Ontologies-based Optical Character Recognition-error Correction Method for Bar Graphs

Sarunya Kanjanawattana
Graduate School
Shibaura Institute of Technology
Tokyo, Japan
e-mail: nb14503@shibaura-it.ac.jp

Masaomi Kimura
Information Science and Engineering
Shibaura Institute of Technology
Tokyo, Japan
e-mail: masaomi@sic.shibaura-it.ac.jp

***Abstract*— Graphs provide an effective method for briefly presenting significant information appearing in academic literature. Readers can benefit from automatic graph information extraction. The conventional technique uses optical character recognition (OCR). However, OCR results can be imperfect because its performance depends on factors such as image quality. This becomes a critical problem because misrecognition provides incorrect information to readers and causes misleading communication. Numerous publications have appeared in recent years documenting OCR performance improvement and OCR result correction; however, only a few studies have focused on the use of semantics to solve this problem. In this study, we propose a novel method for OCR-error correction using several techniques, including ontologies, natural language processing, and edit distance. The input of this study includes bar graphs and associated information, such as their captions and cited paragraphs. We implemented five conditions to cover all possible situations for acquiring the most similar words as substitutes for incorrect OCR results. Moreover, we used DBpedia and WordNet to find word categories and part-of-speech tags. We evaluated our method by comparing performance rates, i.e., accuracy and precision, with our previous method using only the edit distance technique. As a result, our method provided higher performance rates than the other method. Our method's overall accuracy reached 81%, while that of the other method was 54%. Based on the evidence, we conclude that our solution to the OCR problem is effective.**

Keywords- OCR-error correction; dependency parsing; ontology; edit distance; two-dimensional bar graphs.

I. INTRODUCTION

Scientific literature has grown remarkably in recent years, and document recognition plays an important role in extracting information from the literature [1]. Typically, to understand the principal idea of a particular item of literature, readers must gradually read a detailed part of the literature. However, acquiring only descriptive details can result in unclear explanations. Imagine that an author endeavors to explain experimental results and presents some measurement data to readers. In such a case, the most suitable means might be to use a graph to present the data and their tendencies. A graph contains a lot of essential information that people can interpret easily; therefore, developing a system that can extract information from graphs can be expected to be particularly useful for gaining new knowledge more easily than ever; see, e.g., [2] and [3]. Optical character recognition (OCR) is the most basic and effective method for extracting information from graphs. However, this technique cannot guarantee perfect outcomes because OCR performance depends on many factors, such as image quality, specific language requirement, and image noise. As a result, if OCR is sensitive to such factors, error recognition can negatively affect our desired information. To alleviate this difficulty, there have been many studies proposing efficient methods based on several techniques, such as image processing and semantics. In addition, our aim in this study is to mitigate the difficulty of incorrect character recognition as well as develop an automatic system for extracting and correcting OCR errors from graphs based on ontologies.

OCR is an indispensable technique for information extraction from graphs. It has long been well known as an image processing approach that solves such problems as detecting and recognizing text in complex images and video frames [4][5]. Recently, OCR has been used extensively in many applications, such as the medical article citation database MEDLINE [6] and academic applications. For example, Kataria et al. [2] proposed an efficient method for automatically extracting elements (e.g., axis-labels, legends, and data points) from within a two-dimensional graph. Huang et al. [3] also presented a study targeting the association of recognition results of textual and graphical information contained in scientific chart images. They individually recognized text and graphical regions of an input image and combined the results of graph components to achieve a full understanding of an input image. These studies focused on investigating effective methods for extracting important graph components, similarly to our study. In contrast to this previous study, we solved OCR problems practically that might have occurred in our results. We not only extracted graph components using the OCR technique, but also addressed an OCR error problem by correcting errors using our methods.

In general, there are two types of word errors that can be found in our study, non-word and real-word errors [7]. A non-word error occurs when OCR extracts a source text as a string that does not correspond validly to any vocabulary item in the dictionary. If an extracted word matches an item in the dictionary, but is not identical with the source-text

 ISBN: 978-1-61208-507-4

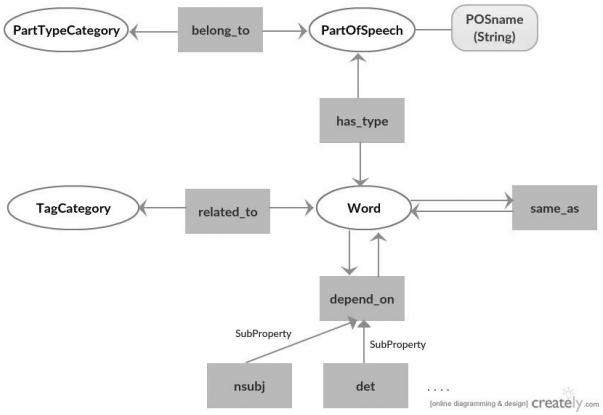

Figure 1. Illustration of our ontology structure to describe classes, properties, and relations.

word, we call it a real-word error. For example, if a source text "A dog is cute" is rendered as "A doq is rule" using OCR, then "doq" is a non-word error, and "rule" is a real-word error.

