Multi-level Fusion for Academic Text Structure-Function Recognition (Postprint)

Authors: Wang Jiamin, Lu Wei, Liu Jiawei, Cheng Qikai

Date: 2023-07-26T00:00:00+00:00

Abstract

[Purpose/Significance] Academic text structural function represents a generalization of the structure and chapter functions of academic literature. To address the limitations that current research seldom integrates multi-level structures of academic texts and that traditional methods rely on manual experience for rule or feature construction, this paper proposes a multi-level fused academic text structural function recognition model based on parsing the hierarchical structure of academic texts. [Methodology/Process] Experiments are conducted on the ScienceDirect dataset. The model first employs deep learning methods to recognize structural functions of academic texts at different levels, and subsequently adopts a voting method to fuse recognition results from different levels and different models. [Results/Conclusion] Research results demonstrate that the integrated performance across all levels shows varying degrees of improvement compared to single models. The comprehensive results achieve overall accuracy, recall, and F1-score of 86%, 84%, and 84%, respectively. Furthermore, deep learning algorithms exhibit superior performance over the traditional machine learning algorithm SVM in academic text classification tasks. Finally, misclassification cases of academic text structural functions are analyzed, and potential application areas and future research directions of this study are identified.

Full Text

Research on Structure Function Recognition of Academic Text Based on Multi-level Fusion

Wang Jiamin[1],[2], Lu Wei[1],[2], Liu Jiawei[1],[2], Cheng Qikai[1],[2]

[1]School of Information Management, Wuhan University, Wuhan 430072
[2]Information Retrieval and Knowledge Mining Laboratory, Wuhan University, Wuhan 430072

Abstract

[**Purpose/Significance**] The structure function of academic text represents a generalization of the structure and section functions of academic literature. Existing research has rarely approached fusion from the perspective of multi-level academic text structure, and traditional methods rely on manual experience to construct rules or features. To address these limitations, this study constructs a multi-level fusion model for academic text structure function recognition based on parsing the hierarchical structure of academic texts. [**Method/Process**] Using the ScienceDirect dataset as an example, the model first applies deep learning methods to identify structure functions at different levels of academic text, then employs a voting method to fuse results from different levels and models. [**Result/Conclusion**] Results demonstrate that the integrated performance across all levels improves to varying degrees compared with single models, with overall precision, recall, and F1-score reaching 86%, 84%, and 84%, respectively. Deep learning algorithms outperform the traditional machine learning algorithm SVM in academic text classification tasks. Finally, we analyze misclassification cases of academic text structure functions and identify potential application areas and future research directions.

Keywords: deep learning; structure function; multi-level fusion; academic text

1 Introduction

In recent years, scientific research output has experienced explosive growth. For instance, Microsoft Academic contained 168 million records as of March 2017, growing at a rate of 1.3 million records per month. Academic papers constitute the primary information source for researchers, who typically access them in a goal-driven manner, focusing on specific sections such as methods, results, or literature reviews. The importance and interest of different structural sections vary among scholars. While academic text structures have become increasingly standardized, particularly with the adoption of IMRaD structure in biology, this format has not been universally adopted across all disciplines. Nevertheless, numerous studies have demonstrated that structure function significantly enhances research in information retrieval, keyword extraction, and citation analysis, making automatic recognition of structure functions in large-scale academic texts both theoretically significant and practically valuable.

Current research on academic text structure function recognition primarily approaches the problem from the perspective of document logical structure, employing rule-based or machine learning methods to identify different levels of logical structure, including title recognition, section recognition, and paragraph recognition. While these methods have achieved certain effectiveness, two major problems persist. First, existing approaches identify different structural components separately rather than fusing multi-level structural features from the overall hierarchical perspective. In reality, different levels of academic text contain distinct features and semantic information, and integrating these compo-

nents can provide more complete and accurate judgments. Second, traditional rule-based or machine learning methods require manual construction of rules or features, with performance heavily dependent on human experience and limited transferability. Deep learning can automatically complete data representation and feature extraction, learning effective representations at different levels and dimensions to enhance data interpretability, offering advantages such as incremental learning and strong transferability.

To address these issues, this study parses the multi-level structure of academic texts, divides academic text bodies into five structure function categories, and constructs a multi-level fusion model for academic text structure function recognition using deep learning and voting methods, tested on a computer linguistics academic text dataset.

2 Related Research

This study focuses on analyzing and recognizing academic text structure functions from a content perspective. Existing research primarily centers on document logical structure and can be categorized into rule-based methods and machine learning methods.

