Postprint: Construction of Text Knowledge Networks Integrating Discourse Structure

Authors: Liu Yao, Zhang Yue, Ye Lu

Date: 2023-04-01T16:02:58+00:00

Abstract

[Purpose/Significance] Text vectorization is an essential preprocessing step in text mining, information retrieval, sentiment analysis, and related fields. Enabling node vectors to encapsulate rich and effective semantic and structural information represents an urgent challenge requiring resolution. [Method/Process] We first analyze the characteristics of science and technology policy texts. Based on a classification system for concepts and inter-concept relationships, we respectively employ BiLSTM-CRF algorithms and SVM to achieve automatic indexing of concepts and conceptual relationships. Feature engineering incorporates both fundamental features and syntactic-semantic features, yielding significant improvements in recognition accuracy and efficiency. Furthermore, we propose a method for constructing concept knowledge networks that integrate inferential knowledge, as well as an advanced knowledge network construction approach that fuses discourse structure. [Results/Conclusion] Based on this knowledge network model, we implement a network representation learning model capable of fusing node semantics, topological structure, and category label information, which enables comprehensive mining and representation of textual semantic and structural information. The effectiveness of the proposed method is verified through visualization and experimental validation.

Full Text

Construction of Text Knowledge Network Integrating Discourse Structure

Liu Yao[1], Zhang Yue[2], Ye Lu[3] [1]Institute of Scientific and Technical Information of China, Beijing 100038 [2]Michigan State University, East Lansing 489132 [3]School of Software & Microelectronics, Peking University, Beijing 100871

Abstract: [Purpose/Significance] Text vectorization is a necessary preprocessing step in fields such as text mining, information retrieval, and sentiment

analysis. A critical challenge is how to make node vectors contain rich and effective semantic and structural information. [Method/Process] This paper first analyzes the characteristics of science and technology policy texts, and respectively employs BiLSTM-CRF algorithms and SVM to automatically index concepts and conceptual relationships according to classification systems for both concepts and inter-concept relationships. The feature engineering integrates both basic features and syntactic-semantic features, achieving significant improvements in recognition accuracy and efficiency. The paper also proposes a conceptual knowledge network construction method that incorporates reasoning knowledge and further integrates discourse structure. [Result/Conclusion] Based on this knowledge network model, we implement a network representation learning model that can fuse node semantics, topological structure, and category label information, enabling full mining and representation of textual semantic and structural information. The effectiveness of the proposed method is verified through visualization and experiments.

Keywords: Named Entity Recognition; Relationship Extraction; Neural Network; Representation Learning; Discourse Structure **Classification Number:** TP182 **DOI:** 10.13266/j.issn.0252-3116.2021.21.019

---

1 Related Work

In text knowledge networks, nodes consist of various entities while edges represent grammatical relationships between entities. Therefore, constructing text knowledge networks requires entity extraction and relationship recognition as foundational steps.

1.1 Entity Extraction

Named entity recognition (NER) is a crucial research task in natural language processing and serves as a prerequisite for information extraction, coreference resolution, question answering systems, and topic modeling. Current NER techniques primarily fall into three categories: rule and dictionary-based methods, statistical machine learning methods, and deep learning methods.

Rule-based NER methods typically rely on dictionaries and knowledge bases, employing manually constructed rule templates by linguistic experts to identify entity types through pattern matching and regular expressions. Each rule has an associated weight, with higher-weight rules taking precedence when conflicts occur. Commonly used manual features include keywords, headwords, and indicator words. A representative system is the ANNIE information extraction system in the GATE project, which relies on handcrafted rules to build entity libraries and performs information extraction for entire documents based on entity extraction rule definitions. When rules are sufficiently comprehensive, rule-based methods often outperform other approaches. However, linguistic phenomena are highly variable in reality, making rule creation extremely

time-consuming and prone to conflicts, resulting in low feasibility. Moreover, this approach requires extensive domain knowledge and dictionaries, leading to poor system portability.

