DOCPRO: A FRAMEWORK FOR BUILDING DOCUMENT PROCESSING SYSTEMS

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

Foxit Software Inc., Albrae Street, Fremont, USA

ABSTRACT

With the recent advance of the deep neural network, we observe new applications of natural language processing (NLP) and computer vision (CV) technologies. Especaully, when applying them to document processing, NLP and CV tasks are usually treated individually in research work and open source libraries. However, designing a real-world document processing system needs to weave NLP and CV tasks and their generated information together. 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. This paper introduces a framework to fulfil this need. The framework includes a representation model definition for holding the generated information and specifications defining the coordination between the NLP and CV tasks.

KEYWORDS

Document Processing, Framework, Formal definition, Machine Learning.

1. 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. 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 Natural Language Processing (NLP) [1], such as named entity recognition, sentiment analysis, semantic relations. Another application is to apply Computer Vision (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, 5, 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.

David C. Wyld et al. (Eds): ICAITA, CDKP, SAI, NCO, CMC, SOFT, MLT, AdNLP - 2020
pp. 213-224, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.100917

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.

2. 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 [10]. The vector spaces could be calculated from terms, corpus, or knowledge [11, 12, 13].

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 [14]. More recent research applies deep learning, such as CNN [15] and CNN-LSTM [16], for document classification. Hingmire et al. [17] 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 [18, 19]. More recent research work applies deep learning [20, 21].

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 [22].

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 [23]. Two common approaches are (1) capturing the main point of an image and generating a description for it [24] and (2) generating a description of each detected object and combined the descriptions [25]. Anderson et al. [26] combine both (1) and (2) approaches to provide different levels of details. Grounded language acquisition is a representatively interdisciplinary field of CV and NLP integration [27]. It requires both disciplines to map language representation to real-world objects. Mavridis and Roy [28] define an architecture with language understanding, visual perception, and action modules for robots and human cooperation tasks.

Truica et al. [29] propose a data processing framework with a flexible data model with several preprocessing techniques. Dawborn, T., & Curran, J. R. [30] propose and implement a document model for document representation. It is relatively rare to wok from a comprehensive approach for document processing.

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

3. ARCHITECTURE

3.1. Overall Design

Figure 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 that are designed for storing analyzed results.

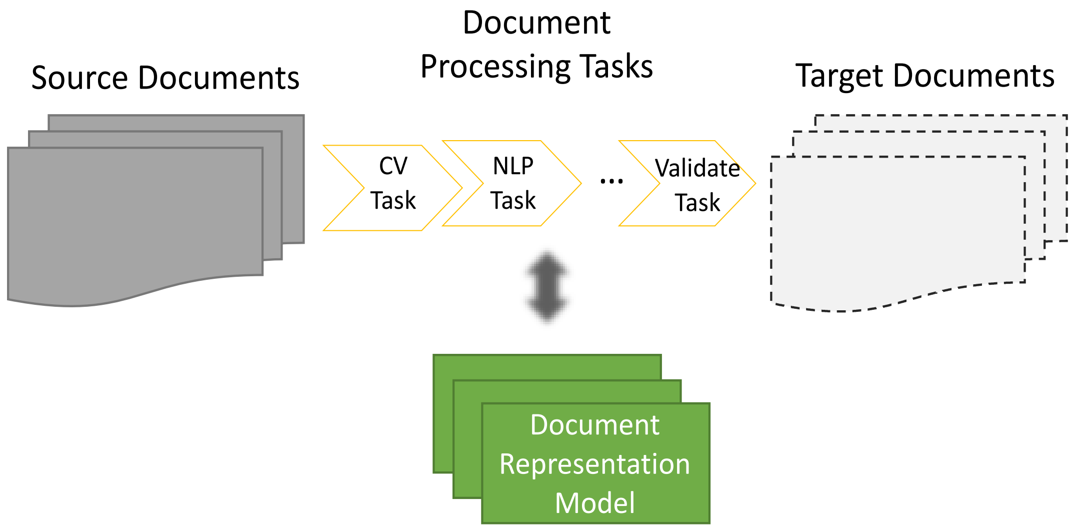


Figure 1. Conceptual Diagram of the Document Processing Framework (DocPro)

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

3.2. 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, formulas, and 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, and as such.
- 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.

