International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 5 (2013) pp. 177- 184

Context Based Automated Image Selector for Articles using Text Illustrator

Ravindra Koggalage[1], Savindhi Samaraweera[2]

[1] General Sir John Kotelawala Defence University,
Ratmalana, Sri Lanka
koggalage@yahoo.com

[2] University of Moratuwa,
Katubedda, Sri Lanka
sssamaraweera@yahoo.com

***Abstract*: Mainly journalists are facing the problem of finding suitable images for their articles. The concept of automated text illustrator has been around for a while. These systems take in an article, identify the keywords of the article and output images from an image database based on the keywords. Such systems may not be suitable as it is difficult to identify keywords in an article to identify images that best suit their texts. It is important to consider the meaning of an article and then use keywords based on those to search for matching images. Most of the available systems have given more emphasis on the image retrieval component and less emphasis on the text processing component. Our attempt is to concentrate on the text processing component, so that it can be used by existing text illustrators. We try to identify the correct meaning of the keywords based on the context of the verbs with which they appear. It also identifies the best possible word that represents the identified meaning.**

***Keywords*:** Text illustrator, Image selector, Natural language processing**.**

I. Introduction

According to Susmitha[1] it is estimated that more than sixty percent of the people in the developed world and almost one third of the world population use the web. "Most of these users are nontechnical people who use the web as an easy, low cost, real time and far reaching communication medium. This necessitates an effective mechanism to retrieve required information from this ever growing knowledge bank by the common man"[1]. As with the expansion of the internet and new developments of image capturing devices, number of images available in the internet is rapidly expanding. "Digital tools are now ubiquitous in the United States, both at home and at school. Digital images and videos are cheaper to generate than the old tape and film-based systems. Most current classroom computers come with digital photo and video manipulation tools"[2]. Search for matching images one by one manually is not practicable anymore. Image search engines are indeed very handy systems that authors could use to select images that could accompany their news articles and stories, but still they need to identify the keywords. This again, should not be an issue as the author of an article, should know the possible keywords that should be associated with the article. Still, it would be further remarkable if an automated system exists, which can process writers' written stories and be able to identify the keywords of the stories and then search in an image database for a set of appropriate images, based on the extracted key concepts. In journalism it is important that writers accompany their articles with images that depict the substance of the text. To find images normally they would use image search engines. However, it is not only a time consuming and tedious task, but also with the risk of losing best matching images. The solution is an automatic text illustrator, which can read the article and search images automatically.

Figure 1 depicts the functional flow of an automated text illustrator. The basic flow includes two main components: processing of the article and image pool retrieval. The first component extracts meaningful words from the article, which represent the substance of the article. The second component extracts images from an image database. Our system concentrates on the text processing capabilities of the system. SPE [3] and TTP [4] are two examples under this topic. Most of the text illustrator systems have fallen short of the text processing component. Therefore, our attempt is to improve the text processing component with the existing image retrieval techniques that would give significant results for the existing automated text illustrator systems.

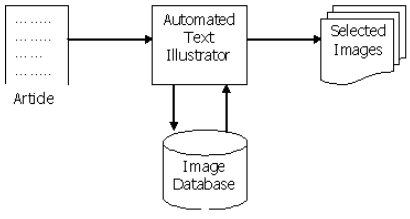

Figure 1. Functional flow of an automated text illustrator

Common mechanism of Information Retrieval systems is to

consider only the keywords that are being typed in. Some keywords have a number of senses, and could be misleading. For example, 'case' has multiple senses – a container, an instance, etc. To differentiate and understand the meanings, humans normally consider the whole or part of a sentence. Another important concept in understanding the meaning of a word is to identify the action context in which the word appears. For example consider 'play cricket' and 'jump cricket'. It is quite evident that in these two phrases the word 'cricket' conveys two completely different meanings. In both these cases humans are able to differentiate the meanings based on the action involved. Therefore, we propose to enhance the text processing component of automated text illustrator systems, by tagging correct senses to the keywords that are identified. For this we propose to consider the whole or part of sentences and to identify nouns and verbs of sentences.

