Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Hierarchical Compositional Hypergraphs Encode Document Structure for Classification

Madjid Eshaghi Gordji
Semnan University
Mohamadali Berahman
mohamadali_berahman@semnan.ac.ir
Semnan University

Research Article

Keywords: directed hypergraphs, hierarchical document representation, text classification, feature hashing, document structure, higher-order networks

Posted Date: September 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10829259/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Hierarchical Compositional Hypergraphs Encode Document Structure for Classification

Madjid Eshaghi Gordji[1] and Mohamadali Berahman[1,*]

[1]Faculty of Mathematics, Statistics and Computer Science, Semnan University, Semnan, Iran

[*]Corresponding author: mohamadali_berahman@semnan.ac.ir

Abstract

Flat text representations discard document boundaries, whereas ordinary word graphs usually collapse structure into a single co-occurrence relation. We introduce a *Hierarchical Compositional Hypergraph* (HCH) that represents token occurrences, sentences, paragraphs, and a document root in ordered layers. Directed composition hyperedges map ordered constituent tuples to their parent units. This occurrence-level formulation is acyclic, admits exact reconstruction of the segmented token sequence, and has linear construction size. For classification, we derive a sparse word-type feature map with three typed channels: within-sentence pairs, within-paragraph pairs, and ordered within-sentence adjacencies. A linear support-vector classifier combines this structural map with TF–IDF.

We evaluate a configuration locked on a stratified split of the official 20 Newsgroups training set and then applied once to the date-based test set (11,314/7,532 documents; headers, footers, and quotations removed). The combined TF–IDF+HCH model obtains 0.6995 accuracy and 0.6873 macro-F1, compared with 0.6920 and 0.6802 for a word unigram–bigram TF–IDF baseline. Paired differences are 0.0076 in accuracy (95% stratified-bootstrap interval 0.0021–0.0131) and 0.0071 in macro-F1 (0.0013–0.0133); the Holm-adjusted exact McNemar value is $p = 0.0098$. HCH features alone do not outperform lexical baselines. Ablation shows that order is the most consequential structural channel, while sentence- and paragraph-level pair channels are partly redundant. The results therefore support HCH as a modest, interpretable complement to lexical features, not as a replacement for them or as a state-of-the-art neural model.

Keywords: directed hypergraphs; hierarchical document representation; text classification; feature hashing; document structure; higher-order networks.

1 Introduction

Document classification requires a representation that is both discriminative and computationally manageable. Bag-of-words and TF–IDF remain strong baselines because sparse lexical features interact effectively with linear classifiers [13, 14]. Their central limitation is structural: documents with the same term multiset receive the same representation even if their sentence boundaries, paragraph grouping, or word order differ. Dense document embeddings and contextual language models address semantics and context [3, 9], while hierarchical neural architectures explicitly aggregate words into sentences and sentences into documents [18]. These methods are effective, but their learned states do not by themselves provide a discrete mathematical object that records the complete observed composition hierarchy.

Graph-of-words models provide a complementary route. They represent word types as vertices and local co-occurrences as edges, enabling graph classification, graph kernels, or graph neural networks [12, 15, 16, 19]. A usual projection nevertheless combines all observed relations into one

graph. Sentence membership, paragraph membership, and order may then be lost or represented only indirectly. Hypergraphs can encode relations of arbitrary arity, and directed hypergraphs distinguish tails from heads [6, 20]. They are therefore natural candidates for representing a sentence as the composition of several tokens and a paragraph as the composition of several sentences.

This paper develops an HCH as a rooted ordered directed hypergraph. A crucial correction relative to type-level formulations is that the primitive vertices are *token occurrences*, accompanied by a type map. Repeated words are thus distinguishable, and the ordered segmented document can be reconstructed. Word types appear only in the downstream projection used for classification. This separation prevents an ambiguity in which repeated lexical items would otherwise be simultaneously treated as one primitive vertex and as several ordered constituents.