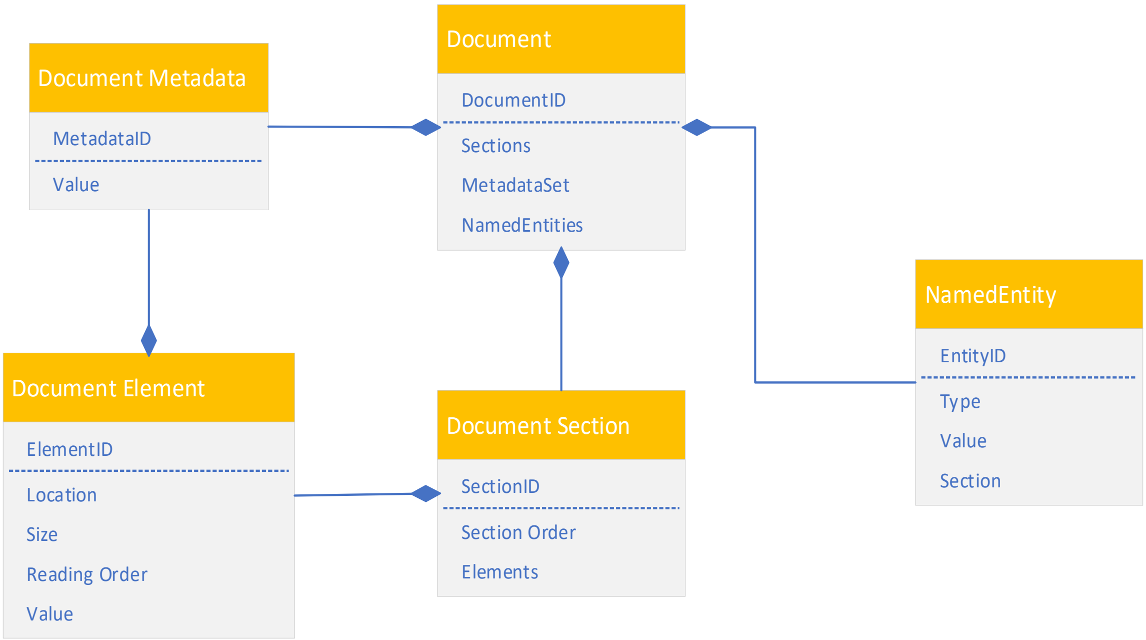


Figure 2. The Core Model Definition of DocPro

The definition of the model consists of two parts. The first is the Core Model definition, a lightweight and concise, and serves as the base model for extending. Figure 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 new model definition based on the Core Model definition to support other specific scenarios. Figure 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.