With the rise of machine learning in NLP, NER tasks have gradually shifted toward statistical machine learning methods. These approaches typically treat NER as a classification problem with two main strategies: first identifying all entity boundaries in text, then classifying the entities, or performing sequential labeling by assigning candidate category tags to each word corresponding to entity positions (e.g., IOB tagging schemes) and finally recognizing entities through classifiers. Classic machine learning methods include Hidden Markov Models (HMM), Maximum Entropy (ME), Support Vector Machines (SVM), and Conditional Random Fields (CRF). Among these, ME offers good generality but incurs high computational costs due to normalization requirements. CRF provides a flexible, globally optimal labeling framework but converges slowly and requires long training times. Generally, ME and SVM achieve higher accuracy than HMM, but HMM's use of the Viterbi algorithm enables faster training, making it more suitable for real-time applications such as short-text NER.

In 2011, Collobert proposed a neural network-based NER model. Deep learning methods have remained popular as they eliminate the need for cumbersome feature engineering. Existing neural network models can be categorized by input granularity into word-level, sentence-level, and combined word-sentence level models. Word-level models use word vectors as inputs. Collobert achieved 89.59% F1 on the CoNLL2003 dataset by feeding word vectors into a CNN+CRF model. Huang et al. proposed an LSTM+CRF model using word vectors as input, achieving 85.19% F1 on CoNLL2003. Sentence-level models represent entire sentences as inputs, incorporating position features to distinguish each character. Pham and Le-Hong used sentence-level representations in a Bi-LSTM+softmax model, achieving 80.23% F1 for Vietnamese NER. Combined word-sentence level models use both word vectors and character-level convolutions. Ma and Hovy fed such combined inputs into a Bi-LSTM+softmax model, reaching 91.21% F1 on CoNLL2003. Results show that NN and CNN perform similarly, but adding CRF to sentence-level models yields significant improvements. Since Bi-LSTM captures sequence information, it outperforms CRF models with rich features and has become the mainstream approach for deep learning-based NER. Overall, deep learning methods achieve excellent results using only word and character vectors without complex feature engineering, with performance further improving when high-quality dictionary features are incorporated. Recent NER advances have introduced attention mechanisms, graph neural networks, transfer learning, and distant supervision techniques on top of neural network structures.

1.2 Relationship Extraction

Relationship extraction comprises two subtasks: detecting whether a sentence contains entity pairs and determining the relationship between them. Machine

learning-based relationship extraction methods are primarily categorized into supervised, semi-supervised, and unsupervised approaches based on manual involvement.

Supervised methods include feature-based and kernel-based approaches. Feature-based methods treat relationship extraction as a binary classification problem, using manually annotated corpora to obtain positive and negative examples, deriving feature sets through lexical, syntactic, and semantic analysis, and training classification models. Common models include traditional machine learning models like CRF, SVM, and maximum entropy classifiers, as well as recent deep learning models such as shallow neural networks and CNNs. Traditional models require extensive manual feature design and selection, while deep learning adopts an end-to-end approach needing only pre-trained word vectors with minimal manual intervention.

Semi-supervised methods primarily employ Bootstrapping, starting with a small set of manually constructed relation instances as seed samples, then using pattern training or learning to summarize rules for discovering new relation instances until a large-scale collection is obtained. The DIPRE system constructed author-book relationships using this approach, leveraging a small set of entity relation pairs as seeds to retrieve documents or sentences containing both entities, using them as annotated samples to build and adjust patterns, and iteratively adding newly annotated data to the seed collection.

Unsupervised methods mostly use pattern clustering. Hasegawa et al. first proposed unsupervised relation extraction by clustering texts containing named entity pairs and using the most frequent words in clusters as relation descriptors, demonstrating effectiveness on large-scale news corpora. Piasecki introduced WordNet to improve similarity calculation for relation extraction template clustering. Unsupervised methods generally require large corpora to mine relation pattern sets but struggle to obtain high-confidence patterns and find it difficult to describe relation names.

1.3 Network Representation Learning

Network representation learning, also known as graph representation learning, has broad applications including node classification, clustering, link prediction, community detection, and recommendation systems. Structure-based network representation learning mainly includes matrix eigenvector methods, matrix factorization methods, and neural network methods.

Perozzi et al. proposed DeepWalk, experimentally verifying that node random walk sequences follow power-law distributions similar to words in documents, thereby applying the Word2vec method to network node random walk sequences. DeepWalk uses the skip-gram model to probabilistically model nodes in local windows of random walk sequences, training model parameters by maximizing random walk likelihood. This method relies only on local random walk information, solving the high computational time and space

problems of matrix eigenvector methods that require storing entire adjacency matrices. Node2vec further extended DeepWalk's random walk strategy by introducing parameters p and q to incorporate depth-first and breadth-first search, reflecting different levels of relationships between nodes.