The empirical question is deliberately narrow: does a sparse, typed projection of this hierarchy add information to a competitive lexical classifier? We use the official date-based 20 Newsgroups split and remove metadata that can make the task artificially easy, following the benchmark recommendation implemented in scikit-learn [8, 11]. Hyperparameters are selected on training data only, after which the test set is evaluated once. The main contributions are:

1. an occurrence-level definition of ordered HCHs with explicit type and layer maps;
2. proofs of acyclicity, rooted-tree structure, lossless reconstruction, linear size, and positive semidefiniteness of the induced explicit kernel;
3. a sparse three-channel projection that avoids the dense document-wide clique responsible for degenerate graph representations;
4. a leakage-controlled experiment with strong lexical and flat-graph baselines, convergence checks, stratified-bootstrap intervals, paired exact tests, and component ablations.

The contribution is representational rather than a claim of neural state of the art. The observed improvement is statistically supported but small, and the negative results for the stand-alone HCH projection are reported alongside the positive hybrid result.

2 Related work

2.1 Flat, distributed, and hierarchical text representations

Sparse term-weighting methods remain widely used because they are transparent and scale well [13, 14]. Paragraph vectors learn dense representations intended to retain distributed information beyond term counts [9], and BERT conditions token representations on bidirectional context [3]. Hierarchical Attention Networks use distinct word- and sentence-level encoders [18]. These approaches demonstrate the usefulness of context and granularity, but they solve a different problem from an explicit combinatorial representation of observed document composition.

2.2 Graphs, graph kernels, and text

Co-occurrence networks reveal non-trivial organization in language [5]. Graph-of-words methods convert a document into a graph whose edges reflect windowed co-occurrence, and graph classification can extract substructures that are unavailable to unigram bags [12]. General graph kernels compare structured objects through shared walks or substructures [16]; the Weisfeiler–Lehman family provides an efficient discrete-label feature map [15]. Corpus-level methods such as TextGCN connect documents and words in a heterogeneous graph [19]. HCH differs by treating the internal segmentation of each document as the primary object and by retaining ordered higher-arity composition before projection.

2.3 Hypergraphs and multilayer structure

Directed hypergraphs generalize directed edges by allowing tail and head sets and support path and cut notions [6]. Spectral hypergraph learning and hypergraph neural networks use higher-order

associations for classification and representation learning [4, 20]. Multilayer networks distinguish interaction contexts [2], while higher-order network theory emphasizes that group interactions cannot always be reduced without loss to pairwise links [1]. HCH uses both ideas but imposes a stricter compositional constraint: every non-root occurrence belongs to exactly one immediate parent, and layer indices increase along every composition arc.

3 Hierarchical Compositional Hypergraphs

3.1 Segmented documents and ordered hyperedges

Let a segmented document be a finite token sequence together with nested sentence and paragraph boundaries. Its layer set is $\mathcal{L} = \{0, 1, \dots, K\}$, with $K = 3$ for tokens, sentences, paragraphs, and the document root. Let V_ℓ be the vertices in layer ℓ, and assume a disjoint union

$$V = \bigsqcup_{\ell=0}^{K} V_\ell, \qquad \lambda : V \to \mathcal{L}.$$

Each $v \in V_0$ is one token *occurrence*. A type map $\tau : V_0 \to \Sigma$ assigns its lexical string in vocabulary Σ.

An ordered composition hyperedge is a pair $e = (T_e, h_e)$, where $T_e = (v_1, \dots, v_m)$ is an ordered tuple of vertices from V_ℓ and $h_e \in V_{\ell+1}$. Repetition is unnecessary because token occurrences are distinct; order in T_e follows the document. Every non-primitive vertex has one incoming composition hyperedge, and every vertex except the root occurs in exactly one tail tuple at the next layer.

Definition 3.1 (HCH)**.** *An HCH is the tuple*

$$\mathcal{H} = (V, E, \lambda, \tau, r),$$

where E is a set of ordered composition hyperedges satisfying the adjacent-layer and unique-parent conditions above, τ is defined on V_0, and r is the unique vertex in V_K.

Figure 1 shows an incidence representation. Diamonds are hyperedge nodes used only for visualization; the mathematical hyperedge remains the ordered tuple-to-head relation.

3.2 Structural properties

Define the composition digraph $G_c = (V, A)$ by adding $(v, h_e) \in A$ whenever v occurs in T_e.

