Artificial Intelligence and Social Computing, Vol. 163, 2025, 23–31
https://doi.org/10.54941/ahfe1006032

A Method of Structured Standard Terminology Based on Decoupling Approach

Xinyu Cao[1], Zhengyuan Han[2], Yi Yang[3], Liangliang Liu[4], Pai Peng[3], and Haitao Wang[1]

[1]China National Institute of Standardization, Beijing, 100191, China
[2]School of Data Science and Engineering, East China Normal University, Shanghai, 200241, China
[3]Electric Power Research Institute, State Grid Jiangsu Electric Power Co., Ltd., Nanjing, 210003, China
[4]Shanghai University of International Business and Economics, Shanghai, 201620, China

ABSTRACT

In the context of increasingly frequent interdisciplinary collaboration and global technological exchanges, constructing a terminology database is crucial for ensuring consistency in terminology and promoting effective communication. However, a large number of existing standard terminologies are stored in unstructured text files, lacking systematic organization, which hinders efficient construction and maintenance of terminology databases. Therefore, there is an urgent need to develop tools capable of accurately parsing and structuring standard terminology files. Current research primarily adopts rule-based matching and ma-chine learning methods for processing these files. However, these approaches suffer from format sensitivity and high coupling issues. The inconsistency in file formats, coupled with the difficulty for manually written style rules to comprehensively cover all scenarios, leads to poor robustness in parsing tools. Moreover, rule-based tools rely heavily on if-else logical judgments, increasing the coupling between rules and making it challenging to add new rules without causing conflicts, thus complicating maintenance and scalability. To address these issues, we propose a parsing tool tailored for standard terminology files that supports the structuring of "terms and definitions" sections from multiple file formats. The contributions of this paper include: 1) presenting a decoupled file parsing workflow; 2) proposing a set of rule matching and rule processing specific to the domain of standard terminology parsing; 3) developing and deploying an online system. In summary, the proposed parsing tool not only resolves the existing problems of format sensitivity and high coupling but also enhances the efficiency and accuracy of terminology file parsing through innovative decoupling design and domain-specific rule sets, providing strong support for the construction of terminology databases.

Keywords: Terminology, Text structuring, Decouplin

INTRODUCTION

In the context of increasingly frequent global technological exchanges and interdisciplinary collaborations, the consistency and accurate communication of specialized terminology have become key factors driving progress across various fields. However, differences in terminology across disciplines and the diversity of languages and cultures pose significant challenges to effective communication, potentially leading to misunderstandings and errors in information transmission, which can hinder knowledge sharing and technological development. Building a systematic and standardized terminology database is crucial for ensuring consistency and improving communication efficiency. Despite the existence of a large number of standard terminologies, they are mostly stored in unstructured text formats, lacking systematic organization and management, which limits retrieval and application efficiency and increases the complexity of construction and maintenance. Moreover, terminologies from different sources may contain repetitions, contradictions, or even errors, further impacting consistency and accuracy. Therefore, developing systematic and intelligent methods and technologies to optimize the collection, organization, and updating processes of terminologies has become essential for enhancing the quality and usability of terminology databases.

Current research primarily focuses on rule-based matching and machine learning parsing methods. However, due to the lack of uniformity in standard terminology file formats, manually written style rules struggle to comprehensively cover all scenarios, resulting in poor robustness of parsing tools and difficulty in handling diverse inputs. Rule-based tools rely heavily on numerous if-else logical judgments, increasing the coupling between rules, making it easy for new rules to cause conflicts, and raising the difficulty of maintenance and expansion. Therefore, exploring more flexible and adaptive parsing methods and technologies has become critical for enhancing the construction and maintenance capabilities of terminology databases, with profound implications for promoting interdisciplinary cooperation and global technological exchanges.

To address the aforementioned challenges, this paper proposes a low-coupling text structure parsing method based on rules, which supports the structured parsing of text files in standard terminology domains. This method provides insights into the text parsing process and decouples the rule matching and parsing processing steps, resolving the high coupling issue in traditional text parsing methods. In addition, the method constructs a rule matching library and a parsing processing library, enhancing the flexibility of configuration and the convenience of maintenance. The main contributions of this paper include: (1) proposing a decoupled file parsing workflow to address the high coupling challenge in traditional text parsing; (2) developing a set of rule matching and rule processing collections for standard terminology domain parsing, forming a text parsing tool configuration solution for standard terminology domain scenarios; (3) developing and deploying an online system to improve the practicality and usability of the parsing tool.

RELATED WORKS

Currently, research on terminology structuring primarily focuses on rule-based and machine learning-based methods.