Unlike shallow neural networks, deep neural network models can model nonlinear relationships between nodes. For example, SDNE uses the Laplace matrix to model first-order similarity between nodes, then employs unsupervised deep autoencoders to model second-order similarity, using the intermediate representation as node embeddings. In real-world scenarios, network nodes often contain rich external information. In social networks, besides friend relationships, users have abundant text information such as posts. Traditional network representation learning primarily captures topological structure, while external information can complement topology to improve representation quality. The TADW algorithm incorporates text content into network representation learning, using matrix factorization to decompose the relation matrix into text feature vectors and two parameter matrices.

Some researchers have proposed Trans-series models that incorporate reasoning structures from knowledge graphs into network representation learning. For instance, Tu et al.'s TransNet model combines the TransE relational reasoning model with network topology representation through autoencoders, demonstrating significant effectiveness on social relation extraction tasks. Although network representation learning has achieved rich results, it still faces major challenges, such as overcoming storage and training efficiency issues when representing large-scale networks with hundreds of millions of nodes and integrating external information.

2 Text Knowledge Network Construction Model

The overall architecture of the text knowledge network algorithm model is shown in Figure 1 [Figure 1: see original paper]. First, concepts and relationships in science and technology policy texts are annotated. A BiLSTM-CRF deep learning model performs concept annotation, and concept pair relationship features serve as input to an SVM-based active learning classifier to predict relationship labels for unlabeled concept pairs. This yields concepts and inter-concept relationships in each sentence of every policy text, stored in JSON format.

Concept relationships are then used to construct both a science and technology policy knowledge network and a discourse-structured knowledge network. Based on the knowledge network model, we employ network representation learning techniques to represent nodes. For concept representation, we first use a network representation model fusing node semantics, topology, and label information, then propose an improved method combining knowledge reasoning models. For chapter node representation, we propose a discourse node representation method based on Doc2vec.

2.1 Concept Relationship Annotation Model

The core task of entity extraction is named entity recognition—extracting person names, locations, organizations, technologies, etc. Relationship extraction extracts semantic relations between entity pairs from sentences, which is crucial for natural language understanding, information retrieval, and automatic knowledge graph construction, enabling extraction of structured data from large-scale unstructured texts to improve information processing efficiency.

2.1.1 BiLSTM-CRF Concept Annotation To address limitations of traditional text analysis methods on science and technology policy texts, BiLSTM (Bidirectional Long Short-term Memory) effectively captures output sequences through contextual features but cannot express strong dependencies between output labels in sequence labeling problems. We add a CRF model to the final layer of the BiLSTM neural network to effectively solve this issue.

We systematically analyze concept categories in science and technology policy texts. While traditional NER identifies three entity types (person, location, organization), policy texts involve broader categories. We classify concepts in science and technology briefings using concept dictionaries, rule extraction, and manual annotation. After accumulating sufficient vocabulary, we summarize the following conceptual taxonomy (selecting the most frequent category if a concept belongs to multiple categories): Organization: organization names; Location: typically includes country names, place names, or general terms like "BRICS countries"; Policy: issued science and technology policies; Money: funds, investments, and capital involved in policies or technologies; Technology: technical terminology; Field: domain definitions; Energy: energy-related terms; Facility: various equipment; People: general or specific references to persons; System: systems, frameworks, or platforms; Element: objects with containment relationships with other categories; Attribute: characteristics describing a technology field; Service: services provided by national policies; Product: product descriptions; Project: proposed projects, methods, or solutions.

We employ Bidirectional LSTM, which contains two LSTM layers: a forward sequence preserving historical information and a backward sequence capturing future information, both passing temporal information to the output layer. This enables BiLSTM to solve long-term dependency problems while obtaining contextual information for sequence labeling. However, BiLSTM cannot express strong label dependencies. We segment Chinese characters as input units to the BiLSTM neural network model and use the IOB annotation method to distinguish concept boundaries in each sentence, where "B" tags the first character of a concept, "I" tags internal characters, and "O" tags non-concept characters. In our concept extraction task, we add a CRF layer to the final BiLSTM layer to effectively address this issue. The BiLSTM-CRF model structure is shown in Figure 2 [Figure 2: see original paper].

