jamnik, and Pietro Liò. 2024. Sphere Neural-Networks for Rational Reasoning. *arXiv preprint arXiv:2403.15297* (2024).
[4] Fuli Feng, Huimin Chen, Xiangnan He, Jie Ding, Maosong Sun, and Tat-Seng Chua. 2019. Enhancing Stock Movement Prediction with Adversarial Training.. In *IJCAI*, Vol. 19. 5843–5849.
[5] Anoushka Harit, Zhongtian Sun, Jongmin Yu, and Noura Al Moubayed. 2024. Breaking Down Financial News Impact: A Novel AI Approach with Geometric Hypergraphs. *arXiv preprint arXiv:2409.00438* (2024).
[6] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.
[7] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[8] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
[9] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. 2007. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)* 1, 1 (2007), 5–es.
[10] Weiyang Liu, Zhen Liu, James M Rehg, and Le Song. 2019. Neural similarity learning. *Advances in Neural Information Processing Systems* 32 (2019).
[11] Danilo Montesi and Giacomo Bergami. [n. d.]. Hypergraph Mining for Social Networks. ([n. d.]).

[12] Zhongtian Sun. 2024. *Robustness, Heterogeneity and Structure Capturing for Graph Representation Learning and its Application.* Ph. D. Dissertation. Durham University.
[13] Zhongtian Sun, Anoushka Harit, Alexandra I Cristea, Jingyun Wang, and Pietro Lio. 2023. Money: Ensemble learning for stock price movement prediction via a convolutional network with adversarial hypergraph model. *AI Open* 4 (2023), 165–174.
[14] Zhongtian Sun, Anoushka Harit, Alexandra I Cristea, Jingyun Wang, and Pietro Lio. 2023. A Rewiring Contrastive Patch PerformerMixer Framework for Graph Representation Learning. In *2023 IEEE International Conference on Big Data (BigData)*. IEEE Computer Society, 5930–5939.
[15] Zhongtian Sun, Anoushka Harit, Alexandra I Cristea, Jialin Yu, Lei Shi, and Noura Al Moubayed. 2022. Contrastive learning with heterogeneous graph attention networks on short text classification. In *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–6.
[16] Zhongtian Sun, Anoushka Harit, Jialin Yu, Alexandra I Cristea, and Noura Al Moubayed. 2021. A generative bayesian graph attention network for semi-supervised classification on scarce data. In *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–7.
[17] The DBLP Team. 2019. DBLP Computer Science Bibliography: Monthly Snapshot Release of November 2019. https://dblp.org/xml/release/dblp-2019-11-01.xml.gz. Accessed: 2019-11-01.
[18] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. 2017. Graph attention networks. *stat* 1050, 20 (2017), 10–48550.
[19] Richard C Wilson, Edwin R Hancock, Elżbieta Pekalska, and Robert PW Duin. 2014. Spherical and hyperbolic embeddings of data. *IEEE transactions on pattern analysis and machine intelligence* 36, 11 (2014), 2255–2269.
[20] Adam Wynn, Jingyun Wang, Zhongtian Sun, and Atsushi Shimada. 2024. Analysing Learner Behaviour in an Ontology-Based E-learning System: A Graph Neural Network Approach. (2024).
[21] Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. 2019. Hypergcn: A new method for training graph convolutional networks on hypergraphs. *Advances in neural information processing systems* 32 (2019).
[22] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)* 52, 1 (2019), 1–38.
[23] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. 2006. Learning with hypergraphs: Clustering, classification, and embedding. *Advances in neural information processing systems* 19 (2006).