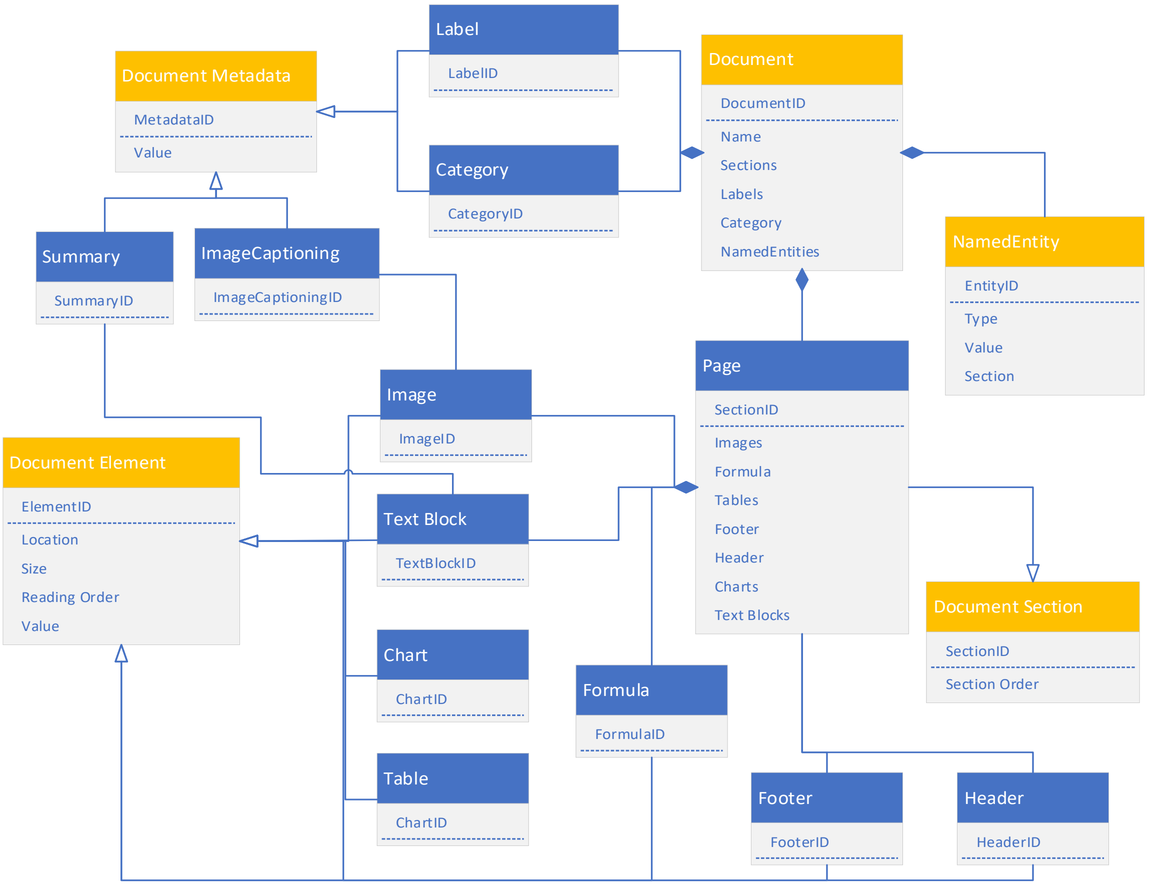


Figure 3. The Extension Model Definition for Standalone Document

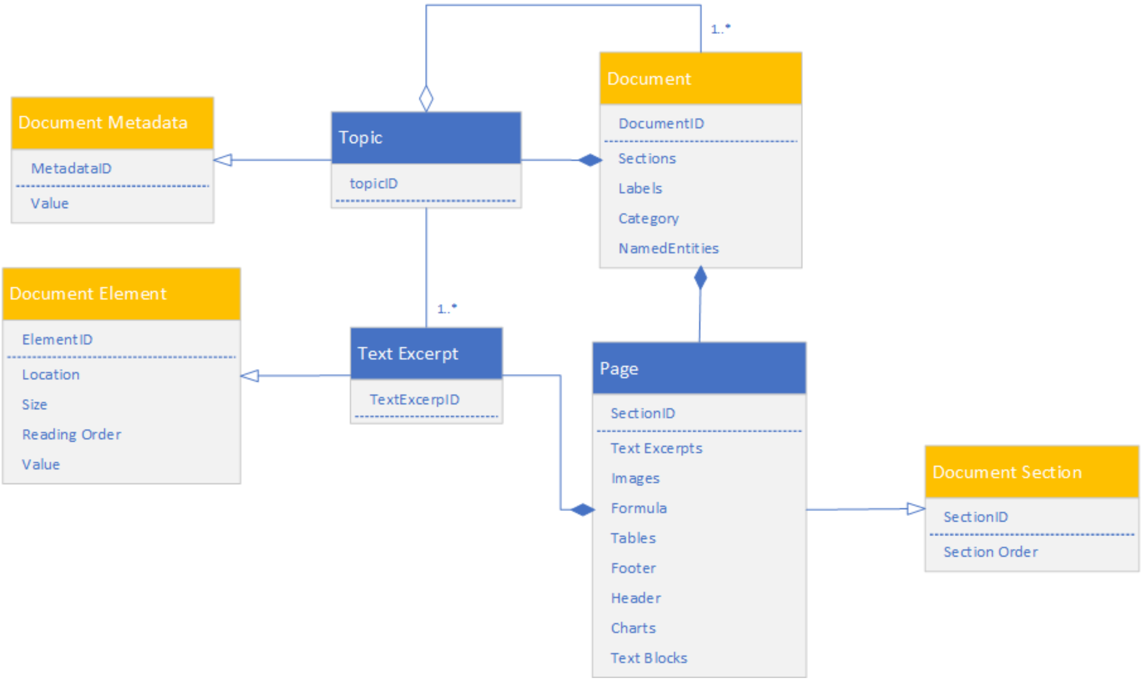


Figure 4. The Extension Model Definition for Cross-Document Knowledge Correlation

