International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:14, No:7, 2020

DocPro: A Framework for Processing Semantic and Layout Information in Business Documents

Ming-Jen Huang, Chun-Fang Huang, Chiching Wei

***Abstract*—**With the recent advance of the deep neural network, we observe new applications of NLP (natural language processing) and CV (computer vision) powered by deep neural networks for processing business documents. However, creating a real-world document processing system needs to integrate several NLP and CV tasks, rather than treating them separately. There is a need to have a unified approach for processing documents containing textual and graphical elements with rich formats, diverse layout arrangement, and distinct semantics. In this paper, a framework that fulfills this unified approach is presented. The framework includes a representation model definition for holding the information generated by various tasks and specifications defining the coordination between these tasks. The framework is a blueprint for building a system that can process documents with rich formats, styles, and multiple types of elements. The flexible and lightweight design of the framework can help build a system for diverse business scenarios, such as contract monitoring and reviewing.

***Keywords*—**Document processing, framework, formal definition, machine learning.

I. Introduction

BUSINESS documents nowadays are usually composed of multiple types of information, such as text, images, tables, charts, formulas. Their semantics, formats, and styles are also abundant. In order to create a system to assist humans in reading, comprehension, and writing documents, there is a need to combine various technologies for analyzing textual and graphical elements. In addition to this, the analyzed results must be able to be stored and consumed by machines.

Document processing has been long considered an application of NLP [1], such as named entity recognition, sentiment analysis, semantic relations. Another application is to apply CV for document-layout analysis [2], which is to determine document structure by detecting locations, bounding boundary, and types of document elements. Another prominent CV task related to document processing is image captioning [3]. All of these individual NLP and CV tasks are already quite common in academic research. Open source components are also developed for many years [4]-[6].

To fulfill the requirements for building such a system, we propose an architecture design as a blueprint for building a system that can process documents with rich formats, styles, and multiple types of elements. The architecture includes: (1) a document representation model definition that can be instantiated with analyzed data and can be consumed by other software components and (2) a customizable framework that coordinates various tasks for analyzing documents. We define the framework with formal definitions and illustrate with examples.

In this paper, we firstly describe previous NLP and CV research work. We then describe the overall architecture of the framework. Finally, we detail the document representation model and the task coordination definition.

Ming-Jen Huang, Chun-Fang Huang, and Chiching Wei are with the Foxit Software Inc., United States (e-mail: brian_huang@foxitsoftware.com, lisa_huang@foxitsoftware.com, jeremy_wei@foxitsoftware.com).

II. Previous Work

Document processing is a vast area with many topics. We listed some of the topics below.

NLP has been applying to various document processing tasks. Name entity recognition (NER) is a task of identifying the type of an entity within the text. Supervised learning approaches usually required to prepare a dictionary or annotated datasets [7], [8]. Even this approach can create a high-performance NER model; it is a time-consuming task and needs multi-language annotated datasets. On the other hand, the weakly supervised approach starts entity classification with a small dataset or rules and expanding more rules with new iterations [9].

Similar text analysis is a task to detect similarity between sentences, paragraphs, and documents. One of the most common approaches is to calculate various types of distances between text vector spaces. The vector spaces could be calculated from terms, corpus, or knowledge [10]-[12].

Text classification is a task to assign a category to a document. Classification approaches are diverse. Among others, SVM (support vector machine) demonstrates that it is an efficient approach for document classification [13]. More recent research applies deep learning, such as CNN [14] and CNN-LSTM [15], for document classification. Hingmire et al. [16] chain two NLP tasks. They first apply topic modeling and text classification. This approach can provide a categorization explanation and high accuracy at the same time.

Summarization is a task to create a shorter version of documents with primary ideas. There are two types of output, abstractive and extractive. The abstractive summary is to generate new sentences that are in the original documents. The extractive summary, regarded as a problem of classification, is composed of sentences or paragraphs in the original documents. In general, the extractive summary can be considered as a classification problem, that is, whether a sentence is a summary sentence or not. Various approaches are proposed [17], [18]. More recent research work applies deep learning [19], [20].