BiLSTM output sequence features combine character vectors with contextual semantic features. We use a Softmax function to map hidden layer outputs to probability distributions over label sets, obtaining a probability distribution matrix for each character's corresponding label. Finally, the CRF layer determines the highest-probability valid label sequence across all possible sequences, assigning final labels to each character.

2.1.2 SVM Active Learning Relationship Annotation Relationship annotation aims to automatically extract relationships between concepts from sentences, a key technology for knowledge structuring. We transform relationship extraction into a classification task, establishing a relationship taxonomy based on textual content features. First, lexical and syntactic analysis tools process partial corpora to extract relevant features between concepts as SVM classifier inputs, then employ active learning to annotate relationships for unlabeled concept pairs. The relationship annotation framework is shown in Figure 3 [Figure 3: see original paper].

Our research corpus exhibits diverse concepts and relationships. Since concepts are primarily nouns appearing as subjects/objects or in subordinate clauses, relationships mainly analyze core predicates linking concepts, focusing on "subject+predicate+object" or "subject+predicate+subordinate clause (subject+predicate+object)" sentence structures. We preset five relationship types: Forward: promoting relationships; Mixation: fusion relationships; Backward: hindering relationships; Inclusion: containment relationships; Likelihood: synonymous relationships.

We employ active learning for relationship classification, first selecting a small sample from candidate sets based on prior knowledge to construct an initial training set. Features include basic and syntactic-semantic categories. Basic features derive from lexical analysis, with their effectiveness validated by previous researchers. Our basic entity relationship features include:

- **Concept categories**: The 15 concept categories defined in Section 2.1.1, with two concept categories concatenated by "-"
- **Adjacent words**: Words preceding and following each concept, using "None" if no such words exist
- **Part-of-speech tags between concepts**: POS tags of all words from one concept to another
- **Contextual environment between concepts**: Including words between the two concept terms

Beyond basic features, we incorporate syntactic-semantic features including dependency parsing and semantic role analysis. Dependency structure reveals relationships between components, with the core predicate dominating other components. Since concept phrases are parts of dependency structures, inter-component dependencies reflect concept relationships. Figure 4 [Figure 4: see original paper] shows the dependency parse for "Nanotechnology powers are

vigorously promoting the integration of nanotechnology and information technology strategic emerging fields," where "ATT" indicates modifier-head, "ADV" adverbial, "HED" head, "COO" coordination, "SBV" subject-verb, "LAD" left adjunct, "RAD" right adjunct, "VOB" verb-object, and "WP" punctuation.

Semantic role labeling is a shallow semantic analysis technique centered on sentence predicates, analyzing relationships between components and predicates. It is an important intermediate step in natural language understanding. For example, in "South Korea's Ministry of Knowledge Economy specially issued the 'Nanotechnology Convergence Promotion Strategy'," semantic role labeling results (Figure 5 [Figure 5: see original paper]) primarily contain three parts: A0 (agent), A1 (patient), and ADV (adjunct marker), reflecting semantic relationships between concepts.

Using the sentence "Nanotechnology powers are vigorously promoting the integration of nanotechnology and information technology strategic emerging fields" with concepts "nanotechnology powers" and "nanotechnology," we demonstrate feature extraction using the LTP natural language processing tool. The word segmentation yields: "nanotechnology powers / vigorously / promoting / nanotechnology / and / information technology / strategic / emerging fields / of / integration." Feature extraction results are shown in Table 1 .

Support Vector Machine (SVM) is a machine learning algorithm for classification and regression based on structural risk minimization, learning classifiers by compressing training data into a support vector set. The active learning algorithm using SVM is detailed in Table 2 .

The key to active learning is the sampling strategy, which affects classifier performance. To identify an appropriate strategy, we simulated manual annotation using labeled training data during early annotation stages to evaluate three sampling strategies: Least Confidence (LC), Margin Selection (MS), and Random Selection (RS). LC selects k samples with minimum confidence; MS selects k samples with smallest difference between two highest class probabilities; RS randomly selects k samples. As shown in Figure 6 [Figure 6: see original paper], MS performs best.