Figure 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 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.

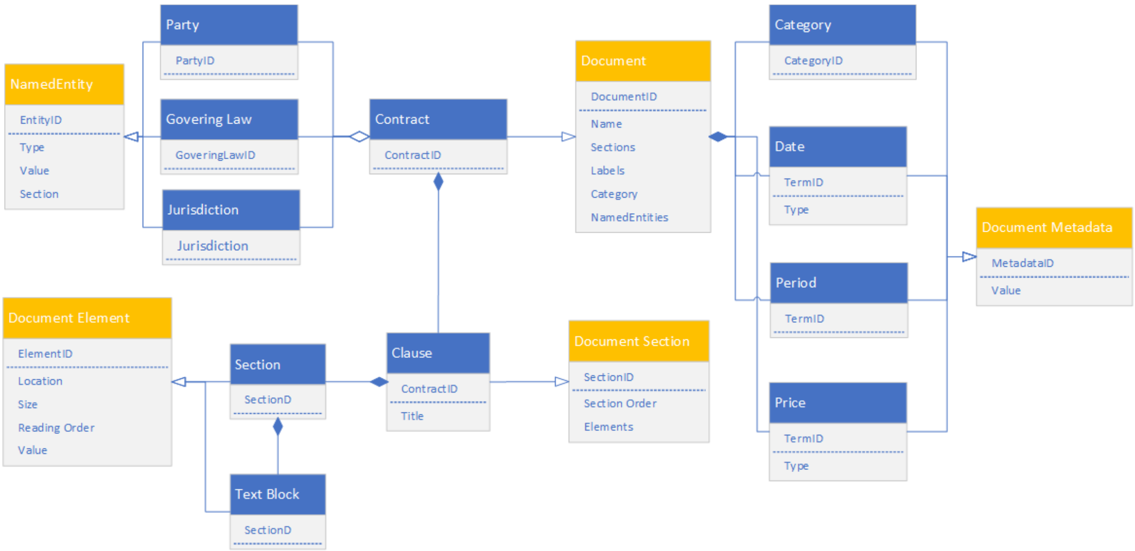


Figure 5. The Extension Model Definition for Contracts

Figure 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 [29]. This model is also defined based on the Core Model shown in Figure 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 contains financial 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.

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.

4. 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.

The limitation of current document processing technologies is isolated tasks as designed. However, with the rich format, styles, and semantics of documents, the integral of document analysis work becomes vital to anyone who wants to create any document processing system. Another critical 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:

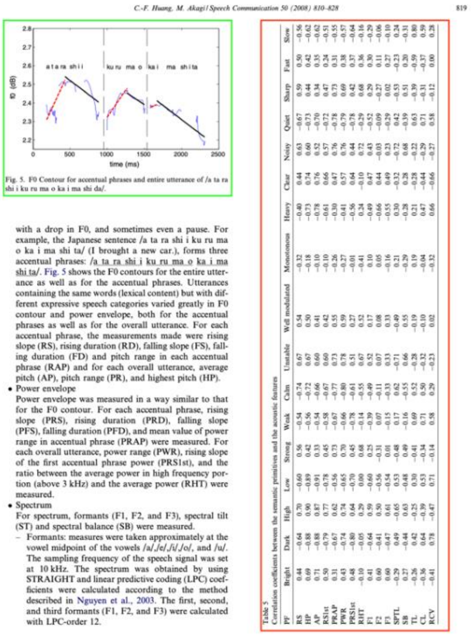
- Data source
- Tasks
- Model
- Checker
- Process

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 existing a 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.