Rule-based approaches manually construct rules from document layout and textual features to partition academic text structures. For example, J. Kim et al. built rules through document layout analysis and OCR result feature extraction to automatically annotate titles, authors, affiliations, and abstracts in biomedical literature. A. Constantin et al. designed a rule-based system called PDFX to reconstruct the logical structure of PDF-format academic texts and describe them semantically.

Machine learning methods transform structure recognition into text classification problems. M. T. Luong et al. employed Conditional Random Fields to identify logical structures such as titles, authors, abstracts, figures, and formulas. S. Tuarob et al. used random forest, SVM, and Naive Bayes to automatically recognize section semantics by identifying section boundaries, achieving 92.38% accuracy on 227 academic documents. Huang Yong et al. implemented automatic structure function recognition from section headers, section content, and paragraph content using CRF and SVM, achieving favorable results.

Deep learning, a branch of machine learning, has advanced rapidly in natural language processing. First proposed by G. E. Hinton in 2006, deep learning establishes hierarchical structures that simulate the human brain to extract features from external inputs, creating complex mappings from low-level to high-level semantics. R. Salakhutdinov et al. applied Deep Belief Networks and stacked autoencoders for document indexing and retrieval. X. Glorot et al. used deep learning for domain-adaptive sentiment classification, extracting meaningful feature representations from unlabeled online reviews, with experimental results showing that classifiers trained with high-level features significantly outperformed other methods. M. M. Rahman et al. introduced Recurrent Neural

Networks (RNN) into document structure understanding research with promising results.

Overall, rule-based methods require manual rule construction and are generally tailored to specific document types without guaranteed accuracy. Machine learning methods improve precision and efficiency over rule-based approaches but rely on manual feature extraction, learning only single-layer features from hierarchical structures. In contrast, deep learning can automatically complete data representation and feature extraction, forming more abstract deep representations through low-level feature combinations to enhance data interpretability, increasingly applied in text classification.

3 Multi-level Fusion Structure Function Recognition Model

3.1 Multi-level Structure of Academic Text

Academic text typically exhibits rigorous logical structure and standardized hierarchy, following the general scientific research process from problem formulation and methodology introduction to results discussion and conclusion. Sections with different purposes and functions constitute a complete academic article. This study focuses on structure function recognition of academic text bodies, dividing them into five categories based on logical structure and prior research: "Introduction," "Related Work," "Method," "Experiment," and "Conclusion."

An academic text body comprises multiple sections, each consisting of a section header and section content, with section content further containing varying numbers of paragraphs, as illustrated in [Figure 1: see original paper]. Structure function recognition essentially constitutes a content-based classification problem that can be approached at three levels: (1) section header-level recognition based on header text; (2) section content-level recognition using full section content to provide richer textual features; and (3) paragraph-level recognition that classifies all paragraphs within a section and determines the section' s structure function by majority voting.

3.2 Model Construction

Based on deep learning technology, this study fuses three levels—section headers, section content, and paragraphs—to recognize academic text structure functions. The overall framework, shown in [Figure 2: see original paper], comprises two modules: deep learning-based structure function classification and voting-based multi-level fusion. This model integrates features from different academic text levels, providing global structure function recognition from the entire text body perspective and cleverly employing ensemble learning through voting to fuse results from different levels. Compared with single-level recognition, this model offers richer application scenarios and superior performance.

3.2.1 Deep Learning-Based Structure Function Classification The deep learning-based classification module consists of five layers: input, word embedding, feature learning, Softmax, and output. The input layer comprises labeled training and test sets from section headers, section content, and paragraphs, with texts uniformly padded to equal length. The word embedding layer converts text into vector representations using word2vec, representing each word as a K-dimensional real vector that resolves sparsity while enabling similarity computation through cosine similarity or Euclidean distance.

The feature learning layer forms the core of the model. This study employs Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM), and CNN+LSTM models to learn from word vector representations and classify test texts. The Softmax layer normalizes outputs to calculate probabilities for each category, serving as the most common normalization function for multi-class classification. The output layer produces predicted categories and probability distributions.