2.2 Knowledge Network Construction and Network Representation Model

Knowledge networks lack a precise definition. According to literature [25], a knowledge network is a collection of knowledge, information, and inter-knowledge relationships. Nodes represent knowledge storage units (books, papers, patents, text fragments, or words at different granularities), while edges represent relationships between knowledge units (citation relationships in citation networks, co-occurrence in co-occurrence networks). The extracted concept-concept relationships can form such knowledge networks, where nodes are concept terms and edges are semantic relationships between concepts. We represent knowledge networks as G = (V, E, D, L), where V = {v_1, v_2, …,

v_N} are nodes (concepts), e_{i,j} = (v_i, v_j) E are edges (relationships), D = {w_1, w_2, ..., w_N} is text information for each node, and L = {l_v, l_r} is the set of concept and relationship labels.

2.2.1 TriDNR Network Representation Model with Reasoning Knowledge Network representation learning aims to learn low-dimensional latent representations of network nodes, capturing node context features and community information from network topology. The TriDNR network representation learning model [26] obtains topological structure representations through DeepWalk. As shown in Figure 7 [Figure 7: see original paper], the framework considers an input network with node set V, where nodes v_1, v_2, v_3, v_4, v_7 each associate with a word set W (w_2, w_3, w_5 being word sequences of lengths 2, 3, 5), and some nodes have different label attribute sets C (c_1 being node v_1's category label). The model simultaneously learns relationships between nodes, between nodes and words, and between labels and words.

TriDNR consists of two skip-gram neural network layers: the upper layer models node topology, while the lower layer models text content and labels. The skip-gram model obtains each node's representation, similar to word vectors. Unlike traditional network representations, DeepWalk uses random walks instead of adjacency matrices, solving high computational space and time problems. The upper structure employs DeepWalk to map random walk strategies to each node representation, which after random shuffling is passed to the lower structure.

The lower structure's objective function is:

$$L = \sum_{i=1}^{|L|} \log P(w_{-b:w_b}|c_i) + \sum_{i=1}^{|N|} \log P(w_{-b:w_b}|v_i)$$

This shows node content and labels are similar to Doc2vec. Overall, we combine node topology, text content, and label information through DeepWalk and Doc2vec. The complete model's objective function maximizes the likelihood estimation of Equation (2):

$$L = (1-\alpha) \sum_{i=1}^{N} \sum_{-b \le j \le b, j \neq 0} \log P(v_{i+j}|v_i) + \sum_{i=1}^{N} \sum_{-b \le j \le b} \log P(w_j|v_i) + \sum_{i=1}^{|L|} \sum_{-b \le j \le b} \log P(w_j|c_i)$$

where α balances topology, text content, and label information, and b is the window size. The first term calculates the probability of surrounding nodes given a node, obtained via softmax:

$$P(v_{i+j}|v_i) = \frac{\exp(v_{v_i}^T v'_{v_{i+j}})}{\sum_{v=1}^{V} \exp(v_{v_i}^T v'_v)}$$

where v_v and v'_v are node v's input and output vectors. Given node v, word probability is:

$$P(w_j|v_i) = \frac{\exp(v_{v_i}^T v'_{w_j})}{\sum_{w=1}^{W} \exp(v_{v_i}^T v'_w)}$$

Label probability is similarly:

$$P(w_j|c_i) = \frac{\exp(v_{c_i}^T v'_{w_j})}{\sum_{w=1}^{W} \exp(v_{c_i}^T v'_w)}$$

Equations (4) and (5) jointly influence word vector representation v'_{w_j}, which affects input v_i through backpropagation, achieving fusion of topology, content, and label information.

However, TriDNR only considers node topology, not edge label information. We draw inspiration from the TransE knowledge representation learning model in the Trans series to incorporate five edge label categories (Backward, Forward, Mixation, Likelihood, Inclusion) that reveal reasoning relationships between concepts. Knowledge representation learning can obtain both node and edge representations, typically applied to entity linking tasks. Since our research only focuses on mapping relationship labels to node representations, we simply average node representations obtained from topology, text semantics, and node labels, using the resulting concept node vectors as knowledge network embeddings. The TransE algorithm is detailed in Table 3 .

2.2.2 Discourse-Integrated Network Representation Learning Model
Extracted concepts and relationships belong to various chapters, with each chapter containing a knowledge network subgraph. When discourse structure and knowledge networks are combined, each policy text forms a tree-structured upper layer and a concept-relationship network lower layer. The article title serves as the Root node, first-level headings as the first tree layer, second-level headings under each first-level heading as the second layer, and so on (typically not exceeding three layers). Each subheading contains corresponding concepts and relationships as the bottom layer, with concepts connecting across chapters to form a network data structure.