C.-F. Huang, M. Akagi / Speech Communication 50 (2008) 810–828 819

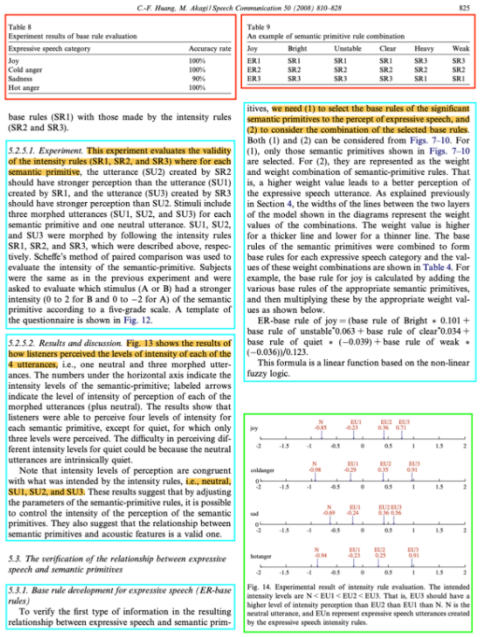
Fig. 5. F0 Contour for accentual phrases and entire utterance of /a ta ra shi i ku ru ma o ka i ma shi da/.

with a drop in F0, and sometimes even a pause. For example, the Japanese sentence /a ta ra shi i ku ru ma o ka i ma shi ta/ (I brought a new car.), forms three accentual phrases: /a ta ra shi i ku ru ma o ka i ma shi ta/. Fig. 5 shows the F0 contours for the entire utterance as well as for the accentual phrases. Utterances containing the same words (lexical content) but with different expressive speech categories varied greatly in F0 contour and power envelope, both for the accentual phrases as well as for the overall utterance. For each accentual phrase, the measurements made were rising slope (RS), rising duration (RD), falling slope (FS), falling duration (FD) and pitch range in each accentual phrase (RAP) and for each overall utterance, average pitch (AP), pitch range (PR), and highest pitch (HP).

- Power envelope
  Power envelope was measured in a way similar to that for the F0 contour. For each accentual phrase, rising slope (PRS), rising duration (PRD), falling slope (PFS), falling duration (PFD), and mean value of power range in accentual phrase (PRAP) were measured. For each overall utterance, power range (PWR), rising slope of the first accentual phrase power (PRS1st), and the ratio between the average power in high frequency portion (above 3 kHz) and the average power (RHT) were measured.
- Spectrum
  For spectrum, formants (F1, F2, and F3), spectral tilt (ST) and spectral balance (SB) were measured.
  - Formants: measures were taken approximately at the vowel midpoint of the vowels /a/,/e/,/i/,/o/, and /u/. The sampling frequency of the speech signal was set at 10 kHz. The spectrum was obtained by using STRAIGHT and linear predictive coding (LPC) coefficients were calculated according to the method described in Nguyen et al., 2003. The first, second, and third formants (F1, F2, and F3) were calculated with LPC-order 12.

Table 5
Correlation coefficients between the semantic primitives and the acoustic features