CV is applied to solve image-based information of

documents. Document layout analysis detects objects and classifies them into different categories. Recent work usually applies CNN for analyzing document layout. Julca-Aguilar et al. propose CNN for detecting text/non-text document elements [21].

Image captioning is a task to give a natural language description to an image. It is a relatively new research area where the chaining of CV and NLP tasks becomes prevalent [22]. Two common approaches are: (1) capturing the main point of an image and generating a description for it [23] and (2) generating a description of each detected object and combined the descriptions [24]. Anderson et al. [25] combine both approaches to provide different levels of details.

Truică et al. [26] propose a data processing framework with a flexible data model with several preprocessing techniques. Dawborn, & Curran [27] propose and implement a document model for document representation. It is relatively rare to wok from a comprehensive approach for document processing.

In brief, in order to apply various NLP and CV tasks to real-world business scenarios, a unified framework is necessary.

III. Architecture

A. Overall design

Fig. 1 shows a conceptual diagram of the framework, which is called DocPro. The basic building blocks of DocPro include several tasks for processing input documents and generating target documents. The processing of documents creates one or more document representation models are created for storing analyzed results.

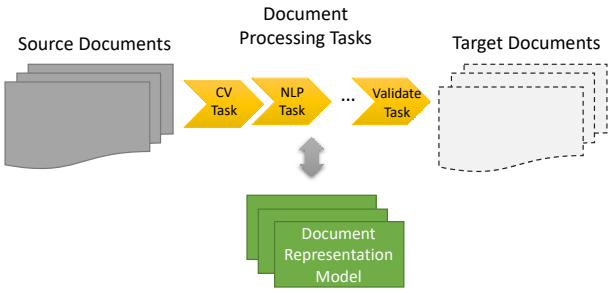

Fig. 1 Conceptual Diagram of the Document Processing Framework (DocPro)

We can assign multiple tasks based on business objectives. There are four types of tasks:

- Tasks to analyze the layout of documents
- Tasks to understand the textual elements of the documents
- Tasks to understand the graphic elements of the documents
- Tasks to write a document

B. Document Representation Model

The paper defines a document representation model to accommodate all of the necessary information for the defined tasks with a structure for holding the following information:

- **Document layout**. It includes the locations and boundaries of each document element. The types of document elements are diverse, such as footers, headers, paragraphs, charts, tables, images, formula, paragraphs.
- **Document reading order**. The semantic order of document objects of a document
- **Document summary**. A paragraph to describe the critical meaning of the original documents.
- **Document metadata**. Information extracted from documents, such as dates, time, person names, location names, organization names, etc.
- **Document text**. All of the textual data of a document originated from various document elements, such as paragraphs, image captions, table captions.
- **Document graphical description**. Textual description of graphical elements. Data can be from image and chart captions already written in a document or decoded from image pixels by machines.

The definition of the model consists of two parts. The first is the Core Model definition, a lightweight and succinct, and serves as the base model for extending. Fig. 2 shows its definition. The central class is the Document entity, which represents a single document. The Document Metadata entity stores a piece of meta-information of a document. The NamedEntity represents all detected entity names, such as person names, location names, and organization names. The Document Section is a sematic segment for holding document elements. The Document Element is an object of a document like an image or a chart.

We can define a model definition based on the Core Model definition to support other specific scenarios. Fig. 3 shows an extended model definition. In this extended version, there are several entities extended from the core model. The Lable and Category entities extended from the Document Metadata entity for holding the labels and category information of a document. The Image, Formula, Header, Footer, Chart, and Table entities are all extended from the Document Element entity. The Page entity extended from the Document Section entity, which represents a page of a document, which also holds many different Document Element entities like images, formula, header, footer, charts, and tables. From the above description, we can conclude that this extended model definition could represent a book or an academic paper. Both layout information and semantics information of books and articles are defined.

Fig. 4 is an Extension Model Definition for describing several documents correlated by some topics. A document could have one or more topics, where each topic is represented by a topic class. A topic is associated with some text excerpts, defined as the Text Excerpt entity. A topic associated with one or more documents models the correlation among documents. With this definition, we can model a knowledge map with many topics discovered from multiple documents.