Concepts connect through directed edges like sentences, while chapter nodes can be viewed as documents containing multiple sentences. Based on this analysis, we categorize discourse nodes into two types: leaf nodes at the bottom layer directly connected to concepts, and other upper-layer nodes. For leaf nodes, we treat connected concepts as words and random walk paths formed by concept connections as sentences, using the Doc2vec algorithm to compute discourse node representations. For upper-layer nodes, we calculate vectors by averaging same-layer nodes recursively until obtaining the root node representation.

The Doc2vec method for discourse leaf node representation is shown in Figure 8 [Figure 8: see original paper], where w represents concept nodes, v represents

concept node embeddings, and paragraph matrix represents chapter node embeddings. Chapter nodes connect with concept vectors through concatenation or simple addition to predict subsequent concepts, building a shallow neural network model whose training yields leaf node representations. This allows different chapters to obtain different vector representations based on their connected concepts, while identical concepts share the same representation across chapters. As Doc2vec is unsupervised, it can train on unlabeled data, enabling fast and efficient chapter node representation.

3 Experimental Setup and Results Analysis

3.1 Concept Relationship Annotation Experiments

For concept recognition, we built a BiLSTM+CRF deep learning model using word vectors as input. For relationship recognition, to mine semantic knowledge between concepts at a high level, we designed four basic features (concept categories, adjacent words, inter-concept POS tags, contextual environment) and two semantic features (dependency parsing and semantic role labeling), training an SVM classifier to validate feature effectiveness.

3.1.1 BiLSTM-CRF Concept Extraction Experiment Dataset: The dataset consists of nearly 1,000 science and technology reference documents publicly released by the Ministry of Science and Technology over 20 years. Mixed by year and manually divided into 10 parts, one part was annotated using rules supplemented with manual correction, containing 4,340 sentences and 4,790 unique concepts. After BIO tagging at character level, there are nearly 230,000 character labels, split 8:1:1 for training, testing, and validation.

Results and Analysis: We transformed concept extraction into a sequence labeling problem. To validate our method, we compared traditional CRF, BiLSTM, and BiLSTM-CRF approaches. With 15 concept categories, final results are averaged across all classes (Table 4).

Table 4: Concept Extraction Results Comparison | Method | Precision | Recall | F1 | |———|————|———|—-| | TextRank + Sentence Pattern | 32.34 | 31.05 | 31.68 | | BiLSTM | 55.67 | 50.37 | 53.42 | | BiLSTM+CRF | 63.97 | 52.02 | 57.38 | | BiLSTM-CRF | 70.89 | 63.59 | 67.32 |

BiLSTM outperforms traditional machine learning methods, and adding a CRF layer to BiLSTM further improves performance. Organization category recognition achieves the best result at 80.25% precision, while Attribute and Service categories perform worse due to their low proportion in the training dataset.

3.1.2 SVM Concept Relationship Recognition Experiment Dataset: We manually annotated 100 documents with 1,980 relationships (five types: Forward, Mixation, Backward, Inclusion, and Unrelated) as initial SVM training samples, with 80% for training and 20% for testing. The unlabeled candidate set

from the remaining 900 documents contains approximately 22,500 relationships. We use active learning, selecting 200 most influential samples per iteration for prediction, adding correctly classified samples back to the training set. Results are shown in Figure 9 [Figure 9: see original paper] and Table 5 .

Table 5: SVM Relationship Extraction Results | Features | Precision | Recall | F1 | |———-|———|———|—-| | Basic Features | 60.49 | 47.89 | 53.31 | | Basic + Syntactic-Semantic Features | 67.55 | 63.29 | 65.42 |

Basic features extract word-level relationships between entities but lack sentence-level grammatical features. However, intra-sentence relationships exhibit strong organizational associations that aid deep semantic mining. Results show that adding syntactic-semantic features improves extraction performance, validating their effectiveness.

3.2 Knowledge Network Construction and Network Representation Experiments

We improved the TriDNR model to incorporate network topology, node content, node labels, and inter-node reasoning information, addressing incomplete node representation. For discourse-integrated knowledge networks, we improved upon simple vector addition by using Doc2vec for discourse nodes, then applied TextRank to rank chapter node importance, validating the effectiveness of our generated discourse node vectors.