Theorem 3.2 (Acyclicity)**.** *The composition digraph of an HCH is a directed acyclic graph.*

Proof. Every arc (v, h) satisfies $\lambda(h) = \lambda(v) + 1$. Along a directed path of length q, the layer index therefore increases by q. A directed cycle would return to its starting vertex with a strictly larger layer index, a contradiction. □

Proposition 3.3 (Rooted arborescence)**.** *After ignoring arc direction, G_c is connected; in the upward orientation, every $v \neq r$ has out-degree one and r has out-degree zero. Consequently, G_c is a rooted arborescence directed toward r.*

Proof. The unique-parent condition gives out-degree one for every non-root vertex. Repeated parent traversal strictly increases λ and must terminate at layer K, which contains only r. Thus every vertex has a path to r, proving connectedness. Acyclicity then yields an arborescence. □

Proposition 3.4 (Lossless reconstruction)**.** *Given $\mathcal{H}$, the ordered token sequence and all represented boundaries are uniquely recoverable.*

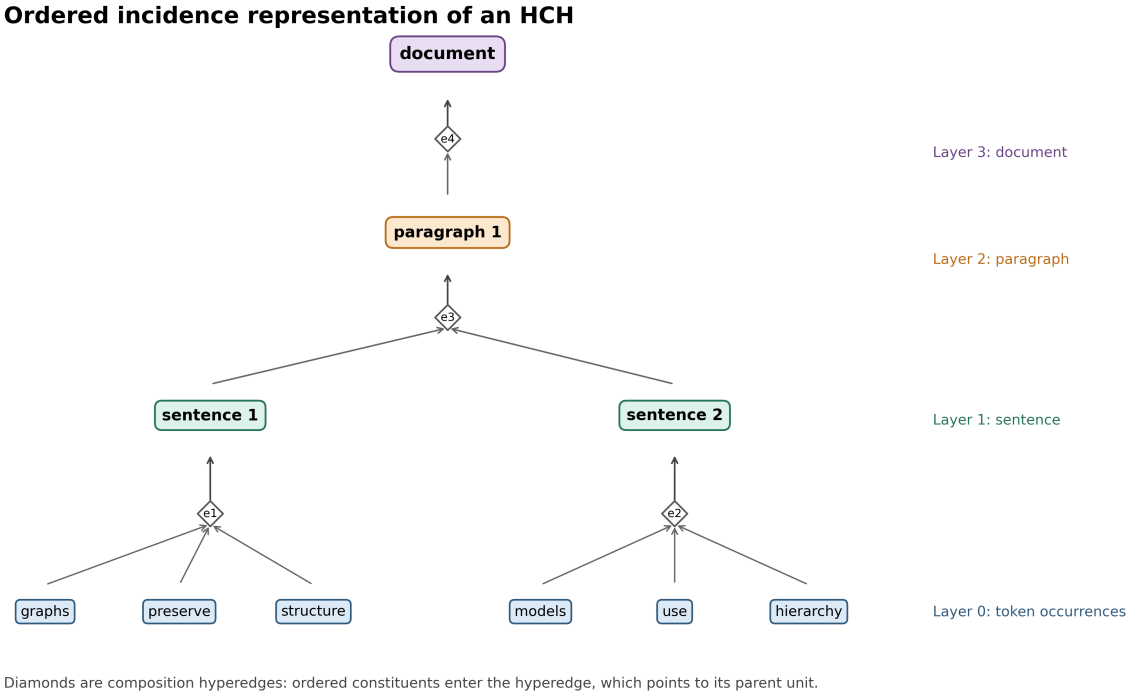


Figure 1: Ordered incidence representation of an HCH. Token occurrences compose sentence vertices, sentences compose a paragraph, and the paragraph composes the document root.