PF	Bright	Dark	High	Low	Strong	Weak	Calm	Unstable	Well modulated	Monotonous	Heavy	Clear	Noisy	Quiet	Sharp	Fast	Slow
RS	0.44	−0.64	0.70	−0.60	0.56	−0.54	−0.74	0.67	0.54	−0.32	−0.40	0.44	0.63	−0.67	0.59	0.50	−0.56
HP	0.69	−0.88	0.90	−0.89	0.42	−0.56	−0.72	0.67	0.50	−0.18	−0.73	0.74	0.60	−0.73	0.44	0.42	−0.62
AP	0.71	−0.88	0.87	−0.91	0.33	−0.54	−0.66	0.60	0.41	−0.10	−0.78	0.76	0.52	−0.70	0.34	0.35	−0.62
RS1st	0.50	−0.79	0.77	−0.78	0.45	−0.58	−0.67	0.60	0.42	−0.10	−0.61	0.66	0.57	−0.72	0.47	0.24	−0.51
PRAP	0.31	−0.67	0.62	−0.56	0.73	−0.67	−0.77	0.73	0.55	−0.26	−0.30	0.47	0.76	−0.78	0.73	0.31	−0.55
PWR	0.43	−0.74	0.74	−0.65	0.70	−0.66	−0.80	0.78	0.59	−0.27	−0.41	0.57	0.76	−0.79	0.69	0.38	−0.57
PRS1st	0.48	−0.80	0.64	−0.70	0.45	−0.78	−0.61	0.51	0.27	−0.01	−0.56	0.64	0.44	−0.78	0.42	0.37	−0.64
RHT	−0.10	−0.05	0.29	0.00	0.68	−0.14	−0.55	0.67	0.52	−0.41	0.24	−0.10	0.72	−0.29	0.68	0.36	−0.16
F1	0.41	−0.64	0.59	−0.60	0.25	−0.39	−0.49	0.52	0.17	0.10	−0.49	0.47	0.43	−0.52	0.29	0.30	−0.29
F2	0.60	−0.41	0.50	−0.56	−0.31	0.07	−0.11	0.07	0.08	0.05	−0.66	0.44	−0.03	−0.09	−0.27	0.11	−0.06
F3	0.60	−0.47	0.61	−0.54	0.01	−0.15	−0.33	0.33	0.33	−0.16	−0.55	0.49	0.23	−0.29	0.02	0.27	−0.10
SPTL	−0.29	0.49	−0.65	0.53	−0.48	0.17	0.62	−0.71	−0.49	0.21	0.30	−0.32	−0.72	0.42	−0.53	−0.23	0.24
SB	0.27	−0.44	0.63	−0.48	0.49	−0.16	−0.55	0.66	0.55	−0.29	−0.28	0.28	0.68	−0.39	0.51	0.20	−0.31
TL	−0.26	0.42	−0.25	0.30	−0.41	0.69	0.52	−0.28	−0.19	0.19	0.21	−0.28	−0.22	0.63	−0.39	−0.59	0.80
CL	−0.36	0.64	−0.39	0.53	−0.34	0.71	0.50	−0.32	−0.10	−0.04	0.47	−0.44	−0.29	0.71	−0.31	−0.37	0.59
RCV	−0.41	0.78	−0.47	0.71	−0.14	0.58	0.29	−0.23	0.02	−0.32	0.66	−0.66	−0.27	0.58	−0.12	0.00	0.28



C.-F. Huang, M. Akagi / Speech Communication 50 (2008) 810–828 825

Table 8
Experiment results of base rule evaluation

Expressive speech category	Accuracy rate
Joy	100%
Cold anger	100%
Sadness	90%
Hot anger	100%

Table 9
An example of semantic primitive rule combination

Joy	Bright	Unstable	Clear	Heavy	Weak
ER1	SR1	SR1	SR1	SR3	SR3
ER2	SR2	SR2	SR2	SR2	SR2
ER3	SR3	SR3	SR3	SR1	SR1

base rules (SR1) with those made by the intensity rules (SR2 and SR3).

5.2.5.1. Experiment. This experiment evaluates the validity of the intensity rules (SR1, SR2, and SR3) where for each semantic primitive, the utterance (SU2) created by SR2 should have stronger perception than the utterance (SU1) created by SR1, and the utterance (SU3) created by SR3 should have stronger perception than SU2. Stimuli include three morphed utterances (SU1, SU2, and SU3) for each semantic primitive and one neutral utterance. SU1, SU2, and SU3 were morphed by following the intensity rules SR1, SR2, and SR3, which were described above, respectively. Scheffe's method of paired comparison was used to evaluate the intensity of the semantic-primitive. Subjects were the same as in the previous experiment and were asked to evaluate which stimulus (A or B) had a stronger intensity (0 to 2 for B and 0 to −2 for A) of the semantic primitive according to a five-grade scale. A template of the questionnaire is shown in Fig. 12.