Over several years, a great deal of effort has developed several techniques for correcting such OCR errors [8]. Nagata [9] proposed an OCR-error correction method for Japanese consisting of a statistical OCR model, an approximate word-matching method using character shape similarity, and a word segmentation algorithm using a statistical language model. However, items such as numbers, acronyms, and transliterated foreign words cannot be extracted properly using his method, which differs from ours, because our method can correct words universally as long as they appear in the source document. Lasko et al. [6] suggested five methods for matching words mistranslated by OCR, viz., an adaptation of the cross-correlation algorithm, the generic edit distance algorithm, the edit distance algorithm with a probabilistic substitution matrix, Bayesian analysis, and Bayesian analysis on an actively thinned reference dictionary, and their accuracy rates were compared. They found that the Bayesian algorithm produced the most accurate results. As our interest, we focus on the results of the generic edit distance algorithm. This suggests a minimum edit distance between two words defined as the smallest number of deletions, insertions, and substitutions required to transform either word into the other word. They obtained an overall accuracy of approximately 77.3% for the generic edit distance. Using only this algorithm was inadequate to correct OCR results, as also occurred in our previous study [10].

Current studies related to OCR-error correction tend to use ontology and semantics to address OCR problems [11] [12]. Jobbins et al. [13] developed a system for automatic semantic-relations identification between words using an existing knowledge source, Roget's Thesaurus. The thesaurus contains explicit links between words, including related vocabulary items for each part of speech (e.g., noun and verb), unlike an ordinary dictionary. However, we consider that this previous study might encounter a problem, if dealing with words in a sentence, because it is possible to obtain a real-word error with a word that is also in the same category or cross-reference. To mitigate this shortcoming, it is necessary to use not only the word categories, but also the dependencies of English grammar to obtain a suitable solution, because each word in the sentence will definitely contain at least one dependency on some other word. Zhuang et al. [14] presented an OCR post-processing method based on multiple forms of knowledge, i.e., language knowledge and candidate distance information provided by the OCR engine, using a huge set of Chinese characters as input data.

The input of our system is a collection of biological bar graphs gathered from PubMed. The input must contain at least an X-category, a Y-title, and optionally, a legend. Moreover, we also use related contents of documents (i.e., image captions and corresponding paragraphs) to create our own ontology.

We propose here a novel method of OCR-error correction using edit distance, natural language processing (NLP), and multiple ontologies applied to two-dimensional bar graphs. The edit distance algorithm was used to measure similarities between OCR results and tokens in documents. Moreover, each word is scored to determine its similarity and then collected in a list of individual images ordered by ascending score. The top five words are selected as candidates to be used to replace incorrect OCR results. We designed a structure for our ontology that supports dependency parsing of English text and word categories queried from DBpedia (e.g., [15]). Our objectives in this study were to develop a new OCR-error correction method utilizing ontologies applied to the bar graphs. Our system clearly contributes some benefit to society, particularly in regard to academics, by suggesting a new means of correcting erroneous recognitions that can be adapted to other applications for enhancing their performance.

The remainder of the paper is organized as follows. In Section 2, we present the details of the methodology used in our study. Section 3 evaluates and describes the results, followed by discussion in Section 4. Section 5 concludes and suggests future work.

II. METHODOLOGY

A. Dataset

The dataset used in this study is a collection of two-dimensional bar graphs from journal articles. A bar graph is a chart that represents data grouped in categories by bars with lengths proportional to their corresponding values. Typically, a bar graph in our study has two axes, X and Y. For the Y-axis, the bar graph presents an axis-title as a sentence, a noun phrase, or a single word. In contrast, the X-axis contains several words representing categories, for example, names of medicines or periods of time. In addition, a legend identifies a label for each bar. Extracting characters from the legend is a challenging task, because its position is changeable, depending on the graph space and the author.

 ISBN: 978-1-61208-507-4

B. Ontology Creation

Our ontology is created using captions and corresponding paragraphs from all documents used in this study. We systematically designed an ontology, shown in Fig. 1, that stores word categories gathered from DBpedia, parts of speech (POS), and grammar dependency data extracted using the Stanford dependency parser. It consists of four entity types (i.e., Word, TagCategory, PartOfSpeech, and PartTypeCategory classes), many object properties (e.g., belong_to, has_type, depend_on), and a data property (i.e., full names of POS).