Proof. For $v \in V_0$, return $\tau(v)$. For a higher-layer head h, recursively expand the unique hyperedge (T, h) and concatenate the expansions in the order of T. Unique hyperedges and finite depth make the recursion deterministic; retaining the expansion points preserves sentence and paragraph boundaries. □

Proposition 3.5 (Linear representation size)**.** *If $n_\ell = |V_\ell|$, then $|V| = \sum_{\ell=0}^{K} n_\ell$, $|E| = \sum_{\ell=1}^{K} n_\ell$, and the total number of tail incidences is $\sum_{\ell=0}^{K-1} n_\ell$. Construction time and storage are $O(|V|)$.*

Proof. Each non-primitive vertex has one composition hyperedge. Each non-root vertex appears once in the tail tuple of its parent. The stated counts follow, and a single pass through the segmented document constructs all vertices and incidences. □

3.3 Typed sparse projection

Let $c_i(u)$ be the number of token descendants of unit u with type $i \in \Sigma$. A naive word-type projection that connects every pair sharing the document root creates an almost complete graph. Such a graph suppresses discriminative topology and caused the collapse observed in the original code. We instead use local typed channels on a document-specific vocabulary Σ_D:

$$\phi_S(i,j) = \sum_{s \in V_1} \mathbf{1}\{c_i(s) > 0,\ c_j(s) > 0\}, \quad i < j, \tag{1}$$

$$\phi_P(i,j) = \sum_{p \in V_2} \mathbf{1}\{c_i(p) > 0,\ c_j(p) > 0\}, \quad i < j, \tag{2}$$

$$\phi_O(i,j) = \#\{\text{adjacent occurrences of type } i \text{ followed by } j \text{ within a sentence}\}. \tag{3}$$

The first two channels represent unordered higher-order membership after word-type projection; the third uses the order in sentence hyperedges. The structural vector is

$$\phi_{\mathrm{HCH}}(D) = [\phi_S(D), \phi_P(D), \phi_O(D)].$$

We retain at most 80 word types and the 600 strongest structural coordinates per document. Feature strings are mapped to 2^{20} non-negative hashed coordinates and transformed with sublinear TF–IDF fitted on training data only. Feature hashing provides a fixed sparse space without a global pair dictionary [17]. The final model concatenates ordinary lexical TF–IDF and the normalized structural block:

$$\phi(D) = [\phi_{\text{lex}}(D),\ \gamma\phi_{\text{HCH}}(D)], \qquad \gamma = 1.$$

Proposition 3.6 (Valid explicit kernel)**.** $K(D, D') = \langle \phi(D), \phi(D') \rangle$ *is positive semidefinite.*

Proof. For arbitrary documents $D_1, \ldots, D_m$ and coefficients $a_1, \ldots, a_m$,

$$\sum_{r,q} a_r a_q K(D_r, D_q) = \left\| \sum_r a_r \phi(D_r) \right\|_2^2 \geq 0.$$

□

The classifier is a one-versus-rest linear SVM applied to this explicit map. This differs from claiming that the algorithm directly evaluates a Weisfeiler–Lehman kernel on the full HCH; the earlier notebook conflated those procedures.

4 Experimental protocol

4.1 Dataset and preprocessing

We use the official date-based 20 Newsgroups split: 11,314 training and 7,532 test documents in 20 classes. Headers, footers, and quoted material are removed. These filters reduce metadata leakage but are heuristic for footers and quotations. The resulting training corpus contains 2,084,370 lexical token occurrences, 140,793 detected sentences, and 55,359 detected paragraphs. A deterministic regular-expression tokenizer and boundary parser are used; 311 training documents become empty after the benchmark filters. Such documents receive zero lexical and structural vectors and do not instantiate non-empty composition hyperedges.

4.2 Model selection and baselines

An 80/20 stratified split of the official training data (seed 42) is used to lock the feature budget, structural scale, and C. The official test labels are not used during selection. The locked hybrid improves validation accuracy from 0.7552 to 0.7614 and macro-F1 from 0.7486 to 0.7522. We compare:

- word-unigram TF–IDF ($C = 1$);
- word unigram–bigram TF–IDF ($C = 2$);
- a flat width-five co-occurrence graph encoded by lexical and unordered pair features ($C = 2$);
- a stand-alone HCH projection with lexical and structural features ($C = 2$);
- the proposed concatenation of lexical TF–IDF and structural HCH channels ($C = 1$, $\gamma = 1$).