Rule-based methods are traditional means for terminology extraction and structuring, which rely on manually crafted rules to parse and structure terminological texts. These methods typically depend on linguistic features (such as part-of-speech tagging and syntactic patterns) and domain knowledge. A terminology extraction method was proposed based on morphological and syntactic rules, which demonstrated high accuracy in specific domains (Kageura and Umino, 1996). Additionally, a rule-based method for standardizing patent terminology was introduced, effectively reducing terminological ambiguity through the definition of standardization rules (such as synonym mapping and contextual constraints) (Lee et al., 2020). To support multilingual terminology processing, TermSuite tool was developed, which provides rule-based terminology extraction and structuring capabilities (Daille, 2017). Although rule-based methods offer high interpretability, they have high costs in rule design and maintenance and struggle to adapt to diverse text formats and domains.

With the development of machine learning technology, data-driven methods have gradually become the mainstream approach for terminology parsing and structuring. These methods train models to automatically learn the contextual features and structural patterns of terminologies. A neural terminology parsing method was proposed which used Bidirectional Long Short-Term Memory networks (BiLSTM) and Conditional Random Fields (CRF) models, achieving significant results across multiple domains (Lample et al., 2016). In recent years, pre-trained language models based on Transformer architectures, such as BERT and GPT, have been widely applied to terminology parsing tasks. BioBERT was developed, which has shown excellent performance in terminology parsing within the biomedical field (Lee et al., 2020). Additionally, machine learning methods have been employed in multilingual terminology alignment and mapping tasks. A neural network-based terminology alignment method was proposed that significantly improved cross-lingual terminology consistency (Chen et al., 2019). Although machine learning methods possess strong generalization capabilities, they require substantial amounts of annotated data and suffer from poor model interpretability.

A LOW-COUPLING TEXT STRUCTURED PARSING METHOD BASED ON RULES

Parsing Processing Workflow

This method takes PDF files as input and ultimately parses and outputs structured terminology from the "Terms and Definitions" section, saving it in XML file format. First, the PDF file is read in, and using the positional information of characters within the PDF, the characters are organized into several text segments. Next, the list of text segments is scanned to determine the scope of the "Terms and Definitions" section. Then, an XML parsing tree

is initialized, and the text segments within the parsing range are traversed, with an iterative process of rule matching and parsing to populate the XML parsing tree. Finally, the XML parsing tree is saved as an XML file format to the local storage.

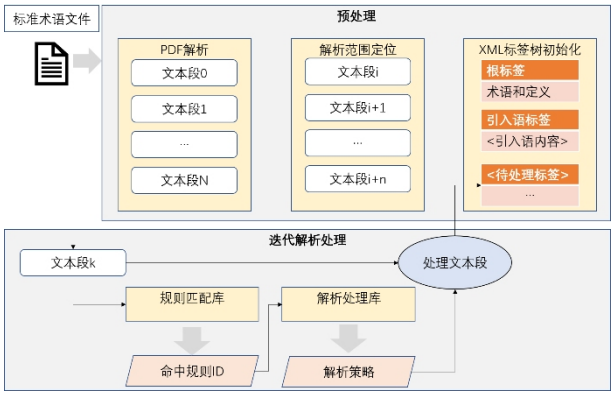

Figure 1: Rule-based low-coupling text structuring and parsing framework.

Pre-Processing

The pre-processing procedure consists of three steps: PDF reading, text block localization, and initialization of the XML parsing tree.

PDF Reading

The article takes standard terminology PDF files as input and parses the text within them to form several text segments. First, based on the positional information of characters on the page, the divided terminology file is filtered to remove irrelevant information, and missing space characters are completed. Next, according to the horizontal and vertical coordinate information of the first character of each line, it is determined whether there is an indentation at the beginning of two adjacent lines, and paragraphs are divided according to the segmentation rules. For adjacent lines that meet the indentation requirement, the two lines and the following several lines are combined into one paragraph; for lines that do not meet the indentation requirement, each line forms a separate paragraph.

Text Block Localization

Use regular expression matching to extract the top-level headings from the standard terminology file, defining the paragraphs between the "Terms and Definitions" heading and the next top-level heading as the extraction range. Specifically, traverse the list of text segments obtained from reading the aforementioned PDF, and perform regular expression matching based on the content of each text segment. This matching process can identify whether a text segment is a heading, its level in the heading hierarchy, and the textual content of the segment. Once the text segment containing the "Terms and Definitions" heading is encountered, mark the segment's number as the

starting position for parsing. When the next top-level heading is matched again, mark the segment's number as the ending position for parsing.

Initialization of the XML Tag Tree

Before the iteration begins, the XML parsing tree needs to be initialized with the root tag "Terms and Definitions," which is essentially a list of term tags. During the subsequent iterative process of rule matching and parsing processing, the text, after matching and processing, will form XML tags. These tags will be sequentially appended to the XML tag tree. Among them, there are differences in the granularity of the tags; that is, some simple tags can be formed with just a single text block, while some complex tags require multiple rounds of iteration across multiple text blocks to be formed. Therefore, when initializing the XML tag tree, it is necessary to simultaneously initialize the current tag pointer, which is used during parsing processing to specify the insertion position of its child tag information in the XML tag tree.