5.2.5.2. Results and discussion. Fig. 13 shows the results of how listeners perceived the levels of intensity of each of the 4 utterances, i.e., one neutral and three morphed utterances. The numbers under the horizontal axis indicate the intensity levels of the semantic-primitive; labeled arrows indicate the level of intensity of perception of each of the morphed utterances (plus neutral). The results show that listeners were able to perceive four levels of intensity for each semantic primitive, except for quiet, for which only three levels were perceived. The difficulty in perceiving different intensity levels for quiet could be because the neutral utterances are intrinsically quiet.

Note that intensity levels of perception are congruent with what was intended by the intensity rules, i.e., neutral, SU1, SU2, and SU3. These results suggest that by adjusting the parameters of the semantic-primitive rules, it is possible to control the intensity of the perception of the semantic primitives. They also suggest that the relationship between semantic primitives and acoustic features is a valid one.

5.3. The verification of the relationship between expressive speech and semantic primitives

5.3.1. Base rule development for expressive speech (ER-base rules)

To verify the first type of information in the resulting relationship between expressive speech and semantic primitives, we need (1) to select the base rules of the significant semantic primitives to the percept of expressive speech, and (2) to consider the combination of the selected base rules. Both (1) and (2) can be considered from Figs. 7–10. For (1), only those semantic primitives shown in Figs. 7–10 are selected. For (2), they are represented as the weight and weight combination of semantic-primitive rules. That is, a higher weight value leads to a better perception of the expressive speech utterance. As explained previously in Section 4, the widths of the lines between the two layers of the model shown in the diagrams represent the weight values of the combinations. The weight value is higher for a thicker line and lower for a thinner line. The base rules of the semantic primitives were combined to form base rules for each expressive speech category and the values of these weight combinations are shown in Table 4. For example, the base rule for joy is calculated by adding the various base rules of the appropriate semantic primitives, and then multiplying these by the appropriate weight values as shown below.

ER-base rule of joy = (base rule of Bright $*$ 0.101 + base rule of unstable*0.063 + base rule of clear*0.034 + base rule of quiet $*$ (−0.039) + base rule of weak $*$ (−0.036))/0.123.

This formula is a linear function based on the non-linear fuzzy logic.

Fig. 14. Experimental result of intensity rule evaluation. The intended intensity levels are N < EU1 < EU2 < EU3. That is, EU3 should have a higher level of intensity perception than EU2 than EU1 than N. N is the neutral utterance, and EUn represent expressive speech utterances created by the expressive speech intensity rules.

Figure 6a. An exemplified result of document processing. This page is marked with two bounding boxes, where the green one is a figure and the red one is a table. They are the analysis result of an object detection task for detecting the bounding box of an identified element and its type.

Figure 6b. Another exemplified result of document processing. This page is marked with more bounding boxes and highlighted summaries. In addition to the object detection task, an NLP task predicts the summary of each text block identified by the object detection task.

Figures 6a and 6b illustrate a document processing example on an academic paper. There are three tasks applied. First, an object detection CV task detects bounding boxes of identified document elements and their types. The results are stored as a document representation model described in Section 3. Then, a summarization task predicts a summary for each text block identified by the object detection task. The document representation model is updated with the summaries. Last, a marking task marks the bounding boxes of the detected elements and highlights summaries on each page of the paper.

5. MATH NOTATIONS

This section depicts the framework notations.

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

D is a set of source documents.
d_0 is the primary 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.
O is a set of output articles.

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.

An M consists of one or more entities defined in Section 3. It always includes a Document entity with several associated objects. For example, after applying a document-layout analysis task to a document, an m_o is initiated with a Document entity and a set of associated objects, including a Header, a Footer, and many Text Blocks. These associated objects are Edefined above.

T of P above can further be expressed as $p^{i\rightarrow j}$is starting from executing task t_i and ending at 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:

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

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

$p_{i\rightarrow j} = t_i(D) \rightarrow t_{i+1}(m_i) \rightarrow t_{i+2}(m_{i+1}) \dots \rightarrow\ t_{j-1}(m_{j-2}) \rightarrow\ c_j(m_{j-1})$,
where $\forall n = \{i, j\} t_n, m_n, t_{n+1})(\ t_{n+1}, m_{n+1}, c_{n+2})$

Below we use eight examples to illustrate the syntax and semantics of this formal definition.