All vectorizer statistics are fitted only on training documents. Every LinearSVC run uses `dual=auto`, tolerance 10^{-4}, and a 20,000-iteration ceiling. All models converge in at most 75 iterations. The implementation uses scikit-learn [11]; the complete script, notebook, metadata, predictions, and output tables accompany the manuscript.

4.3 Metrics and uncertainty

Primary metrics are accuracy and macro-F1. We compute 95% percentile intervals from 1,000 paired stratified bootstrap resamples of the test set. Pairwise differences use the same resamples. Accuracy errors are compared by the two-sided exact McNemar test [10]; four comparisons with the proposed model are adjusted by Holm's step-down method [7]. Runtime is reported for reproducibility but not treated as hardware-independent evidence.

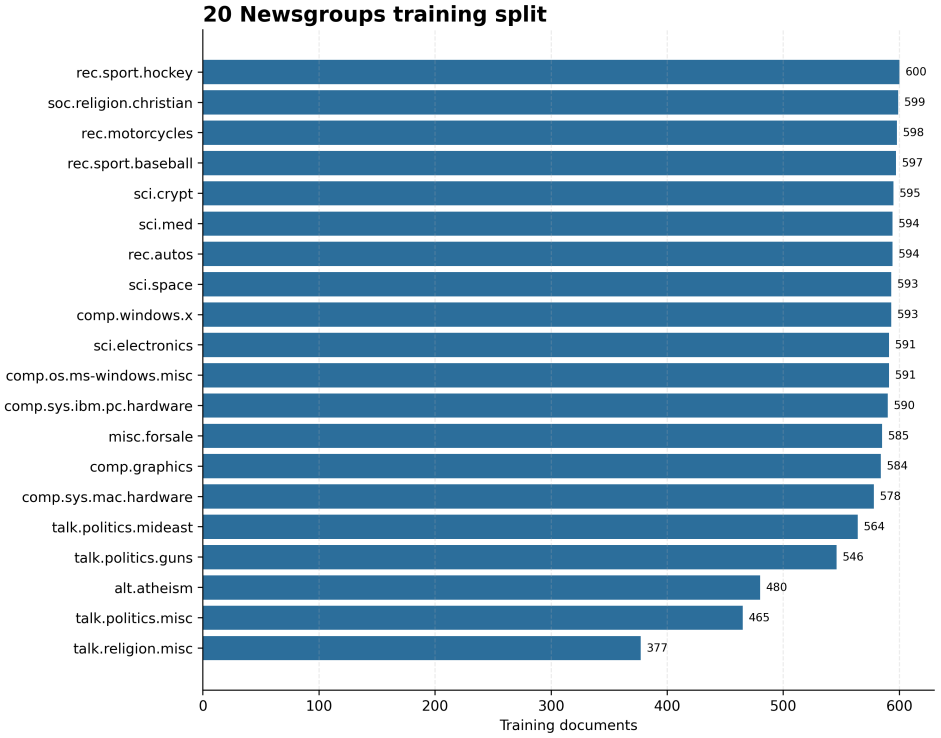


Figure 2: Class distribution in the official training split. The smallest class has 377 documents and the largest has 600.

Table 1: Official test-set performance. Parentheses contain stratified-bootstrap 95% intervals.

Method	Accuracy	Macro-F1
TF–IDF	0.6868 (0.6772–0.6974)	0.6760 (0.6659–0.6872)
TF–IDF 1–2 grams	0.6920 (0.6826–0.7025)	0.6802 (0.6705–0.6907)
Flat co-occurrence	0.6717 (0.6617–0.6823)	0.6577 (0.6476–0.6690)
HCH projection	0.6467 (0.6369–0.6581)	0.6301 (0.6199–0.6410)
TF–IDF + HCH	**0.6995 (0.6893–0.7104)**	**0.6873 (0.6775–0.6983)**

5 Results

5.1 Classification performance

Table 1 reports the locked test results. The combined TF–IDF+HCH model yields the highest accuracy and macro-F1. Relative to unigram–bigram TF–IDF, its accuracy gain is 0.0076 (95% paired interval 0.0021–0.0131) and macro-F1 gain is 0.0071 (0.0013–0.0133). The exact McNemar test remains significant after Holm adjustment ($p = 0.0098$). Against unigram TF–IDF, the gains are 0.0127 in accuracy and 0.0113 in macro-F1 (adjusted $p = 2.80 \times 10^{-7}$).