The Word entity represents every individual word tokenized from captions and corresponding paragraphs. The TagCategory entity collects category names of each word in documents, such as mammal, plant, and medicine. Such categories are obtained by querying DBpedia via its SPARQL endpoint; moreover, we also use the Stanford Named Entity Recognizer (Stanford NER) to classify words in sentences into seven classes, i.e., Location, Person, Organization, Money, Percent, Date, and Time. The PartOfSpeech entity provides information about the POS tagging of each word. The total number of individuals is fixed at 36 instances, whose names come from Penn treebank nodes, such as CC, VB, and NNP. The last entity is PartTypeCategory, representing groups of POS taggings. For instance, NNP indicates a singular proper noun belonging to the Noun group.

There are several properties described in our ontology, belong_to, has_type, related_to, same_as, and depend_on, object properties that connect entities to specify their relations. The same_as property represents relations of at least two synonymous words. For example, the word “Japan” appears as JPN, Nihon, and more, which are related by the same_as property. This property is useful for covering words expressing the same concept.

Another crucial property is depend_on, representing dependency relationships between paired words parsed from sentences. We created 67 sub-properties of dependencies used by the Stanford parser, e.g., conj, dep, and nsubj.

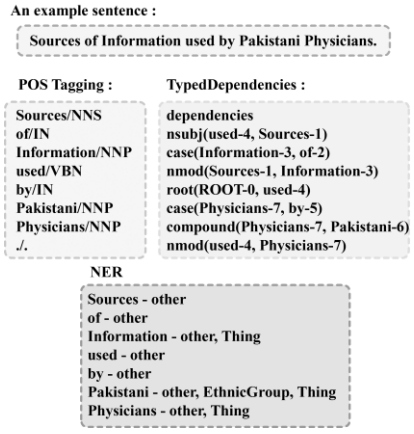

Figure 2. Example of grammar dependency parsing and its results, such as POS tags and typed dependencies, including NER classes queried from DBpedia.

C. Our Proposed Method

In this study, we propose a new method of OCR-error correction combining the edit distance technique and the ontology concept. We divide our system into three major steps: word selection based on edit distance, ontology creation, and OCR-error correction.

1) Word selection based on edit distance: Our input consists of bar graphs that contain an X-category, a Y-title, and optionally, a legend. We use the OCR library to obtain results from the axis descriptions (i.e., X-category and Y-title) and the legend; however, the OCR might produce some recognition errors as a result of unpredictable effects.

The major purpose of this step is to use the edit distance technique to measure the similarity of two words, one of which comes from an OCR result and the other from the caption or paragraphs. The similarity value varies with the distance scale, as shown in (1).

$$Sim(A, B) = 1–(EditDis(A, B)/(L(A) + L(B))) \quad (1)$$

A and B represent two strings. EditDis(A, B) is the edit distance between the strings A and B representing the difference between words. L(A) and L(B) are the lengths of string A and B, respectively. Sim(A, B) is the similarity of strings A and B.

After we make a list of the compared words and their similarities, we sort the records in ascending order of distance. The number of lists is equal to the number of tokens of OCR results. The minimum edit distance score represents the highest similarity between two compared words.

After measuring word similarities, we select only the top five words closest to each OCR result and discard those with smaller similarities. We selected five words as candidates, because this quantity is reasonable in terms of utilization and resource management. For example, in an image, we have a word “well” incorrectly rendered as “woll.” Our system can select candidates ordered by ascending scores, for example, welt, will, wall, well, and more. This example obviously illustrates that if the number of candidates is too small (e.g., one or three), we miss a correct word, “well.” Moreover, if there are too many candidates, more memory space is consumed unnecessarily. Consequently, we obtain lists of similar words corresponding to OCR results.

2) Ontology creation: We construct our ontology following the design procedure in Section II-B. Before building the ontology, we must properly prepare our inputs for storage in our database, including several essential kinds of information regarding bar graphs, such as images’ captions and paragraphs, and axis descriptions extracted using OCR.