Fig. 5 is another Extension Model definition defined for a more specific domain – contract review. Contract tasks are essential activities of any business. The tasks might include contract monitoring, reviewing, and drafting [28]. This model is also defined based on the Core Model shown in Fig. 2. The central component becomes the Contract entity inherited from the Document entity. The Clause entity is to model contract articles and clauses. A clause consists of one or more sections,

which are represented by the Section entity. The Text Block entity models the textual description of a section. Special items, such as recitals and preambles, can also be described by the Clause and section entities.

Temporal information is modeled by the Date class, which can hold contract start, termination, effective dates. The Value entity holds monetary information. The Period entity represents the number of days a contract is valid. Inheriting from the NamedEntity entity represents other important information like contracting parties, governing law, and jurisdiction.

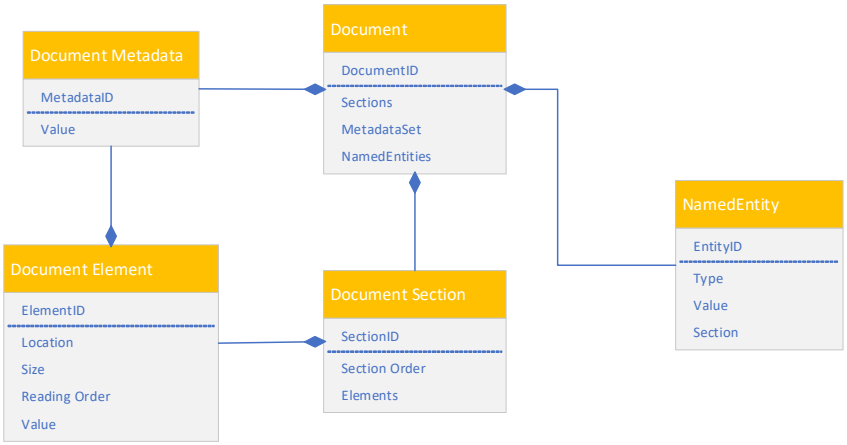

Fig. 2 The Core Model Definition of DocPro

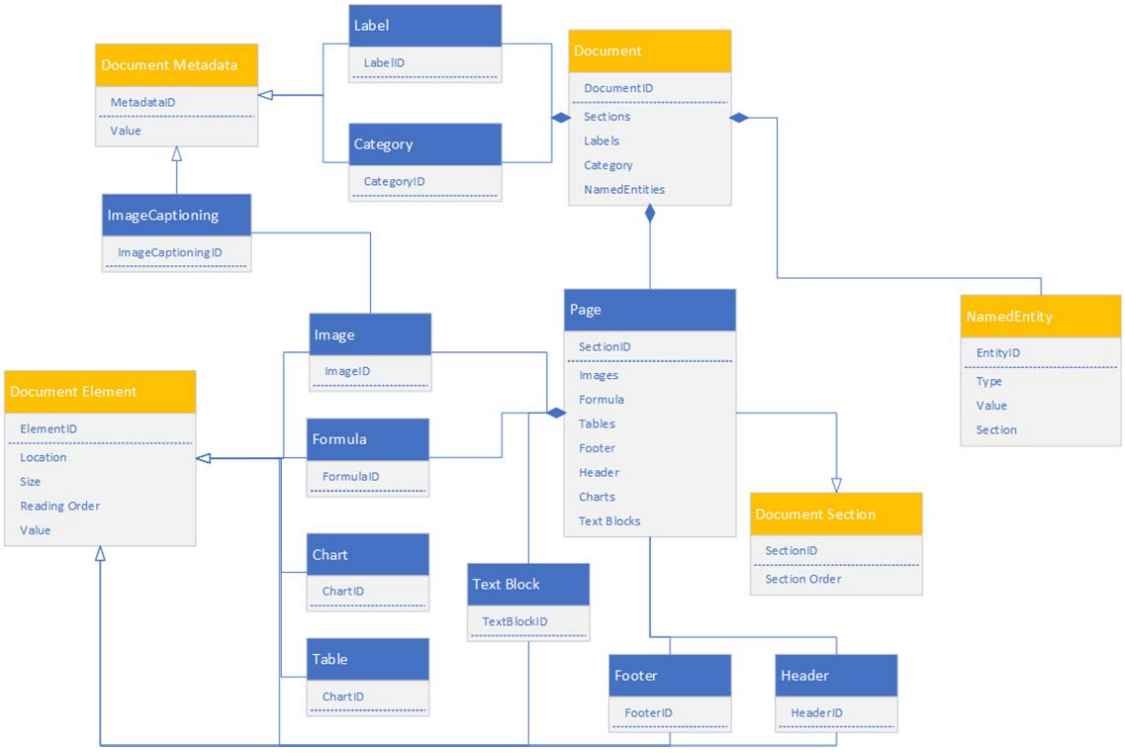

Fig. 3 The Extension Model Definition for Standalone Document

We can observe from the description above that the document representation model is beneficial for many business scenarios. One exemplified scenario is to create a tool for editing scanned documents with the information above. Another scenario can be business activity automation, like contract monitoring and review. With such core and extension model definitions, machines could streamline the decision-making process based on document contents.

IV. Document Processing Coordination Framework

In this section, we describe a framework for coordinating various types of tasks for analyzing documents with formal definitions and examples.