The structural representations alone do not beat lexical TF–IDF. In particular, the standalone HCH projection is 0.0401 below TF–IDF in accuracy. This negative result shows that the HCH channels are useful as a residual structural signal rather than as a self-sufficient topic representation. Lexical identity remains essential for this benchmark.

5.2 Ablation

For ablation, $C = 1$ and $\gamma = 1$ remain locked. Removing order reduces macro-F1 from 0.6873 to 0.6845. In contrast, removing the sentence channel or paragraph channel gives 0.6892 and 0.6897, respectively (Table 2). The result does not support a claim that all channels are

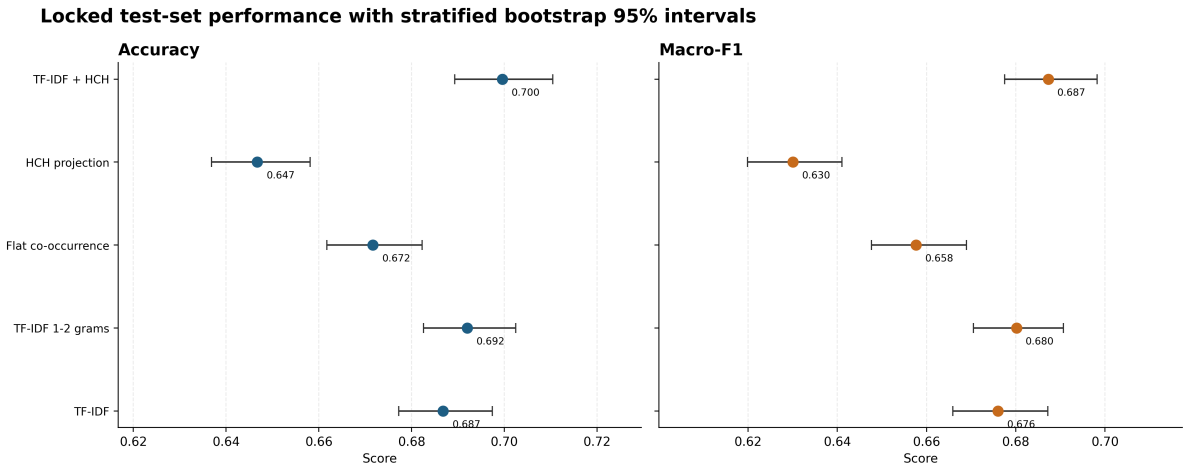


Figure 3: Point estimates and stratified-bootstrap 95% intervals on the locked test set. A dot-and-whisker design is used because truncated bar axes would exaggerate the small differences.

Table 2: Component ablation using the locked classifier and structural scale.

Structural channels	Accuracy	Macro-F1
Paragraph + order (without sentence)	0.7018	0.6892
Sentence + order (without paragraph)	0.7017	0.6897
Sentence + paragraph (without order)	0.6966	0.6845
Full: sentence + paragraph + order	0.6995	0.6873

complementary. Sentence and paragraph co-membership encode overlapping pair evidence, and their joint inclusion adds some noise under a fixed 600-coordinate budget. Order is the channel whose removal consistently damages the locked full model.

6 Discussion

The corrected experiment changes the interpretation of HCH in three important ways. First, a document-wide clique is not an adequate operationalization of hierarchy. All word pairs share the document root, so adding a constant document-level relation makes graphs dense and topologically similar. The very low 0.0633 accuracy in the earliest notebook and the later 0.5643 result were symptoms of this construction, compounded by non-converged SVM fits. The corrected local projection eliminates the clique and checks convergence explicitly.

Second, hierarchy does not replace lexical content. Topics in 20 Newsgroups are strongly associated with vocabulary, and a stand-alone structural projection removes too much direct term evidence. The statistically supported result is narrower: typed HCH features improve a lexical model by about 0.8 accuracy percentage points over the stronger unigram–bigram baseline. The gain is meaningful for evaluating the representation, but too small to justify broad performance claims.