(1) CNN. The CNN architecture, shown in [Figure 3: see original paper], comprises input, convolution, pooling, and output layers. The input layer is an n×d word vector matrix S, where n represents input text length (with zero-padding for shorter texts) and d represents word vector dimension (256 in this study). The convolution layer extracts high-level features using VALID Padding with stride 1. For a convolution kernel w, each step performs convolution within an h-height window to extract new feature c_i:

$$c_i = f(w * S_{i:i+h-1} + b)$$

where f is the ReLU activation function, b is bias, and h is window size. To comprehensively extract local features at different granularities, this study employs three kernel sizes (3, 4, and 5). Kernel w performs complete convolution across n-h+1 windows, generating feature vector C = [c_1, c_2, ···, c_{n-h+1}].

Max pooling extracts the most useful text fragments by selecting the maximum value $\hat{c} = max(C)$, identifying the most influential factors for classification while fixing the number of neurons in fully connected layers. Finally, the output layer connects all local optimal features to output nodes through Softmax for category prediction, with parameters updated via backpropagation.

(2) LSTM. LSTM, an improved RNN model proposed by S. Hochreiter and J. Schmidhuber in 1997 to address gradient vanishing, replaces hidden units with memory cells comprising a cell, input gate, forget gate, and output gate. The cell state records information, with gates controlling modification and propagation. At time t, LSTM updates as follows:

$$i_t = sigmoid(W_i * [h_{t-1}, x_t] + b_i)$$

$$f_t = sigmoid(W_f * [h_{t-1}, x_t] + b_f)$$

$$o_t = sigmoid(W_o * [h_{t-1}, x_t] + b_o)$$

$$C_t = f_t * C_{t-1} + i_t * tanh(W_C * [h_{t-1}, x_t] + b_C)$$

$$h_t = o_t * tanh(C_t)$$

where i_t, f_t, o_t, and C_t represent input gate, forget gate, output gate, and cell state at time t; x_t is input vector; h_t is hidden state; and W and b terms are weight matrices and bias vectors. This gate structure and memory unit enable selective information retention and forgetting, avoiding gradient vanishing while capturing long-term dependencies.

(3) CNN+LSTM. CNN excels at extracting local text features through sliding windows but is position-insensitive and lacks sequence modeling capability. LSTM effectively captures sequential relationships but is biased toward later words. This study combines CNN and LSTM by feeding LSTM outputs into CNN convolution layers, integrating LSTM hidden states with CNN pooling results, and producing final categories through fully connected output layers.

3.2.2 Voting-Based Multi-level Fusion Voting is an ensemble learning strategy for classification that selects the majority class across all algorithms. Compared with single classifiers, ensemble methods offer better generalization and higher quality results. Following majority voting rules, this study fuses results using:

$$R(x) = Vote(H(x), P(x), S(x))$$

where for each section x, H, P, and S represent classification results from section headers, paragraphs, and section content, respectively, and R is the fused result —the category receiving the most votes.

4 Experiments and Results Analysis

4.1 Experimental Environment

All experiments were conducted in the environment shown in .

4.2 Dataset

The experimental data comprises computer linguistics journal papers from ScienceDirect (2000-2014). We randomly selected 4,000 papers from 101 journals as our dataset. Using stratified sampling by journal name, we divided the dataset into 101 strata and randomly selected 3,500 papers for training and 500 for testing. Each paper contains three levels of text: section headers (21,526 instances), section content (21,526 instances), and paragraphs (184,433 instances).

4.3 Evaluation Metrics

We employ Precision (P), Recall (R), and F1-score (F1) for evaluation:

$$P = \frac{\text{Correctly identified structure functions}}{\text{Identified structure functions}}$$

$$R = \frac{\text{Correctly identified structure functions}}{\text{Actual structure functions}}$$

$$F1 = \frac{2 * P * R}{P + R}$$

Overall metrics are weighted arithmetic averages across categories.

4.4 Experimental Results and Analysis

We applied CNN, LSTM, and CNN+LSTM models on TensorFlow to identify structure functions at three levels. CNN used kernel heights of 3, 4, and 5; LSTM employed a unidirectional two-layer model; optimization used Adam with ReLU activation; word vector dimension was 256. Each model trained for 200 epochs (with early stopping if performance plateaued). Optimal hyperparameters were established through single-factor experiments, as shown in . Training time comparisons appear in [Figure 4: see original paper], showing that paragraph-level training required the longest time, followed by section content, then headers—demonstrating that training time correlates with data volume and complexity. LSTM training took longer than CNN, while CNN+LSTM showed no consistent time pattern.

Header-Level Results (Table 3): All models achieved over 85% overall precision, with CNN performing best (86% precision, 85% recall, 85% F1). "Introduction" achieved perfect scores across all models, while "Conclusion" reached 98% precision. "Method" performed worst (72% precision) due to diverse header expressions and limited corpus size.