 ISBN: 978-1-61208-507-4

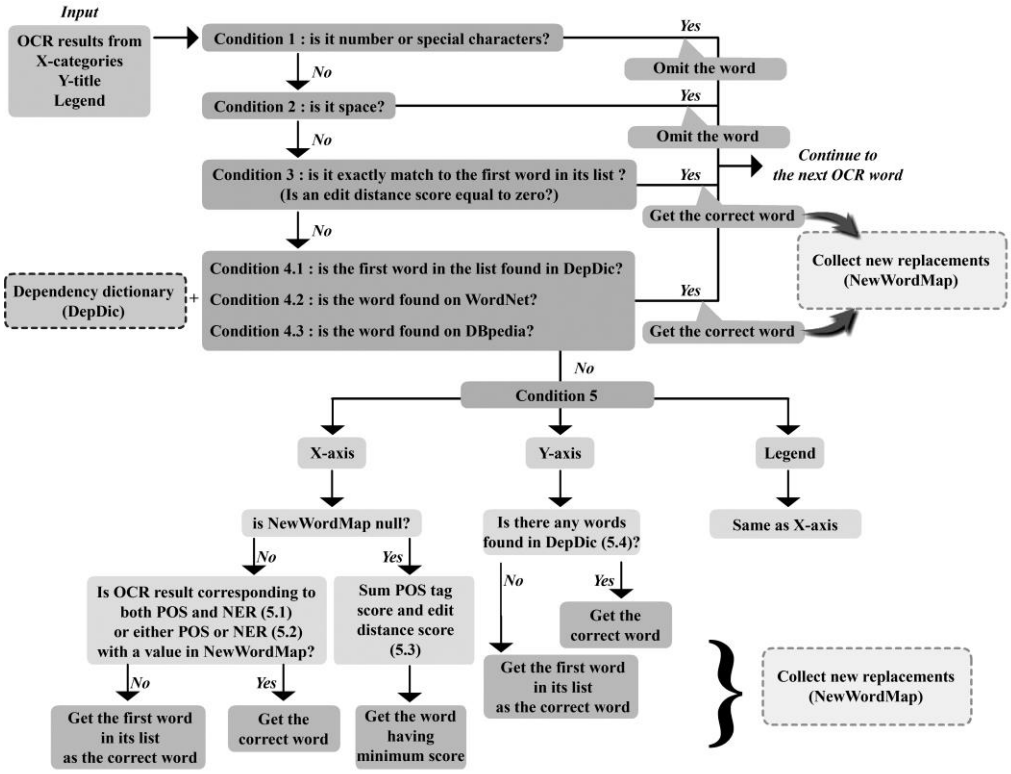

Figure 3. Third step of our method presenting five conditions to correct OCR errors.

We implement a tokenizing program to break the captions and the corresponding paragraphs into tokens. Then, we apply a dependency parser (Stanford parser) to analyze sentences and obtain their NER classes and POS. We separate this step into three minor parts.

First, our system automatically obtains POS tags for each token from the Stanford parser. Second, concurrently, it also obtains the typed-dependency of each pair of words in sentences based on grammar dependency parsing. Fig. 2 shows an example of the dependency parsing process. Third, we endeavor to find the categories that each word belongs to, by querying in DBpedia, all instances of which are represented in the form of triples including the subject, predicate, and object. Here, to acquire the categories, we focus only on the class hierarchies of each token that are queried on the predicate name "rdf:type" or "rdfs:subClassOf." Finally, we obtain our ontology.

3) OCR-error correction: The final step is the core of our system. After we acquire lists of similar words and our ontology from previous steps, we are ready to correct the error recognition.

Initially, we begin to create a mapping dictionary, called DepDic. This records the chain dependencies of the tokens contained in the axis description or the legend. In each image, we can create this mapping if we have at least one OCR result exactly matching the first word in its own list. We use this as the head of the dependency chains. For example, a Y-title contains a word "Information" that also appears in the example sentence. Suppose OCR correctly recognizes it. After following links of dependency relationships, we can obtain a dependency chain of "Information" that includes "Sources," "of," "used," "Physicians," "Pakistani," and "by."

To cover all possible situations for correcting errors, we divide our processes into five core conditions (Fig. 3).

The first condition is whether the OCR result is numeric. In general, the graph component descriptions must be described by alphabet letters, rather than in numerical terms. Numerical representation is inappropriate for our study, because we use the axis descriptions and a legend, which are mostly expressed in letters; on the other hand, numerical terms often appear as measurements. Moreover, we eliminate escape characters in sentences that interfere. It can be troublesome, if a sentence contains escape characters (such as /, -, <, >, and *), because they are reserved characters of SPARQL. If any OCR result contains such characters, our system ignores it.

The next condition is whether the OCR result contains only spaces. We omit it, because we cannot obtain information from it.

The third condition is whether the OCR result finds an exact match in a list. Our system examines the similarity between the OCR result and the first word of its list whose similarity is maximal. If the distance score is equal to zero, the paired words are identical. Hence, we do not need a replacement, because the OCR result is accurate. Further, we

 ISBN: 978-1-61208-507-4

collect it into NewWordMap, which is used to store the OCR results and their new replacements.

The basic idea of our study is that in a graph image, a component description must correspondingly appear at its own caption or referred paragraph(s), since the OCR result, which is extracted from the component descriptions, is expected to have found a matched token in the caption or paragraph(s). However, the description might not appear anywhere in the item of literature, even in the caption or paragraph(s). In this situation, we obtain a list containing words with high distance scores that becomes an obstacle for our correcting system. The fourth condition has been proposed as a solution for this case.