Third, the observed benefit is not evenly distributed across hierarchical levels. Ordered adjacency is most useful, whereas sentence and paragraph pair channels are partially redundant. This finding is consistent with the fact that a paragraph contains its sentences: many pairs counted at the sentence level are counted again at the paragraph level. A future projection could orthogonalize channels, for example by assigning a pair only to its smallest common compositional ancestor.

The formal HCH remains valuable independently of the particular classifier. Its occurrence-level vertices make the representation lossless, its directed incidence structure exposes parent-child

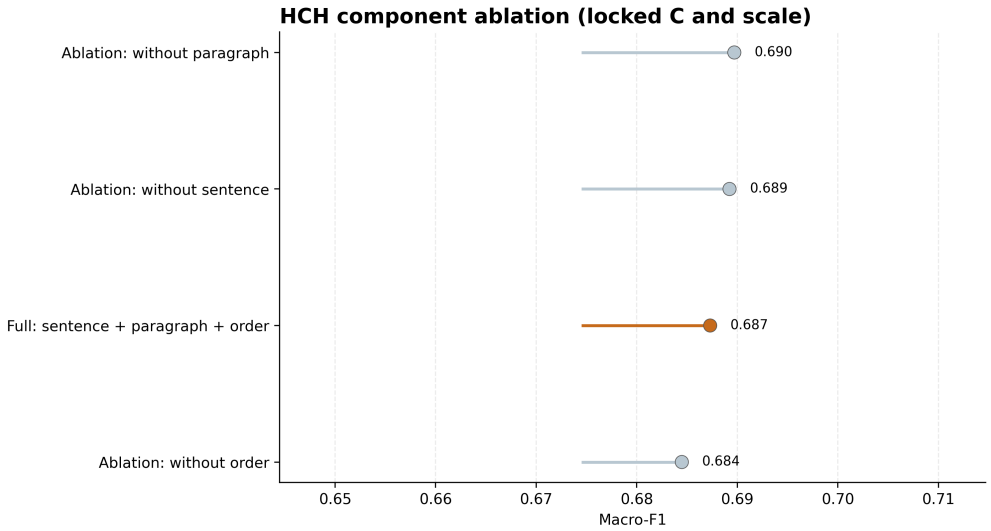


Figure 4: Macro-F1 for the locked HCH component ablations. Reduced variants are diagnostic results, not models selected after observing the test labels.

composition, and its linear size permits alternative downstream operators. Possible extensions include kernels defined directly on ordered hyperedges, hypergraph neural message passing, learned channel weights, and discourse-aware segment types.

6.1 Limitations

The study has six main limitations. (i) It evaluates one English benchmark, so cross-domain and cross-lingual generalization are unknown. (ii) Sentence and paragraph boundaries are obtained by deterministic heuristics rather than a linguistic parser. (iii) Metadata removal produces 311 empty training documents and uses imperfect footer/quote filters. (iv) Feature hashing can create collisions, although the 2^{20} space is large. (v) The 80-type and 600-coordinate caps trade completeness for feasibility. (vi) We do not compare with current pretrained language models; the objective is an interpretable structural complement to sparse lexical baselines, not state-of-the-art leaderboard performance. These limitations make the work a controlled proof of concept. A stronger empirical claim would require multiple corpora, repeated development splits, neural baselines, and an external replication.

7 Conclusion

We introduced an occurrence-level Hierarchical Compositional Hypergraph that records how tokens compose sentences, sentences compose paragraphs, and paragraphs compose a document. The corrected definition is acyclic, reconstructible, and linear in document size. A sparse projection with sentence, paragraph, and order channels significantly improves a locked TF–IDF classifier on the metadata-filtered 20 Newsgroups test set, but only when used as a complement to lexical features. The stand-alone HCH projection is weaker, and ablation identifies order rather than uniform multi-level complementarity as the main contributor. These results support a precise and limited conclusion: explicit compositional structure can add modest predictive information to flat lexical representations, provided the projection avoids document-wide densification and the evaluation is leakage-controlled.

Declarations

Funding

The authors received no specific funding for this work.

Competing interests

The authors declare no competing interests.