Automating a text illustrator would be a cumbersome task, as it covers a number of subject areas, such as natural language processing, information retrieval, image processing, etc. Each of these areas contains different types of algorithms that can be used efficiently in different types of scenarios. Therefore, an in-depth analysis will need to be performed to identify the appropriate techniques that could be used in different components. As accentuated by the SPE system [3] and by Barnard [5] it is important to analyse both the document and images to extract common features in order to retrieve a set of images that are appropriate for the article. Each of the components will handle different areas separately and finally these components need to be integrated.

Currently, quite a few researches have been carried out under automated text illustrator systems. Research can be divided into two categories:

1. Text-to-Picture
2. Text-to-Scene

Text-to-picture systems are systems that select images relevant to a given text and text-to-scene systems are systems that deal with animation generation for a given text. In both these categories the task of the text processing component would be identical as both types of systems need to identify the keywords of a given text. The difference of these two categories would be the amount of information that is retrieved from a text and how they produce the output of the images, that is either as a set of still images or as an animated scene. A number of research studies have been carried out in the animation generation area. SPRINT [6], WordsEye [7] and CarSim [8] are a few examples of animation generation systems. Comparatively, limited research has been carried out on image selection systems. SPE [3] and TTP [4] are two systems in this category. Most of these systems have fallen short of the basic expectations of an automated text illustrator, where some systems do not perform the text processing component adequately. Another shortcoming in some of the systems is that detailed information such as height, length, how components are placed next to each other, etc. needs to be given. For example in the WordsEye system [7] you need to give the measurements of the objects, such as "the wall is 7 feet tall", which removes the icing of the cake.

The hardest level in an NLP model would be the semantic analysis level, which relies on knowing the meaning of individual words, how the meaning of individual words combine to form the meaning of a group of words and how they all fit in with the meaning of the sentence. This level includes word sense disambiguation (WSD). In order to solve issues in the semantic level, linguists consider the semantic relations between words. WordNet [9] is a research project, which attempts to model a lexical reference system. The system is a manually made database of lexical semantic relations and can be used as a dictionary, a lexical reference system, etc., with relationships, such as synonymy, hypernymy, etc. WSD is the process of determining the most relevant sense that applies to a word depending on the instance the word has been used. WordNet is extensively used in research related to WSD, because of its variety of detailed lexical relations.

II. Methodology

Figure 2 depicts the data flow of the overall system[10]. The article is first fed to the system, which in turn forwards the article to the external keyword extractor, to identify the initial pool of keywords. These identified keywords are processed by the WSD component to identify the correct meanings of the keywords. For this we need to use the verb identifier to identify the verb of the concerned keyword. A semantic calculation is performed in the WSD component in order to identify the most probable meaning for the keyword. Then the lemma identifier identifies the probable synonymous word that should be used, which best represents the identified meaning. The lemma identifiers will constitute the final set of keywords, which is forwarded to the image database in order to retrieve images.

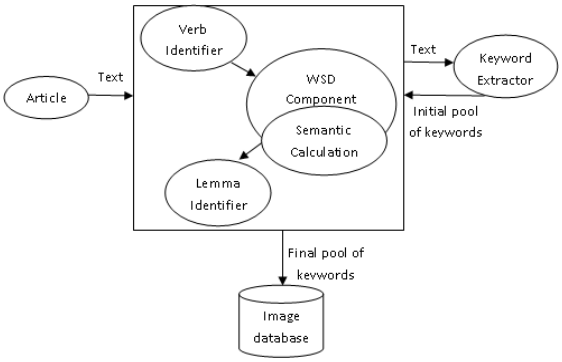

Figure 2. Data flow of the system

In this implementation we have considered words as elements of a sentence. We can say that a verb of a sentence plays a prominent role when trying to identify the correct sense of a word. Further, we can narrow down this idea to state that the meaning of a noun can be identified based on the verb