Condition 4 is whether OCR provides correct results that match nothing in the caption or paragraph(s). In this condition, we designed three further sub-conditions. First is whether the first word of the list is matched in DepDic (Condition 4.1). If the matched word has been found, our system suggests using it as a new replacement, because it not only has the smallest distance score but is also collected in the same dependency chain. Otherwise, our system moves to the second sub-condition (Condition 4.2), whether the OCR result actually exists, by querying WordNet. If the system receives a return value from the SPARQL endpoint, this vocabulary exists exactly and can be used itself as the new replacement. Then, it is recorded in NewWordMap. For this sub-condition, its list of similar words is not used. The process of the third sub-condition (Condition 4.3) is similar to the second, but it differs in using DBpedia instead of WordNet. In general, we apply these conditions in the order 4.1, 4.2, 4.3. However, the order of conditions is changed in the case in which the distance score exceeds a threshold, following Conditions 4.2, 4.3, 4.1.

In Condition 5, ideas for correcting the OCR results are separated depending on the types of graph components. For the X-axis, we introduce a method for extracting the X-category based on the generality of bar graphs. Each word in the description of the X-axis must be classified into the corresponding category. For example, a graph might present some descriptions in the X-axis as follows: Suc, Fru, Glc, Gol, Raf, and Sta. After querying DBpedia, we acknowledge that these are names of soluble sugars and have the same POS tags, which are nouns. Based on this method, we obtain the correct OCR results from the X-axis. Initially, the system checks whether NewWordMap is available. Condition 5.1 is satisfied if it is not null, hence we select one of the replacements already stored in NewWordMap to find its POS tags and NER class by querying our ontology. Simultaneously, considering the current OCR result, we also query the POS tag and NER class of the first word in its list with our ontology. If the POS tags and NER class for them are consistent, the first word of the list is taken as the new replacement. Condition 5.2 is an extended version of Condition 5.1. If either the POS tag or NER class is matched, we also flexibly accept the first word of the list as the new replacement.

In contrast, Condition 5.3 checks whether NewWordMap is unavailable or null, hence we compute new scores based on both edit distance score and POS tags for all elements in the list. We assign scores to the POS tags to order their priorities for choosing the new replacement based on our experience of the tags' appearance on the X-axis. The tagging scores are assigned as follows: noun (score = 0), adjective (score = 1), verb (score = 2), article (score = 3), adverb (score = 4), preposition (score = 5), conjunction (score = 6), interjection (score = 7), others (score = 8), and number (score = 9). Nouns are assigned as minimum score, because descriptions of X-categories are mostly nouns. The new replacement of the OCR result is to be the word in the list that contains the lowest score. Note that the minimum score is typically assigned to the noun with the least distance.

Regarding the Y-axis, Condition 5.4 is satisfied if the OCR result is found in DepDic. The idea differs from that of the X-axis, because it contains a description as a title, not a group of words. Commonly, a description of a Y-title often appears as a sentence, a noun phrase, or a single word. Each token in a title must be connected in a chain of dependency; thus, using DepDic is an appropriate idea for selecting the most similar word in the list as a new replacement. Correcting OCR results located at the legend resembles the process at the X-axis. Moreover, as described above, Condition 4.1 also uses DepDic, which is similar to Condition 5.4. However, Condition 4.1 uses only the first word of the list to search in DepDic, whereas Condition 5.4 uses words in the list to explore DepDic until a match is retrieved. Whole words in the list are the top five with the closest distance to the OCR result; therefore, it might be necessary to use every word in the list to find candidates to be a new replacement.

In addition to the cases mentioned above, the OCR result can also be replaced by the first word of the list because of its lowest edit distance score.

III. EXPERIMENTAL RESULTS

We conducted an experiment to compare performance differences between the method used in our previous study (Setting 1) [10] and the method proposed in this study (Setting 2). In the previous study, we proposed a method for correcting OCR results only using the edit distance technique.

After running both systems, we obtained a total of 1,112 OCR tokens from 100 bar graphs. We evaluated both settings by verifying the differences between the OCR results and their new replacements through comparison with actual words showing in the graphs. Setting 1 was tested using the edit distance method based on the previous study, while our method was tested and shown in Setting 2.

We compared accuracies from both settings, as presented in Fig. 4. Our method provided a higher accuracy rate, reaching 81%, and also produced an improvement over the previous method's 54%. Moreover, the precision rate of Setting 2 is approximately 81%.