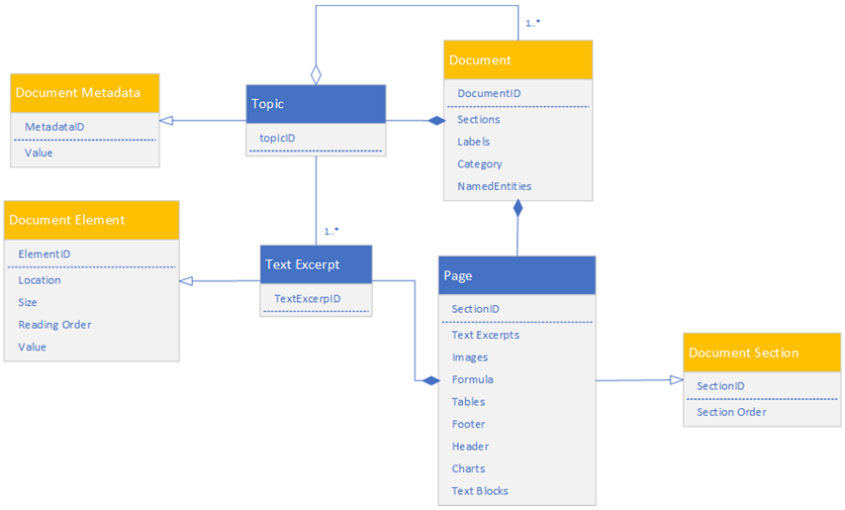

Fig. 4 The Extension Model Definition for Cross-Document Knowledge Correlation

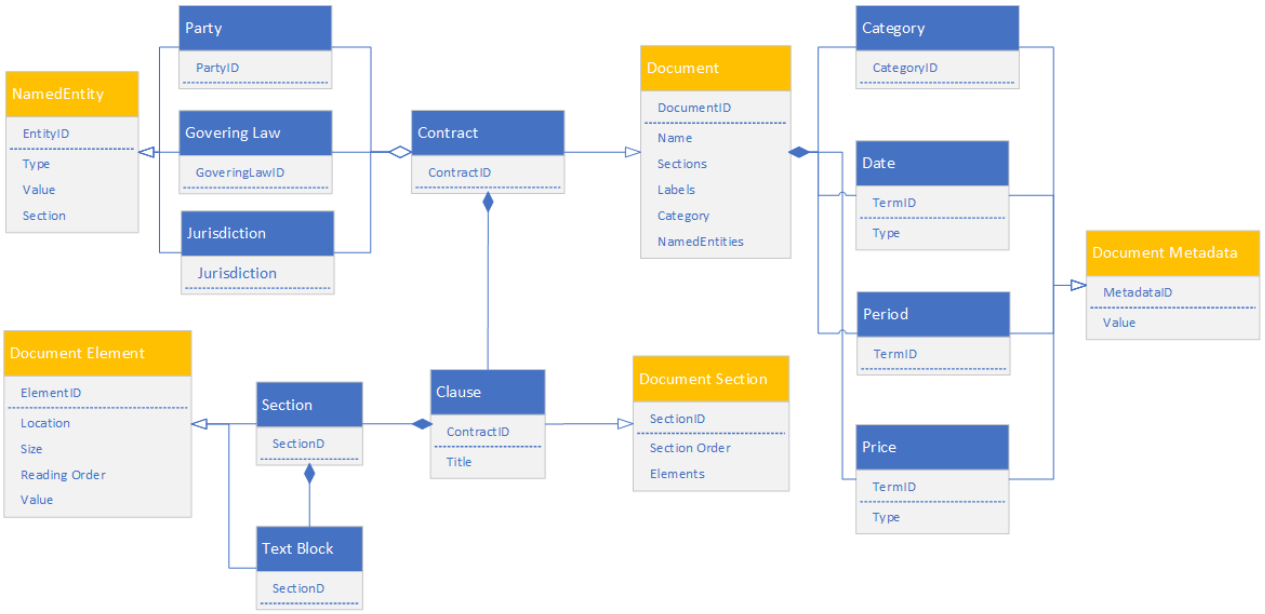

Fig. 5 The Extension Model Definition for Contracts

The limitation of current document processing technologies is isolated tasks as designed. However, with the rich format, styles, and semantics of modern documents, the integral of document analysis work becomes vital to anyone who wants to create any document processing system. Another important factor in designing the framework is that the types of tasks do not limit to machine learning implementation. We do not consider that all of the tasks for solving document processing problems would belong to the machine learning type. Therefore, the logic within each task is varied, including tasks like machine learning type for predictive work, deterministic computation type for analysis, and rule-based type for reactive responses.

The framework defines the five types of elements:

1. Data source
2. Tasks
3. Model
4. Checker
5. Process

A task consumes one or more documents emitted from a data source. A checker is a task to validate and verify the current status of models. A model is an instance for representing the analysis results of a document representation, defined in the previous section. A process is a series of connected tasks and checkers.

The framework does not specify formats of source documents. They can be PDF, MS Word, images, or others. The logic of tasks can be various, such as machine learning, rule-based, and even simple deterministic procedure. Each task should provide new information with more details and might

create a new model or update an existing model with more accurate information.

Checkers are tasks designed explicitly for validating models. Checkers are attached to a model or can be attached to elements of a model for checking the whole model or parts of the model.

Tasks and checkers can be chained together to react to the update of a model. For example, after an OCR task updates a model, a checker is triggered to check if adding new paragraphs and executing a task to classify the document into a category. Another checker checks if a document category is an employment contract and executes a new task for automatically reviewing contract contents by pre-defined rules. A process manages these chained tasks, checkers, documents, and models.

V. Math Notations

This section depicts the framework notations.

The behavior of a task is a tuple $T = (D, d_0, M_s, L, M_o)$

- D is a set of source documents.
- d_0 is the main source document.
- M_s is a set of source document representation models.
- L is a set of task labels.
- M_o is a set of output document representation models.

The behavior of a checker is a tuple $C = (M, E, L, T)$

- M is a set of document representation models.
- E is a set of elements of a document representation model m.
- L is a set of checker labels.
- T is a set of conditional tasks.

The behavior of a process is a tuple $P = (T, D, M, C)$

- T is a set of tasks.
- D is a set of source documents.
- M is a set of document representation models.
- C is a set of checkers.