3.2.1 TriDNR with Reasoning Knowledge Experiment Dataset: After training the BiLSTM-CRF model, we sequentially selected one of the remaining nine parts for concept prediction and manual correction. We annotated concepts in nearly 45,000 sentences across 1,000 documents, obtaining approximately 35,000 unique concepts and 28,000+ relationships to form a large concept network.

Results and Analysis: To validate our method, we compared it with other concept node representation methods. For fairness, all methods use 300-dimensional representations, with parameter $p > 1$ to control depth-first search breadth. Table 6 compares different methods.

Table 6: Comparison of Concept Node Representation Methods | % of Labeled Nodes | DeepWalk | Doc2vec | Node2vec | TriDNR | Our Method | |———————-|———-|———|———-|———|————| | 10 | 0.243 | 0.275 | 0.312 | 0.476 | 0.512 | | 30 | 0.315 | 0.361 | 0.386 | 0.553 | 0.593 | | 50 | 0.394 | 0.428 | 0.443 | 0.618 | 0.649 | | 70 | 0.431 | 0.452 | 0.492 | 0.642 | 0.683 |

Methods considering only topology or text content achieve the lowest F1, especially with small training sets. Combining both significantly improves classification. Model-based training proves more effective than simple information addition. Our method further incorporates knowledge reasoning, outperforming previous approaches.

Network representation learning maps knowledge network concepts into 300-dimensional vectors, integrating topology, semantics, node labels, and reasoning information for application in other machine learning or deep learning models. Visualization is shown in Figure 10 [Figure 10: see original paper].

3.2.2 Discourse-Integrated Network Representation Learning Experiment Dataset: Using "German Industry 4.0" as the theme, we selected 10 relevant texts. Chapter node representations serve as input, and we compare experimental ranking results with original chapter ordering to validate effectiveness.

Figure 11 [Figure 11: see original paper] shows a discourse-integrated knowledge network example, where the tree-structured discourse organizes the knowledge network. Identical concepts across chapters are merged while different concepts interconnect, forming a hybrid tree-network structure.

We improved upon simple vector addition by using Doc2vec for discourse nodes and TextRank for importance ranking. Taking Germany's "Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0" as an example, Figure 12 [Figure 12: see original paper] shows top-8 ranked chapters: (a) TextRank only, (b) concept node averaging, and (c) our method.

Discourse-integrated text knowledge network validity is proven through single-document chapter importance ranking, determined by two metrics: content richness (more concepts = more important) and structural position (titles > first-level headings > second-level headings, etc., with upper-layer nodes being more general and important). Results show our method ranks three first-level headings in top positions, outperforming other methods. Concept averaging outperforms TextRank alone. Our method doesn't rank Chapters 4 and 6 highly because they lack subheadings and are short, yielding fewer extractable concepts.

References

[1] Zhang Xiaoyan, Wang Ting, Chen Huowang. Research on named entity recognition [J]. Computer Science, 2005, 32(4): 44-48. [2] Collins M, Singer Y. Unsupervised models for named entity classification [C]//1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora. Stroudsburg: ACL, 1999. [3] Bikel DM, Schwartz R, Weischedel RM. An algorithm that learns what's in a name [J]. Machine Learning, 1999, 34(1-3): 211-231. [4] Curran JR, Clark S. Language independent NER using a maximum entropy tagger [C]//Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003. Stroudsburg: ACL, 2003: 164-167. [5] McNamee P, Mayfield J. Entity extraction without language-specific resources [C]//Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003. Stroudsburg: ACL, 2003: 188-191. [6] McCallum A, Li W. Early results for named entity recognition with conditional