Example 1 – OCR

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:

$t_a = (d_0, ocr, m_0)$

Example 2 – Summarization

t_b is a task to analyze text blocks for generating summaries. When it is applied to the same document d_0 as Example 1:

$t_b = (d_0, summarize, m_0)$

Example 3 -Topic Discovery

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.

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

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

Example 4 – Topic Correlation

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.

$t_d = (d_r, correlate - knowledge, m_0)$

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

Example 5 – Knowledge Mapping Automation

A process p_0 of building a knowledge map with connected documents correlated by topics. The process includes the tasks and checker of Examples 1 ~ 4. A process is usually for defining a data processing pipeline.

$p_0 = t_a(d_0) \rightarrow t_b(m_0) \rightarrow c_{te}(m_0) \rightarrow c_{toc}(m_0)$

Example 6 – Report Generation

t_r is a task to fine-tune a language model LR for generating financial reports. D_t is a set of financial documents to be used as training data for t_r. t_w is a task for generating a report $R$from a specific topic. A checker c_{rpt} is defined for checking topics discovered from documents and execute t_w.

$t_r = (D_t, fine - tune, LR)$

$t_w = (M, write, R)$

$c_{rpt} = (M, e_{toc}, check - topics, t_w)$

Example 7 – Contract Review Assist

$d_c$is a contract. t_e is a task to classify a contract into a category. t_r is a task to extract essential data, such as contractual party names, start/end dates, and monetary values, in d_c. m_c is a model for storing contractual elements. t_l is a task to analyze the contract layout. More specifically, it recognizes clauses appearing in d_c. c_{cpvl} is a checker checking the contract category, parties, dates, values, and layout of the contract in m_c. t_f is a task to forward the contract to the right person for further review based on the analyzed results by t_e , t_r, and t_l.

$t_e = (d_c, classifiy, m_c)$

$t_r = (d_c, recognize - named - entity, m_c)$

$t_l = (d_c, analyze - layout, m_c)$

$t_f = (d_c, forward, m_c)$
$c_{cpv} = (m_c, \{e_{category}, e_{parties}, e_{value}, e_{layout}\}, check - contract - errors, t_f)$

Example 8 – Contract Review Automation

The first requirement of the contract automation system is to automate the process of contract routing. Most of the components are already defined in Example 7. The process p_c is defined as below:

$$p_c = t_e(d_c) \rightarrow t_r(m_c) \rightarrow t_l(d_c) \rightarrow c_{cpvl}(m_c)$$

6. 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 extend 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] 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).

[11] 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).

[12] 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.

[13] 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).

[14] Manevitz, L. M., & Yousef, M. (2001). One-class SVMs for document classification. Journal of machine Learning research, 2(Dec), 139-154.

[15] Johnson, R., & Zhang, T. (2014). Effective use of word order for text categorization with convolutional neural networks. arXiv preprint arXiv:1412.1058.

[16] Zhou, C., Sun, C., Liu, Z., & Lau, F. (2015). A C-LSTM neural network for text classification. arXiv preprint arXiv:1511.08630.

[17] 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).

[18] 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).

[19] Liao, K., Lebanoff, L., & Liu, F. (2018). Abstract meaning representation for multi-document summarization. arXiv preprint arXiv:1806.05655.

[20] Liu, Y., &Lapata, M. (2019). Text summarization with pretrained encoders. arXiv preprint arXiv:1908.08345.

[21] See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368.

[22] 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.

[23] 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).

[24] Chen, X., &Zitnick, C. L. (2014). Learning a recurrent visual representation for image caption generation. arXiv preprint arXiv:1411.5654.

[25] 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).

[26] 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).

[27] Wiriyathammabhum, P., Summers-Stay, D., Fermüller, C., &Aloimonos, Y. (2016). Computer vision and natural language processing: recent approaches in multimedia and robotics. ACM Computing Surveys (CSUR), 49(4), 1-44.

[28] Mavridis, N., & Roy, D. (2006, October). Grounded situation models for robots: Where words and percepts meet. In 2006 IEEE/RSJ international conference on intelligent robots and systems (pp. 4690-4697). IEEE.

[29] 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.

[30] 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).

[31] 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.