T of P above can further be expressed as $p^{i \to j}$ is starting from executing task t_i and ending at executing task t_j. m_i is the model updated by t_i and m_j is updated by t_j. The process can also be denoted as follows:

$$p_{i \to j} = t_i(D) \to t_{i+1}(m_i) \to t_{i+2}(m_{i+1}) \dots \to t_j(m_{j-1}), \text{ where } \forall n = \{i, j\}: (t_n, m_n, t_{n+1}) \quad (1)$$

We can also add checkers at the end of the process and denote the process as follows:

$$p_{i \to j} = t_i(D) \to t_{i+1}(m_i) \to t_{i+2}(m_{i+1}) \dots \to t_{j-1}(m_{j-2}) \to c_j(m_{j-1}),$$

where

$$\forall n = \{i, j\}: (t_n, m_n, t_{n+1})(t_{n+1}, m_{n+1}, c_{n+2}) \quad (2)$$

A. Common Scenarios

1) Example 1

d_0 is a scanned academic paper, t_a is a task to recognize text elements from d_0, m is a model for storing the recognized results:

$$t_a = (d_0, ocr, m_0) \quad (3)$$

2) Example 2

t_b is a task to analyze images for generating their textual descriptions. When it is applied to the same document d_0 as Example 1:

$$t_b = (d_0, image - captioning, m_0) \quad (4)$$

3) Example 3

t_c is a task for discovering topics from d_0. However, this task relies on parsed textual elements. We need a conditional task c_{te} with a checker.

$$t_c = (d_0, topic - discovering, m_0) \quad (5)$$

$$c_{te} = (m_0, e_{txt}, check - textual - elements, t_c) \quad (6)$$

4) Example 4

t_d is a task to discover related topics from documents d_r for a target document d_0. In this example, t_d is applied to a model the same as to Examples 1 ~ 3. A checker c_{toc} is defined for checking topics discovered from documents.

$$t_d = (d_r, correlate - knowledge, m_0) \quad (7)$$

$$c_{toc} = (\{m_0, m_r\}, e_{toc}, check - topics, t_d) \quad (8)$$

5) Example 5

A process p_0 of building a knowledge map from a scanned document is defined as follows. The process includes the tasks and checker of Examples 1 ~ 4.

$$p_0 = t_a(d_0) \to t_b(m_0) \to c_{te}(m_0) \to c_{toc}(m_0) \quad (9)$$

B. Domain-Specific Scenarios

In this subsection, we use the same mathematical definition to define more examples specific to the contract reviewing domain.

Contract reviewing is usually a time-consuming task for lawyers or the legal departments of an enterprise. They have to verify contractual elements, read through clauses, make sure if the deal terms are property documented as expected, and no essential terms are missing. The main goal of the system is to automate some steps of the contract reviewing process, which are fully managed by humans before.

1) Example 6:

d_c is a contract. t_e is a task to classify a contract into a category. t_f is a task to recognize contractual elements in d_c. m_c is a model for storing contractual elements. c_{cpv} is a checker checking the category, parties, and the value of the contract in m_c. t_g is a task to forward the contract to CFO of a company for review.

$$t_e = (d_c, classifiy, m_c) \quad (10)$$

$$t_f = (d_c, recognize - elements, m_c) \quad (11)$$

$$t_g = (d_c, forward - high - priority - conract, m_c) \quad (12)$$

$$c_{cpv} = (m_c, \{e_{category}, e_{parties}, e_{value}\}, check - contract - priority, t_g) \quad (13)$$

2) Example 7

The first requirement of the contract automation system is to automate the process of contract routing. In Example 6, most of the components are already defined. The process p_c is defined as below:

$$p_c = t_e(d_c) \rightarrow t_f(m_c) \rightarrow c_{cpv}(m_c) \quad (14)$$

VI. Conclusions and Future Work

The primary contribution of this paper is two folds. First, we proposed a document representation model that has a lightweight and concise core definition. The core definition can be extended to define more document types used in different business scenarios. Secondly, formal definitions of a document processing framework are detailed. A modern document processing application integrating with various types of tasks can be built based on the framework. The format definitions can become the design language of any such systems.

Our future work is to create an open-source project, including the model definition, software components, and API definitions used for creating real-world systems for processing documents in different business scenarios.