associated with the noun. Therefore, to disambiguate the meaning of a given noun, we use the main verb of its sentence. As the system would be processing (noun, verb) pairs we need to make sure that all keywords are paired up with appropriate verbs. Therefore, we have introduced the verb identifier component, which tries to identify the verb in several phases. The first phase is to identify verbs from the identified keywords. Some of the keywords that are returned by the keyword extractor system contain phrases. For example the keyword 'shaved head' is used for the WSD component as (head, shave). Therefore, we check in the returned keyword set to find out whether there are any verbs that accompany the noun. If such pairs are found out, then we use them as the (noun, verb) pair for the WSD component. For the keywords that such verbs do not exist, we use collocations to identify verbs. A collocation is a sequence of words that occur together unusually often. For example 'take care' is a collocation, whereas 'do care' is not. To identify collocations we use Brown corpus. In these collocation entries we check whether the keyword exists, and if so, we check whether it exists with a verb, and check whether this verb exists in the sentence. For example, the phrase 'unwed mother' can be found. If the word 'unwed' exists in the sentence with the word 'mother', then we could consider (mother, unwed) pair for the WSD component. The third phase of identifying verbs is to consider verbs from the sentence context. We post tag all the words in the sentences that contain the keyword. Then based on basic linguistic rules we try to identify the main verb of the sentences. If we cannot still identify a suitable verb for the keyword, then from the pre-compiled file of (noun, verb), we identify the verbs that could exist with the keyword, and then validate whether it exist in the sentence. If so, then we use those (noun, verb) pairs for the WSD component.

The WSD component is based on [11] and is processed under four phases. The first phase is to identify if the noun in the (noun, verb) pair has only one single meaning. If so, that word will always hold one meaning, whatever the context of the rest of the sentence means. If the relevant noun contains multiple senses, then we need to evaluate which sense best suites the current context of the word. Therefore, in phase two we try to identify whether the noun contains a semantic similarity with another candidate noun in the noun's verb context. If a semantic similarity is identified between the noun and the candidate noun that exists with the verb, then we consider the meaning of that noun exists with the current verb context. To identify the candidate nouns that can exist with the verb we use a pre-compiled (noun, verb) file. For each candidate noun identified in the file, we check whether a semantic similarity exists between the noun and the candidate noun. If so, we interpret the considered noun's synset as a candidate synset. In phase three, we check whether any candidate verb exists within the noun context, and then check whether there is a semantic similarity with the verb and the candidate verb. To identify candidate verbs we use the pre-compiled file. After identifying that a similarity exists with the candidate verb, we process the (noun, candidate_verb) pair. If a resultant is found by processing (noun, candidate_verb) pair, the synset of the noun is considered as a candidate synset. Still, if the system is unable to find a correct sense for the noun, in phase four it will identify a separate (candidate_noun, candidate_verb) pair and check whether a semantic similarity exists between the verb and the candidate verb as well as the noun and the candidate noun. In this method we first check whether a semantic similarity exists with the noun and the candidate noun. If so, we then check whether a similarity exists between the verb and the candidate verb. If so, we identify the resultant synset between the noun and the candidate noun as to be the correct synset.

The system considers four semantic relations out of the large amount of semantic and lexical relations that exist in WordNet. Synonymy, hypernymy, hyponymy and coordinate relationships are considered. Synonyms were chosen as this would be the basic and obvious semantic similarity between two words. Hypernym and hyponym relationships were chosen as the parent and child synsets will denote a generalised and specialized concept of the concerned synset. Coordinate relationships were considered because, as peer synsets would have meaningful attributes of the concerned synset.

The WSD algorithm can retrieve multiple synsets within a phase. We need to identify only one synset, which would represent the most probable meaning of a word. To solve this issue, we extended the system into a similarity calculation component. We have experimented with different types of similarity calculation measures in order to choose the best calculation method that suits the system. In phases two to four we would be calculating the semantic relationship of hypernymy, hyponymy and coordinate relationship. We consider the noun synset and the candidate noun synset and perform the similarity measures. Based on the retrieved similarity measures we identify the maximum similarity measure that was retrieved and consider the synset with that measure to be the synset which correctly represents the meaning of the noun. If we encounter multiple synsets with the same maximum value then the summation of the values relevant synsets are considered in order to identify the maximum value that is being returned. We have employed four similarity calculation measurements: path distance similarity, Leacock Chodorow similarity [12], Resnik similarity [13] and Jiang-Conrath similarity [14]. All these similarity calculation methods use synset pairs in order to perform the calculations. Path distance similarity measure returns a score denoting how similar two word synsets are based on the shortest path that connects the senses in the hierarchy. The path length is measured in nodes. If multiple paths exist between the two synsets then the shortest path is selected. Leacock Chodorow similarity measure returns a score denoting how similar two synsets are, based on the shortest path that connects the senses and the maximum depth of the hierarchy in which the synsets occur. Maximum depth of the hierarchy is the longest distance between the root and any leaf of the concerned hierarchy. Resnik presents a new measure of semantic similarity in a hierarchy, based on the notion of information content. This returns a score denoting how similar two word senses are, based on the information content of the least common subsumer that is the most specific ancestor node for the synsets