Rule Matching Library

This article designs a rule matching library for the scenario of annotated terminology parsing, which aggregates several preset rules into a rule library. Specifically, matching rule conditions are configured in the form of configuration files, with matching rules being if-else judgment conditions in Python. These conditions include regular expression matching of text segments and the matching results of the preceding text segment. During the iteration process, text segments within the extraction range are traversed one by one according to the set flow and judgment order to match the rules in the rule library. The conditions in the matching rules are inserted into the program as string formats to construct Python statements, which are then executed immediately for matching judgments. Once a text segment matches a corresponding rule, the marking number of the matched rule is recorded. This marking number serves as an index for subsequent parsing processing modules to determine the appropriate processing strategy.

Table 1: Rule matching library.

Rules	Contents
Rule1	<Terms and Definitions>, matches the regular expression ^(\d)+\.{0,1}\s+(term[andlor]define)$
Rule2	<Introductory Phrase>, this paragraph conforms to Rule1 and its text matches the regular expression.+ (term[andlor]define)
Rule3	numbering, the text of this paragraph matches the regular expression \s*(\d+([\..]{0,1}\d)*)\s*$
Rule4_0	<Preferred Term>, the previous paragraph conforms to Rule3, and the text of this paragraph matches the regular expression (.*?)\s(.*)
Rule4_1	<Preferred Term> (Chinese only), requires that this paragraph is not the last one, the previous paragraph conforms to Rule 3, and the next paragraph's text matches the regular expression ([a-zA-Z]{2})\s(.*)

Continued

Table 1: Continued

Rules	Contents
Rule5_0	<Preferred Term>/<Allowed Term>, the previous paragraph conforms to Rule5 or Rule4, and the text of this paragraph matches the regular expression (.*?)\s(.*), with the previous paragraph conforming to Rule4 or Rule5
Rule5_1	<Preferred Term> (Multilingual), the previous paragraph conforms to Rule5 or Rule4, and the text of this paragraph matches the regular expression ([a-zA-Z]{2})\s(.*), where the language code is in a pre-set vocabulary list
Rule5_2	Symbol, the text of this paragraph matches the regular expression \[(.*)\], and the matched result is in a pre-set symbol table
Rule6	<Term Source>, the text of this paragraph matches the regular expression \[(resource[::]){0,1}(.*?)[„](.*?)\]
Rule7	<Term Note>, the text of this paragraph matches the regular expression \s*note\s{0,1}\d{0,1}[::](.*)
Rule8	<Term Example>, the text of this paragraph matches the regular expression (example\s{0,1}\d{0,1})[::](.*)
Rule9	<N-level Term Classification>, the previous paragraph conforms to Rule3, and the text of this paragraph does not match the regular expression (.*?)\s(.*)
Rule10	<Abbreviation>, the previous paragraph conforms to Rule4 or Rule5; and the text of this paragraph has a similarity of no less than 0.4 with the current <Chinese Term>
Rule11	<Term Definition>, the previous paragraph conforms to Rule4, Rule5, or Rule10
Rule12	<Reference>, the text of this paragraph matches the regular expression reference[::](.*)

Parsing Processing Library

This article pairs each rule in the rule matching library with a corresponding parsing processing method to structurally process the current text segment and insert it into the corresponding position in the XML parsing tree.

Table 2: Parsing processing library.

Strategies	Contents
Strategy1	Begin extraction and perform regular expression matching on this text segment: ^(\d)+.{0,1}\s+(Term[and|or]Define)$, where Result1 is the serial number and Result2 is the title.
Strategy2	This line is a lead-in line, which has been processed by Strategy1, so this line is skipped.
Strategy3	This line is a redundant serial number line following the lead-in, so this line is skipped.
Strategy4_0	Perform regular expression matching on this text segment: \s*(\d+([\..]{0,1}\d)*)\s*$, Result1 is number; Perform regular expression matching on this text segment: (.*?)\s(.*), the Result1 is Chinese term.