Data availability

The 20 Newsgroups by-date dataset used in this study is publicly available from the dataset homepage at http://qwone.com/~jason/20Newsgroups/ and through the scikit-learn loader at https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups.html. Headers, footers, and quotations were removed using the documented loader options. No raw documents are redistributed with this study. The accompanying reproducibility package contains the code, configuration, predictions, aggregate results, and figure-generation scripts required to reproduce the analyses.

Code availability

The accompanying reproducibility package contains the complete Python script and notebook, locked configuration, environment metadata, aggregate tables, per-class metrics, predictions, and figure-generation code. Running the script downloads the checksum-verified public dataset when it is not already cached.

Consent for Publication

Not Applicable.

Author contributions

M.E.G. conceived the central representation idea and contributed to the theoretical formulation. M.B. developed the methodology and software, performed validation and visualization, and prepared the initial manuscript draft. Both authors analyzed the results, revised the mathematical and empirical presentation, reviewed the final manuscript, and accept responsibility for the work.

Ethics statement

This study uses a public text-classification benchmark and involves no human participants, identifiable private data, animals, or plants. Ethics approval and informed consent were not applicable.

Use of artificial intelligence

Generative AI tools assisted with language editing and with debugging and restructuring analysis code. The authors reviewed the mathematical definitions, experimental design, source code, numerical outputs, interpretation, citations, and final manuscript, and take full responsibility for the content.

References

[1] Federico Battiston, Giulia Cencetti, Iacopo Iacopini, Vito Latora, Maxime Lucas, Alice Patania, Jean-Gabriel Young, and Giovanni Petri. Networks beyond pairwise interactions: Structure and dynamics. *Physics Reports*, 874:1–92, 2020. doi: 10.1016/j.physrep.2020.05.004.

[2] Stefano Boccaletti, Ginestra Bianconi, Raul Criado, Clara I. del Genio, Jesús Gómez-Gardeñes, Miguel Romance, Ignacio Sendiña-Nadal, Zhen Wang, and Massimiliano Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 2014. doi: 10.1016/j.physrep.2014.07.001.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019. doi: 10.18653/v1/N19-1423.

[4] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3558–3565, 2019. doi: 10.1609/aaai.v33i01.33013558.

[5] Ramon Ferrer i Cancho and Ricard V. Solé. The small world of human language. *Proceedings of the Royal Society B*, 268(1482):2261–2265, 2001. doi: 10.1098/rspb.2001.1800.

[6] Giorgio Gallo, Giustino Longo, Stefano Pallottino, and Sang Nguyen. Directed hypergraphs and applications. *Discrete Applied Mathematics*, 42(2–3):177–201, 1993. doi: 10.1016/0166-218X(93)90045-P.

[7] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

[8] Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339, 1995.

[9] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, 2014. URL https://proceedings.mlr.press/v32/le14.html.

[10] Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947. doi: 10.1007/BF02295996.

[11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. URL https://jmlr.org/papers/v12/pedregosa11a.html.

[12] François Rousseau, Emmanouil Kiagias, and Michalis Vazirgiannis. Text categorization as a graph classification problem. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 1702–1712, 2015. doi: 10.3115/v1/P15-1164.

[13] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988. doi: 10.1016/0306-4573(88)90021-0.

[14] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002. doi: 10.1145/505282.505283.

[15] Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler–lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561, 2011. URL https://jmlr.org/papers/v12/shervashidze11a.html.

[16] S. V. N. Vishwanathan, Nicol N. Schraudolph, Risi Kondor, and Karsten M. Borgwardt. Graph kernels. *Journal of Machine Learning Research*, 11:1201–1242, 2010. URL https://jmlr.org/papers/v11/vishwanathan10a.html.

[17] Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th International Conference on Machine Learning*, pages 1113–1120, 2009. doi: 10.1145/1553374.1553516.

[18] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*, pages 1480–1489, 2016. doi: 10.18653/v1/N16-1174.

[19] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377, 2019. doi: 10.1609/aaai.v33i01.33017370.

[20] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in Neural Information Processing Systems*, volume 19, pages 1601–1608, 2006. URL https://papers.nips.cc/paper/3128-learning-with-hypergraphs-clustering-classification-and-embedding.