that are considered in the similarity calculation. Jiang-Conrath similarity measure returns a score denoting how similar two word senses are, again based on the information content of the least common subsumer and the two input synsets.

In the lemma identifier module we identify the commonest or the most used word that describes a particular sense. This was considered as a needed component because some words are not commonly used in the English language, and as we are using these words in order to search an image database it would be appropriate if we could identify the commonest word that is used to denote the meaning. For example, consider the 'bass', which contains a fish sense. There would hardly be images that are annotated by the 'bass' that depicts fish, but these images would definitely be annotated with the 'fish'. Therefore, we believe it is important to have a component to identify the commonest used word for a meaning. To identify the commonest used word we use the frequency distribution of the words in Brown corpus. All the lemma names or simply synonyms, in the identified synset are extracted. Then using the constructed frequency distribution we identify the frequency of each lemma name and select the lemma name with the highest frequency.

III. Evaluation

We have disintegrated our evaluation into smaller components in order to identify strengths and weaknesses of each component. Therefore, our first experiment was based on the WSD component. The functionality of the WSD component is to identify the correct meaning of a noun for a given verb context in a (noun, verb) pair. The input of the component would be a (noun, verb) pair and the output would be a synset denoting the meaning of the noun for the given verb context. For each identified synset, we have calculated the outcome based on similarity calculation measures in order to identify the most probable synset, if multiple synset outputs exist from the WSD component. We have used four similarity calculations: path distance similarity, Leacock Chodorow similarity, Resnik similarity and Jiang-Conrath similarity.

According to Resnik [13], the shorter the path from one node to another, the more similar they are. The path length is measured in nodes. The value is in the range of one and zero, and if no path exists between the synsets then null is returned. If multiple paths exist between the two synsets then the shortest path is selected. Path distance similarity measure is calculated as in equation (1).

$$Similarity(S_1, S_2) = \frac{1}{(ShortestPath(S_1, S_2) + 1)} \quad (1)$$

Leacock Chodorow similarity measure returns a score denoting how similar two synsets are, based on the shortest path that connects the senses and the maximum depth of the hierarchy in which the synsets occur [12]. Maximum depth of the hierarchy is the longest distance between the root and any leaf of the concerned hierarchy. One requirement needs to be satisfied when this similarity measurement is used that is the POS tags of both synsets need to be similar. The values are greater than zero, and if no path is found between them null is returned. Calculation for Leacock Chodorow similarity measure is shown in equation (2).

$$Similarity(S_1, S_2) = -\log\left(\frac{(ShortestPath(S_1, S_2) + 1)}{2 * depth}\right) \quad (2)$$

Resnik in his paper [13] presents a new measure of semantic similarity in a hierarchy, based on the notion of information content (IC). In normal context information content would be the frequency distribution of words, but in our calculation we consider the information content of the synsets. The Resnik similarity measure returns a score denoting how similar two word senses are, based on the information content of the least common subsumer (lcs) that is the most specific ancestor node for the synsets that are considered in the similarity calculation. Wordnet consist of pre-calculated information content files for corpuses such as Brown Corpus, Semcor, etc. Resnik similarity measure is shown in equation (3).

$$Similarity(S_1, S_2) = IC(ICS(S_1, S_2)) \quad (3)$$

Jiang-Conrath similarity measure [14] returns a score denoting how similar two word senses are, again based on the information content (IC) of the least common subsumer (lcs) and the two input synsets. Equation (4) denotes the calculation for Jiang-Conrath similarity measure.

$$Similarity(S_1, S_2) = \frac{1}{IC(S_1) + IC(S_2) - 2 * IC(ICS(S_1, S_2))} \quad (4)$$

Out of these similarity measures Jiang-Conrath similarity proved to be the most successful, as shown in table 1. The results show the number of examples that were able to identify the correct meaning of the noun based on the verb, how many were identified incorrectly and how many examples were not identified.