Paragraph-Level Results (Table 4): Overall precision exceeded 69%, with LSTM performing best (73% precision, 70% recall, 68% F1). "Conclusion"

achieved 96% precision but only 53% recall. "Method" showed lowest precision, consistent with header-level results. "Related Work" exhibited low recall, indicating frequent misclassification.

Section Content-Level Results (Table 5): Performance varied significantly across models, with CNN achieving best results (75% precision, 74% recall, 73% F1). "Introduction" and "Conclusion" performed best (86% and 85% precision, respectively). "Related Work" again showed lowest recall.

Across levels, header-level recognition performed best, followed by section content, then paragraphs. Headers contain explicit functional words, making features more salient. Paragraph and section content, being longer texts, increase difficulty for feature extraction. Model performance differences were smallest at header level and largest at section content level, with CNN excelling at short and long texts while LSTM performed best at paragraph level.

Voting Results (Table 6): Voting improved or maintained performance across all levels. Integrated voting achieved 86% overall precision—1% lower than header voting but 16.22% and 14.67% higher than paragraph and content voting, respectively. While header features contributed most to fusion results, paragraph and content features provided valuable supplements, particularly for "Related Work" and "Experiment" categories (5.62% and 6.10% precision improvements). This demonstrates that comprehensive multi-level fusion achieves optimal recognition and generalization, remaining feasible even when headers are absent.

4.5 Comparison Analysis

SVM, a traditional machine learning algorithm widely used in text classification, served as the baseline. Using Python's sklearn with term frequency features (following prior work), SVM results appear in . Deep learning models outperformed SVM by 7.50%, 25.86%, and 20.97% at header, paragraph, and content levels, respectively. SVM' s reliance on small samples and manual feature extraction limits performance on academic texts with high inter-category similarity, while deep neural networks effectively leverage inter-sentence and inter-word features, demonstrating greater advantages in multi-class academic text classification.

4.6 Error Analysis

Analyzing integrated voting misclassifications () reveals that "Related Work" is most frequently misclassified as "Method," followed by "Introduction," due to high textual similarity between method descriptions in these sections and the fact that some papers merge "Related Work" into "Introduction" or "Method," reducing its proportion in the corpus. "Method" and "Experiment" show the highest mutual misclassification rates, indicating structural similarity consistent with prior research. "Method" receives the most misclassifications from other categories, suggesting method descriptions are distributed throughout papers.

Based on these findings, we propose two improvements: (1) Incorporate vocabulary features that differentiate categories, exploring attention mechanisms to enhance model comprehension and discrimination; (2) Increase corpus size and balance category distributions, as neural networks perform better on large-scale data. Future work will combine manual and automatic annotation to construct multi-domain, large-scale structure function datasets.

5 Conclusion

This study innovatively introduces deep learning to academic text structure function recognition, employing CNN, LSTM, and CNN+LSTM models across three hierarchical levels and fusing results via voting. Header-level recognition achieves best performance, followed by section content, then paragraphs. CNN demonstrates superior overall performance compared to LSTM, while CNN+LSTM fusion shows no improvement over individual models. Compared with SVM, deep learning exhibits superior performance. Voting-based fusion enhances all levels, with integrated voting achieving 86% precision, 84% recall, and 84% F1. The proposed multi-level fusion model proves efficient, practical, and capable of incremental and transfer learning.

In the academic big data environment, fine-grained and deep semantic understanding of academic texts is increasingly important. Structure function-based understanding can advance related research, such as analyzing vocabulary functions and semantic roles at finer granularity, fusing text structure functions with citation functions for citation recommendation and knowledge discovery, and exploring structure function-based paper evaluation to support content-based assessment.

References

[1] Xia F, Wang W, Bekel T M, et al. Big scholarly data: a survey [J]. IEEE transactions on Big Data, 2017, 3(1): 18-35.
[2] Hug S E, Brandle M P. The coverage of microsoft academic: analyzing the publication output of a university [J]. Journal of informetrics, 2017, 11(3): 1551-1571.
[3] Ribaud P H D, Falquet G. Extracting discourse elements [C]//Proceedings of the 3rd IEEE/ACM international conference on Big Data computing, applications and technologies. Shanghai: IEEE/ACM, 2017: 63-73.
[4] Rahman M M, Finin T. Deep understanding of a document' s structure [C]//Proceedings of the 3rd IEEE/ACM international conference on Big Data computing, applications and technologies. Shanghai: IEEE/ACM, 2017: 63-73.
[5] Alzahrani S, Palade V, Salim N, et al. Using structural information and citation evidence to detect significant plagiarism cases in scientific publications [J]. Journal of the American Society for Information Science and Technology, 2012, 63(2): 286-312.
[6] Khan S, Liu X F, Shakil K A, et al. A survey on scholarly data: from big