Continued

Table 2: Continued

Strategies	Contents
Strategy4_1	Perform regular expression matching on previous text segment: \s*(\d+([\..]{0,1}\d)*)\s*$, the Result1 is number; This text is Chinese term.
Strategy5_0	Obtain the conceptEntry tag mapped by term_tag, perform regular expression matching on this text segment: (.*?)\s(.*), then Result1 is the Chinese terminology; determine whether it is a preferred term or an allowable term based on the font of the Chinese terminology, judge if it has been replaced, and process the corresponding word list.
Strategy5_1	Obtain the conceptEntry tag mapped by term_tag, perform regular expression matching on this text segment: (.*?)\s(.*), then Result1 is the language code; judge if it has been replaced, in this segment, the part following the language code is several corresponding words, extract the corresponding words and determine preferred and allowable terms.
Strategy5_2	Obtain the langSec tag mapped by en_term_tag; extract symbols and determine preferred and allowable terms; perform regular expression matching on this text segment:.*[(.*)\], where Result1 is the symbol source.
Strategy6	Obtain the conceptEntry tag mapped by term_tag, perform regular expression matching on this text segment: \[(resource[::]){0,1}(.*?)[„](.*?)\], where Result1 is the term source.
Strategy7	Obtain the conceptEntry tag mapped by term_tag, perform regular expression matching on this text segment: \s*note\s{0,1}\d{0,1}[::](.*), where Result1 is the term note.
Strategy8	Obtain the conceptEntry tag mapped by term_tag, perform regular expression matching on this text segment: (example\s{0,1}\d{0,1})[::](.*), where Result1 is the term example.
Strategy9	Obtain the sec tag mapped by root_tag, perform regular expression matching on this text segment: ^(\d+([\..]{0,1}\d)*)\s*(.*)?, where Result1 is the serial number and Result3 is the title name.
Strategy10	Obtain the conceptEntry tag mapped by term_tag, the text of this segment is the abbreviated term.
Strategy11	Obtain the conceptEntry tag mapped by term_tag, perform regular expression matching on this text segment: <(.*)>, where Result1 is the domain; in this text segment, the part following the domain is the term definition.
Strategy12	Obtain the conceptEntry tag mapped by term_tag, perform regular expression matching on this text segment: reference[::](.*), where Result1 is the reference.

Low-Coupling Incremental Maintenance

The key to structured parsing lies in the matching and judgment of text, as well as the corresponding parsing methods. This paper decouples rule matching and parsing processing to form corresponding library configuration files, which greatly facilitates the incremental maintenance process. During testing, special cases newly discovered can be flexibly maintained and improved.

EXAMPLE OF ONLINE PARSING SYSTEM

This paper has developed and deployed an online system platform based on the low-coupling structured text parsing method, which is used for real-time online processing of standard terminology text parsing. The system includes two modules: file upload and parsing, and parsing preview and download.

File Upload and Parsing

After entering the terminology extraction module, users can upload standard terminology files for online parsing. This module supports the upload of files in both PDF and Word formats. For PDF files, the system will directly use the parsing method described in this paper after receiving the file. For Word files, the system will first convert them into PDF format before parsing.

Parsing Preview and Download

After the parsing is complete, the system saves the uploaded PDF original files and the parsed XML files in folders organized by date and time. Additionally, the system provides an online preview function for both the original PDF files of the standard terminology and the parsed XML files.

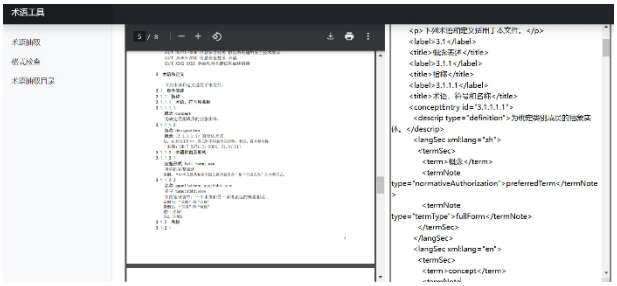

Figure 2: Parsing preview and download.

CONCLUSION

This article proposes a set of text parsing tools tailored for standard terminology scenarios, achieving effective structural processing of the "Terms and Definitions" section in standard terminology files. The main contributions of this article are as follows: Firstly, a decoupled file parsing workflow is designed and implemented, solving the issues of format sensitivity and high coupling inherent in traditional rule-based matching methods; secondly, a set of rule matching and rule processing sets specific to the standard terminology field is developed, enhancing the accuracy and efficiency of the parsing process; finally, an online parsing system platform is developed and deployed, improving the online availability of the method.

ACKNOWLEDGMENT

This research was supported by State Grid Corporation Headquarters Science and Technology Project (Project Code: 5700-202318834A-4-2-KJ)

REFERENCES

Chen, X., Liu, Z., & Sun, M. (2019). A neural approach to term alignment in multilingual texts. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP).

Daille, B. (2017). TermSuite: Terminology extraction with term variants. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL).

Kageura, K., & Umino, B. (1996). Methods of automatic term recognition: A review. Terminology, 3(2), pp. 259–289.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, 36(4), pp. 1234–1240.

Lee, S., & Martinez, P. (2020). Rule-based normalization of technical terminology in patent documents. Proceedings of the International Conference on Language Resources and Evaluation (LREC).