If we consider the overall test results they show that all four phases have been utilised to identify the results. About 60% of the results had been achieved by phase 2, which identifies the semantic relationship between a noun and a candidate noun. From the test results, 28% of the experiments had given incorrect senses. This is mainly due to identifying incorrect (noun, verb) pairs. According to the results most of the verbs were identified by the pre-compiled file with (noun, verb) pairs. Compilation of the file was not done manually, but done programmatically based on basic English grammar. During this compilation incorrect (noun, verb) pairs might have got entered, and due to this reason the system may not function as expected. Therefore, to minimise the errors that occur due to this reason, it would be better to manually go through the set of identified (noun, verb) pairs and validate the entries. Another possible solution for this would be to have separate (noun, verb) pair based on domains, such as religion, sports, etc. Then each (noun, verb) file will only contain verbs that are relevant

to that domain, which would reduce the probability of erroneous sense identification.

We have disintegrated our evaluation into smaller components in order to identify strengths and weaknesses of each component. Therefore, our first experiment was based on the WSD component.

As described in the methodology section, the functionality of the WSD component is to identify the correct meaning of a noun for a given verb context in a (noun, verb) pair. The input of the component would be a (noun, verb) pair and the output would be a synset denoting the meaning of the noun for the given verb context. Table 1 contains some sample inputs and the relevant synset output identified by the WSD component. The third column indicates whether the identified synset is the correct synset or not. For example ('studio', 'walk') pair gave Synset('studio.n.01') as the output. The definition of Synset('studio.n.01') is "workplace for the teaching or practice of an art", which is the correct meaning as expected. Therefore, the accuracy value should be "1". If we consider ('picture', 'paint') pair the identified output was Synset('movie.n.01'). The definition for Synset('movie.n.01') is "a form of entertainment that enacts a story by sound and a sequence of images giving the illusion of continuous movement", which is not correct when considering the verb "paint". The correct output should have been Synset('picture.n.01'). Therefore, the accuracy value would be "0". The evaluation was conducted based on randomly identified (noun, verb) pairs. We used hundred (noun, verb) pairs for this experiment.

Table 1. Sample input and relevant output values of the WSD component

Input: (noun, verb) pair	Output: Synset value	Accuracy
('studio', 'walk')	studio.n.01	1
('art', 'paint')	art.n.02	1
('award', 'receive')	prize.n.01	1
('photo', 'take')	photograph.n.01	1
('bird', 'photograph')	dame.n.01	0
('picture', 'paint')	movie.n.01	0

For each identified synset, we calculated the outcome based on similarity calculation measures in order to identify the most probable synset, if multiple synset outputs existed from the WSD component. We have used four similarity calculations as explained earlier with calculation equations and compared the output. Table 1, table 2, table 3 and table 4 respectively show the outputs of the WSD component based on similarity measures of path distance similarity measure, Leacock Chodorow similarity measure, Resnik similarity measure and Jiang-Conrath similarity measure. The tables depict results of hundred (noun, verb) pairs that were used to test the WSD component based on the similarity measures that were used. The results show the number of examples in which the correct meaning of the noun was identified based on the verb, how many were identified incorrectly and how many examples were not identified.

Table 2. Test results of path distance similarity measure

	No. of (noun, verb) pairs
Correct senses	65
Incorrect senses	29
Non identified words	6

As you could see from the output there is not much of a difference between the outputs of the four similarity measures. If we compare the test results of table 2 and table 3, the results are identical. If we closely look at the calculations that are performed in these two similarity measures, which we described in the methodology section, we will notice that there should not be any difference in the test results, where distance path similarity measure uses the shortest path between the two synsets and Leacock Chodorow similarity measure uses the shortest path as well as the depth of the hierarchy in which the synsets occur. The depth of the hierarchy would anyway be the same because we are considering the same pair of synsets. Therefore, it is inevitable that both similarity measures returned the same results.