data perspective [J]. Information processing and management, 2017, 53(4): 923-944.
[7] Lu W, Huang Y, Cheng Q K. Structure function recognition of academic text: functional framework and recognition based on section headers [J]. Journal of the China Society for Scientific and Technical Information, 2014, 33(9): 979-987.
[8] Luong M T, Nguyen T D, Kan M Y. Logical structure recovery in scholarly articles with rich document features [J]. International journal of digital library systems, 2010, 1(4): 1-23.
[9] Sollaci L B, Pereira M G. The introduction, methods, results, and discussion (IMRAD) structure: a fifty-year survey [J]. Journal of the medical library association, 2014, 92(3): 364-367.
[10] Fang L, Li X, Huang Y, et al. Structure function recognition of academic text: application in automatic keyword extraction [J]. Journal of the China Society for Scientific and Technical Information, 2017, 36(6): 599-608.
[11] Hu Z G, Chen C M, Liu Z Y. Where are citations located in the body of scientific articles? a study of the distributions of citation locations [J]. Journal of informetrics, 2013, 7(4): 887-896.
[12] Ding Y, Liu X Z, Guo C, et al. The distribution of references across texts: some implications for citation analysis [J]. Journal of informetrics, 2013, 7(3): 583-592.
[13] Tuarob S, Mitra P, Giles C L. A hybrid approach to discover semantic hierarchical sections in scholarly documents [C]//Proceedings of the 13th international conference on document analysis and recognition. Nancy: IAPR, 2015: 1081-1085.
[14] Huang Y, Lu W, Cheng Q K. Structure function recognition of academic text: recognition based on section content [J]. Journal of the China Society for Scientific and Technical Information, 2016, 35(3): 293-300.
[15] Huang Y, Lu W, Cheng Q K, et al. Structure function recognition of academic text: recognition based on paragraphs [J]. Journal of the China Society for Scientific and Technical Information, 2016, 35(5): 530-538.
[16] Xi X F, Zhou G D. Deep learning for natural language processing [J]. Acta automatica sinica, 2016, 42(10): 1445-1465.
[17] Mao S, Rosenfeld A, Kanungo T. Document structure analysis algorithms: a literature survey [J]. Proc spie electronic imaging, 2003(5010): 197-207.
[18] Kim J, Le D X, Thom G R. Automated labeling in document images [C]//Proceedings of the SPIE conference on document recognition and retrieval VIII. San Jose: SPIE, 2000: 111-122.
[19] Constantin A, Pettifer S, Voronkov A. PDFX: fully-automated PDF-to-XML conversion of scientific literature [C]//Proceedings of the ACM symposium on document engineering. Florence: ACM, 2013: 177-180.
[20] Hinton G E, Salakhutdinov Y R. Reducing the dimensionality of data with neural networks [J]. Science, 2006, 313(5786): 504-507.
[21] Salakhutdinov R, Hinton G E. Semantic hashing [J]. International journal of approximate reasoning, 2009, 50(7): 969-978.
[22] Glorot X, Bordes A, Bengio Y. Domain adaptation for large-scale senti-

ment classification: a deep learning approach [C]//Proceedings of the 28th international conference on machine learning. Washington: Omnipress, 2011: 513-520.
[23] Zhang L. Grasping the structure of journal articles: utilizing the functions of information units [J]. Journal of the Association for Information Science and Technology, 2012, 63(3): 469-480.
[24] Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural Computation, 1997, 9(8): 1735-1780.
[25] Graves A. Supervised sequence labeling with recurrent neural networks [D]. München: Technische Universität München, 2008.
[26] Zhang C Z. Research on ensemble learning-based automatic indexing method [J]. Journal of the China Society for Scientific and Technical Information, 2010, 29(1): 3-8.

Author Contributions

Wang Jiamin: Experimental analysis, data processing, paper writing;
Lu Wei: Research framework design, paper revision;
Liu Jiawei: Experimental analysis, paper revision;
Cheng Qikai: Framework development, paper revision.

Note: Figure translations are in progress. See original paper for figures.