Fig. 5 presents the accuracy rates of each condition implemented in our method. Moreover, the proportion of correct and incorrect replacements for each condition is also presented there. There were conditions in which the number of correct replacements was greater than the number of incorrect ones, i.e., Conditions 1 or 2, Condition 3, Condition

 ISBN: 978-1-61208-507-4

4.1, Condition 4.2 and Condition 4.3, which attained accuracies of 86%, 97%, 85%, 62%, and 55%, respectively. The highest accuracy was attained in Condition 3, with Condition 5.4 attaining the lowest accuracy.

In addition, we examined the significant differences between these two settings. We observed that our outputs were of the nominal type, classified as "Wrong" or "Correct." We collected the results from both settings and tested them using McNemar's test. This is a statistical test used on paired nominal data to examine a change between two different sets of data that are obtained from before and after treatment. We calculated a two-tailed probability value (*p*), which we used to decide to accept or reject a null hypothesis. It was less than 0.0001. A small *p* value indicates a significant difference.

IV. DISCUSSION

This paper presents a solution for OCR-error correction based on multiple ontologies, NLP, and edit distance. The focus is to develop a system that can effectively correct OCR's errors and enhance its performance (i.e., accuracy and precision rates) over traditional methods. Moreover, our method is not limited only to biology, but is available also for use with other domains, as long as there is a related ontology to apply it with. In this study, we evaluated our method by comparison with the method in our previous study, in which we used only the edit distance to correct OCR results. We applied these two systems to the same dataset containing 100 images of bar graphs and 1,112 OCR tokens.

Reviewing the accuracy rates of Settings 1 and 2, we see that the second setting provided better performance than the first for two reasons. First, our method potentially classifies irrelevant OCR results, which are not to be recorded in NewWordMap. For example, some tokens are meaningless, because they do not come from descriptions of the graph but from other sources, e.g., a part of a bar. OCR can misleadingly recognize such tokens as characters, such as "l-l" or "III." Our method used Conditions 1 and 2 to detect this case, differently from the method of Setting 1, which cannot distinguish relevant from irrelevant characters. This is a shortcoming of Setting 1, which causes many recognition errors. Second, the method of Setting 1 is limited to using the least distance. It can provide an incorrect replacement, because the lowest score represents only a similar word, except for the case of a distance score equal to zero. On the other hand, our system applied many techniques to overcome the OCR difficulty. In addition to using the edit distance to find a list of similar words for each OCR result, we also used ontologies to discover the most suitable replacements for correcting OCR errors.

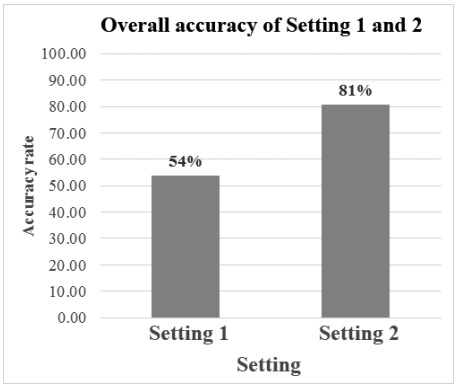

Figure 4. Overall accuracy of Settings 1 and 2.

In other respects, we analyzed some errors that occurred during the experiment and discovered two possible causes. First, some axis descriptions are originally compound nouns. When OCR was used to process the graphs to extract the descriptions, it independently separated them into tokens. On the other hand, to extract words from captions and paragraphs, we used the dependency parser to handle compound nouns. Thus, when our system compared a similarity between OCR results and tokens from captions or paragraphs, we might not be able to find a match. For example, the word "part-of-speech" was a compound noun. OCR divided this word into three independent words, i.e., "part," "of," and "speech." Simultaneously, the dependency parser extracted the caption and obtained this word "part-of-speech" without separation. Hence, we could not find a match between the separated and non-separated words. Second, some OCR results were not mentioned anywhere in a caption or in paragraphs in the text body. We consider that there are two reasons why the words found in a bar graph are not mentioned in its caption or in the paragraphs. First, a token described in axis descriptions is either too general or completely explains itself. For example, if a graph appearing in a biological journal contains some sugar names on the X-axis, an author who is familiar with biology might find these words too general for other researchers who work in related areas; hence, he or she omits explanations in the paper. Second, the extracted token might not be definitely related to the study.