random fields, feature induction and web-enhanced lexicons [C]//Association for Computational Linguistics. Stroudsburg: ACL, 2003: 188-191. [7] Collobert R, Weston J, Bottou L, et al. Natural language processing (almost) from scratch [J]. Journal of Machine Learning Research, 2011, 12(Aug): 2493-2537. [8] Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging [J]. Computer Science, 2015: 1-10. [9] Pham TH, Le-Hong P. End-to-end recurrent neural network models for Vietnamese named entity recognition: Word-level vs. character-level [C]//International Conference of the Pacific Association for Computational Linguistics. Singapore: Springer, 2017: 219-232. [10] Ma X, Hovy E. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF [J]. arXiv preprint, 2016, arXiv:1603.01354. [11] Wang W, Chang L, Bin C, et al. ESN-NER: Entity storage network using attention mechanism for Chinese NER [C]//Information Processing and Cloud Computing. New York: ACM, 2019: 1-6. [12] Yu Chuanming, Huang Tingting, Lin Hongjun, et al. Research on cross-language entity extraction based on label transfer and deep learning [J]. Modern Information, 2020, 40(12): 3-16, 35. [13] Brin S. Extracting patterns and relations from the world wide web [C]//International Workshop on the World Wide Web and Databases. Berlin: Springer, 1998: 172-183. [14] Hasegawa T, Sekine S, Grishman R. Discovering relations among named entities from large corpora [C]//Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. Stroudsburg: ACL, 2004: 415. [15] Piasecki M, Ramocki R, Kalinski M. Information spreading in expanding WordNet hypernymy structure [C]//Proceedings of the International Conference on Recent Advances in Natural Language Processing. New York: ACM, 2013: 553-561. [16] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations [C]//Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2014: 701-710. [17] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality [C]//Advances in Neural Information Processing Systems. MA: MIT Press, 2013: 3111-3119. [18] Tu Cunchao, Yang Cheng, Liu Zhiyuan, et al. Survey on network representation learning [J]. Science China: Information Science, 2017(8): 32-48. [19] Grover A, Leskovec J. Node2vec: Scalable feature learning for networks [C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2016: 855-864. [20] Wang D, Cui P, Zhu W. Structural deep network embedding [C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2016: 1225-1234. [21] Yang C, Liu Z, Zhao D, et al. Network representation learning with rich text information [C]//International Joint Conference on Knowledge Discovery and Data Mining. New York: ACM, 2015: 2111-2117. [22] Tu C, Zhang Z, Liu Z, et al. TransNet: Translation-based network representation learning for social relation extraction [C]//IJCAI. New York: ACM, 2017: 2864-2870. [23] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data [C]//Advances in Neural Information Processing Systems. New York: ACM, 2013: 2787-2795. [24] Liu Dandan, Peng Cheng, Qian Longhua, et al. Comparison of lexical se-

mantic information impact on Chinese entity relationship extraction [J]. Computer Applications, 2012, 32(8): 2238-2244. [25] Liu Xiang, Ma Feicheng, Chen Xiaojun, et al. Structure and evolution of knowledge networks: Concepts and theoretical advances [J]. Information Science, 2011(6): 801-809. [26] Pan S, Jia W, Zhu X, et al. Tri-party deep network representation [C]//International Joint Conference on Artificial Intelligence. New York: ACM, 2016: 1895-1901.

Author Contributions: Liu Yao: Proposed research ideas and direction Zhang Yue: Designed research plan, conducted experiments, drafted paper Ye Lu: Responsible for data preparation, paper revision

English Abstract: Construction of Text Knowledge Network Integrating Discourse Structure

Liu Yao[1], Zhang Yue[2], Ye Lu[3]

[1]Institute of Scientific and Technical Information of China (ISTIC), Beijing 100038 [2]Michigan State University, East Lansing 489132 [3]School of Software & Microelectronics, Peking University, Beijing 100871

Abstract: [Purpose/Significance] Text vectorization is a necessary preprocessing step in text mining, information retrieval, sentiment analysis, etc. It is urgent to make node vectors contain rich and effective semantic and structural information. [Method/Process] This paper first analyzes characteristics of science and technology policy texts, then uses BiLSTM-CRF algorithms and SVM respectively to automatically index concepts and relationships according to classification systems for concepts and inter-concept relationships. Feature engineering integrates basic and syntactic-semantic features, significantly improving recognition accuracy and efficiency. We also propose a conceptual knowledge network combining reasoning knowledge and a knowledge network construction method further integrating discourse structure. [Result/Conclusion] Based on this knowledge network model, we implement a network representation learning model that can integrate node semantics, topology, and category label information, fully mining and representing textual semantic and structural information, verified through visualization and experiments.

Keywords: Named Entity Recognition; Relationship Extraction; Neural Network; Representation Learning; Discourse Structure

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