References

[1] Brants, T. (2003, September). Natural Language Processing in Information Retrieval. In CLIN.
[2] Breuel, T. M. (2003, April). High performance document layout analysis. In Proceedings of the Symposium on Document Image Understanding Technology (pp. 209-218).
[3] Liu, X., Xu, Q., & Wang, N. (2019). A survey on deep neural network-based image captioning. The Visual Computer, 35(3), 445-470.
[4] OpenCV official web site. (https://opencv.org/)
[5] Gensim official web site (https://radimrehurek.com/gensim/index.html)
[6] NLP Architect by Intel (http://nlp_architect.nervanasys.com/)
[7] GuoDong, Z., & Jian, S. (2004, August). Exploring deep knowledge resources in biomedical name recognition. In Proceedings of the international joint workshop on natural language processing in biomedicine and its applications (pp. 96-99). Association for Computational Linguistics.
[8] Jiang, R., Banchs, R. E., & Li, H. (2016, August). Evaluating and combining name entity recognition systems. In Proceedings of the Sixth Named Entity Workshop (pp. 21-27).
[9] Irmak, U., & Kraft, R. (2010, April). A scalable machine-learning approach for semi-structured named entity recognition. In Proceedings of the 19th international conference on World wide web (pp. 461-470).
[10] Huang, A. (2008, April). Similarity measures for text document clustering. In Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand (Vol. 4, pp. 9-56).
[11] Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological review, 104(2), 211.
[12] Mihalcea, R., Corley, C., & Strapparava, C. (2006, July). Corpus-based and knowledge-based measures of text semantic similarity. In Aaai (Vol. 6, No. 2006, pp. 775-780).
[13] Manevitz, L. M., & Yousef, M. (2001). One-class SVMs for document classification. Journal of machine Learning research, 2(Dec), 139-154.
[14] Johnson, R., & Zhang, T. (2014). Effective use of word order for text categorization with convolutional neural networks. arXiv preprint arXiv:1412.1058.
[15] Zhou, C., Sun, C., Liu, Z., & Lau, F. (2015). A C-LSTM neural network for text classification. arXiv preprint arXiv:1511.08630.
[16] Hingmire, S., Chougule, S., Palshikar, G. K., & Chakraborti, S. (2013, July). Document classification by topic labeling. In Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval (pp. 877-880).
[17] Wang, L., & Cardie, C. (2013, August). Domain-independent abstract generation for focused meeting summarization. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1395-1405).
[18] Liao, K., Lebanoff, L., & Liu, F. (2018). Abstract meaning representation for multi-document summarization. arXiv preprint arXiv:1806.05655.
[19] Liu, Y., & Lapata, M. (2019). Text summarization with pretrained encoders. arXiv preprint arXiv:1908.08345.
[20] See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368.
[21] Julca-Aguilar, F. D., Maia, A. L., & Hirata, N. S. (2017, October). Text/non-text classification of connected components in document images. In 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI) (pp. 450-455). IEEE.
[22] You, Q., Jin, H., Wang, Z., Fang, C., & Luo, J. (2016). Image captioning with semantic attention. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4651-4659).
[23] Chen, X., & Zitnick, C. L. (2014). Learning a recurrent visual representation for image caption generation. arXiv preprint arXiv:1411.5654.
[24] Elliott, D., & Keller, F. (2013, October). Image description using visual dependency representations. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (pp. 1292-1302).
[25] Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., & Zhang, L. (2018). Bottom-up and top-down attention for image captioning and visual question answering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6077-6086).
[26] Truică, Ciprian-Octavian, Jérôme Darmont, and Julien Velcin. "A scalable document-based architecture for text analysis." International Conference on Advanced Data Mining and Applications. Springer, Cham, 2016.
[27] Dawborn, T., & Curran, J. R. (2014, August). docrep: A lightweight and efficient document representation framework. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers (pp. 762-771).
[28] Milosevic, Z., Gibson, S., Linington, P. F., Cole, J., & Kulkarni, S. (2004, July). On design and implementation of a contract monitoring facility. In Proceedings. First IEEE International Workshop on Electronic Contracting, 2004. (pp. 62-70). IEEE.