Condition 1 and 2 were very useful due to reduced number of errors by discarding irrelevant OCR results. These

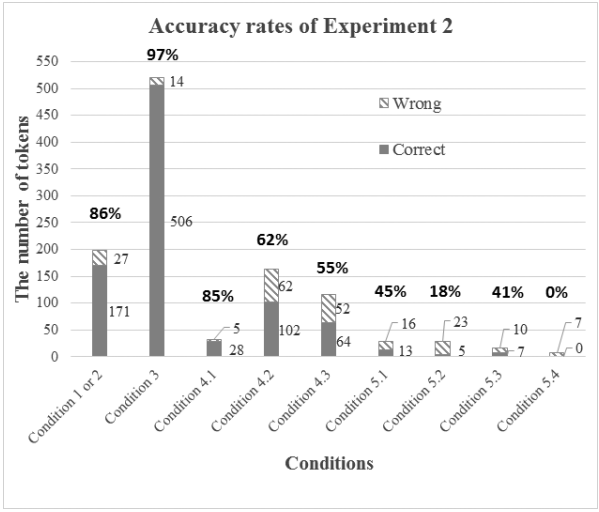

Figure 5. Illustration of accuracy rates in Setting 2 and the proportion of correct and incorrect replacements of each condition.

conditions are also a main reason that makes our method much better than the previous method. They accurately detect irrelevant characters and provide good accuracy rate, 86%.

For Condition 3, we obtained the best accuracy (97%), because OCR was an effective application; moreover, we prepared the input data efficiently. At the beginning, we collected the bar graphs and cleaned them by decreasing noise, omitting irrelevant parts, and increasing sharpness. Based on this evidence, we admit that this condition highly impacts our method's performance. However, we know that other conditions also substantially supported our system, because the accuracy rate was reduced to 45%, if our system used only Condition 3.

For Condition 4, we used ontologies and dependency relationships to correct OCR errors. Obviously, Condition 4.1 provided appropriate accuracy. It applied our chain dependency dictionary to find a match by using the first words of the lists. We proved that the viewpoint of using grammar dependencies was acceptable, because we obtained accurate results, 84%. Moreover, we also used ontologies (i.e., WordNet and DBpedia) to overcome the difficulty of OCR errors. We used them in this study, because we needed to confirm that the words existed. Owing to the advantage of ontology, the recognition errors were moderately reduced; furthermore, the average accuracy rate of this condition was approximately 60%.

However, under these conditions, we encountered errors if the length of a word was too short, especially for two or three characters. The short-length words were often represented as prepositions (such as "in," "on," or "at"), conjunctions (such as "so" or "as") and abbreviations (such as CG and NLP). We realized that every sentence regularly contained at least one preposition or conjunction, since in the DepDic, short-length words (such as prepositions) were ordinarily stored. As a consequence, it was easy for a short-length word to be replaced accidentally by an incorrect selection, because candidates (such as prepositions), even incorrect ones, had usually been found in DepDic. For instance, we assume that we have a word "so," and the first word in its list is "hi," as recorded in DepDic. It is clear that these two example words are totally different, but their distance score is only two. In this case, the system assigns the word "hi" as an incorrect replacement for the OCR result, "so." It was essential to reduce the probability to counter the incorrect matching in DepDic, in particular for short-length words. We decided to rearrange the order of conditions based on the distance score and the word length. If the length of a word was greater than five characters, and the distance score was less than three, then the word was processed through Conditions 4.1, 4.2, and 4.3, respectively. Otherwise, we began the process by querying ontologies (Condition 4.2 and 4.3) to confirm the word's existence and then applying DepDic (Condition 4.1). To evaluate this idea, we conducted a minor test of the order of conditions. As the result, the sequence of conditions definitely impacted the accuracy of the system. After rearranging, the accuracy rate increased dramatically from 39% to 59%.

Observing the results of Condition 5, we see that the overall accuracy rate was approximately 31%. We obtained this low accuracy because we could not find a match in the ontologies (WordNet and DBpedia), since it was impossible to acquire a correct word category. Investigation of why the ontologies had not returned any results revealed that the word might have many equivalents or different spellings.

Moreover, we attempted to compare the results of our study with those of another existing approach. The evaluation presented in [14] aimed to compare results obtained from the proposed method and a basic method that created lists of candidates of each character based on distances. After comparing the differences in the experimental results, which proposed method reduced errors better than the basic method by approximately 29%. Similarly, in our study, our method also attained remarkable results that were much improved over the edit distance method. The error reduction was approximately 27%. Based on this finding, the results from our method and the other method were in agreement, because the key idea of using semantics to reduce OCR errors and the obtained results were in agreement.

Regarding the statistical evidence, we conclude that the difference of both settings (i.e., the edit distance method and ours) is considered to be extremely statistically significant, because the two-tailed *p* value is very small.

V. Conclusion and Future Work

A graph can represent data visually, rendering them easy for a human to interpret and understand. However, automatic information extraction obtained from OCR is desirable. In order to acquire information correctly, in this paper, we proposed a novel OCR-error correction method utilizing the concepts of ontology, NLP, and edit distance. We constructed our ontology to support sentence dependencies, POS tagging, and word categories (NER). Moreover, we also used DBpedia and WordNet by querying via their endpoints to obtain useful information. Sentence dependencies were very efficient in handling the difficulty of OCR errors. We created a dictionary based on the dependency relationships. The edit distance is a traditional technique that we also used in the previous study. However, in this study, we used it only for ranking similar words based on distance scores and storing them in a list corresponding to each OCR result. Our objective was to find a suitable solution for correcting OCR errors that would provide better accuracy and precision than the previous method.

As noted above, we evaluated our method by conducting an experiment with two different settings and then comparing the outcomes. Explicitly, our method provided better results than the previous one. Based on the experimental results of this study, Condition 3 clearly provided the highest accuracy rates, definitely improving the overall performance of our method. Without other supportive conditions, it would not likely reach such high accuracy (81%); therefore, the idea of using dependency

 ISBN: 978-1-61208-507-4

relationships and ontologies in Conditions 4 and 5 was very fruitful.

In our future research, we will continue to develop a semantic system based on this method. We will extract significant information from the graph and apply it to available ontologies. Moreover, other types of graphs will also be of concern and will be used in the future as target data.

REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, Nov 1998, pp. 2278–2324.

[2] S. Kataria, W. Browuer, P. Mitra, and C. L. Giles, "Automatic extraction of data points and text blocks from 2-dimensional plots in digital documents." in AAAI, vol. 8, pp. 1169–1174, 2008.

[3] W. Huang, C. L. Tan, and W. K. Leow, "Associating text and graphics for scientific chart understanding," in Document Analysis and Recognition, Proceedings. Eighth International Conference on. IEEE, pp. 580–584, 2005.

[4] D. Chen, J.-M. Odobez, and H. Bourlard, "Text detection and recognition in images and video frames," Pattern Recognition, vol. 37, no. 3, pp. 595–608, 2004.

[5] C.-J. Lin, C.-C. Liu, and H.-H. Chen, "A simple method for Chinese video ocr and its application to question answering," Computational linguistics and Chinese language processing, vol. 6, no. 2, pp. 11–30, 2001.

[6] T. A. Lasko and S. E. Hauser, "Approximate string matching algorithms for limited-vocabulary ocr output correction," in Photonics West 2001-Electronic Imaging. International Society for Optics and Photonics, pp. 232–240, 2000.

[7] X. Tong and D. A. Evans, "A statistical approach to automatic ocr error correction in context," in Proceedings of the fourth workshop on very large corpora, pp. 88–100, 1996.

[8] D. D. Walker, W. B. Lund, and E. K. Ringger, "Evaluating models of latent document semantics in the presence of ocr errors," in Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pp. 240–250, 2010.

[9] M. Nagata, "Japanese ocr error correction using character shape similarity and statistical language model," in Proceedings of the 17th international conference on Computational linguistics-Volume 2. Association for Computational Linguistics, pp. 922–928, 1998.

[10] S. Kanjanawattana and M. Kimura, "A proposal for a method of graph ontology by automatically extracting relationships between captions and X- and y-axis titles," Proceedings of the International Conference on Knowledge Engineering and Ontology Development (KEOD 2015), vol. 2, pp. 231–238, Nov 2015. [Online]. Available from: http://dx.doi.org/10.5220/0005602102310238

[11] M. Jeong, B. Kim, and G. G. Lee, "Semantic-oriented error correction for spoken query processing," in Automatic Speech Recognition and Understanding, ASRU'03. 2003 IEEE Workshop on. IEEE, pp. 156–161, 2003.

[12] Y. Bassil and M. Alwani, "Ocr post-processing error correction algorithm using google online spelling suggestion," arXiv preprint arXiv:1204.0191, 2012.

[13] A. Jobbins, G. Raza, L. Evett, and N. Sherkat, "Postprocessing for ocr: Correcting errors using semantic relations," in LEDAR. Language Engineering for Document Analysis and Recognition, AISB 1996 Workshop, Sussex, England, 1996.

[14] L. Zhuang and X. Zhu, "An ocr post-processing approach based on multi-knowledge," in Knowledge-Based Intelligent Information and Engineering Systems, Springer, pp. 157–157, 2005.

[15] A. Garcia, M. Szomszor, H. Alani, and O. Corcho, "Preliminary results in tag disambiguation using dbpe-dia," 2009.

[16] E. Loggi, F. K. Bihl, C. Cursaro, C. Granieri, S. Galli, L. Brodosi, G. Furlini, M. Bernardi, C. Brander, and P. Andreone, "Virus-specific immune response in hbeag-negative chronic hepatitis b: relationship with clinical profile and hbsag serum levels," PloS one, vol. 8, no. 6, p. e65327, 2013.

 ISBN: 978-1-61208-507-4