Table 3. Test results of Leacock Chodorow similarity measure

	No. of (noun, verb) pairs
Correct senses	65
Incorrect senses	29
Non identified words	6

Table 4. Test results of Resnik similarity measure

	No. of (noun, verb) pairs
Correct senses	65
Incorrect senses	29
Non identified words	6

If we look at the output of the Resnik similarity measure shown in table 4, the results look the same, but actually the output was different, where some words had correct meanings and some other words had incorrect meanings. This would be the resultant of using the information content of the synsets' least common subsumer. We used the information content of Brown corpus. Had we used another set of information content then the result would have inevitably been different.

The results of Jiang-Conrath similarity measure performed far better than the rest of the similarity measure. As seen in table 5, 72% of the words were given correct meanings, compared to the 65% of the other similarity measures. This is because the

information content of both synsets, as well as the least common subsumer is considered. Therefore, we chose to use Jiang-Conrath similarity measure for the rest of the evaluations of the system.

Table 5. Test results of Jiang-Conrath similarity measure

	No. of (noun, verb) pairs
Correct senses	72
Incorrect senses	22
Non identified words	6

If we consider the overall test results they show that all four phases have been utilised to identify the results. About 60% of the results had been contributed by phase 2, which identifies the semantic relationship between the noun and the candidate noun. Therefore, we could consider that the pre-compiled (noun, verb) file contains useful entries. About 5% of the wordings contain only one sense.

From the test results, 28% of the experiments had given incorrect senses. This is mainly due to the identification of incorrect (noun, verb) pairs. In the methodology section, we described that in order to identify candidate verbs, we use the verb identifier component. According to the results most of the verbs were identified by the pre-compiled file with (noun, verb) pairs. Compilation of the file was not done manually, but done programmatically based on basic English grammar. During this compilation, incorrect (noun, verb) pairs might have got entered and due to this reason the system may not have functioned as expected. For example while inspecting the pre-compiled file we came across the pair (‘grams’, ‘fee’). According to WordNet the word “fee” does not contain a verb sense, but according to Penn Treebank the word “fee” is considered as a verb. Therefore, inconsistencies between corpuses also lead to issues in the system. Another pair we came across was (‘han’ ’, ‘hold’). The pre-compiled (noun, verb) pair was composed based on Brown corpus, which contains articles that were written in 1960s. The word “han’ ” could have meant something in the 60s, but it does not mean anything today, or may be it could have been just a plain error of the text. Therefore, it is inevitable that if the (noun, verb) pairs in the pre-compiled file are inaccurate then the WSD algorithm is also going to produce inaccurate meanings. Therefore, to minimise the errors that could occur due to this reason, it would be better to manually go through the set of identified (noun, verb) pairs and validate the entries. Another possible solution for this would be to have separate (noun, verb) pairs based on domains, such as religion, sports, etc. Then each (noun, verb) file will only contain verbs that are relevant to that domain, which would reduce the probability of erroneous sense identification.

From the results of table 2 to table 5, we could see that about 6% of the tests did not return any value. This could be due to non-availability of sufficient (noun, verb) pairs in the pre-compiled file. This means that not even a candidate (noun, verb) pair existed to meet phase 4, described in the methodology section. For example, the pair (‘sewing’, ‘induce’) resulted in not returning any values. To minimise the effects of this issue, we believe it would have been more conclusive if the pre-compiled (noun, verb) pair file had been constructed based on not just one corpus, but on multiple corpuses. Another reason that can be identified is that in our semantic similarity detection, we only tracde one level up or down, but if the tracing had been performed for several levels then we would not have had any null senses in our system. Though, this solution would eliminate the null value issue, still this could produce erroneous results.

The next evaluation was performed for the lemma identifier component. The lemma identifier component was used in order to identify the best possible word that describes the meaning. As described in the methodology section, the synonymous words of the synset are considered based on the frequency distribution of the words and collocations in Brown corpus. Table 6 contains some sample values and the resultant output given by the component during the evaluation. As you could see the first three entries have given incorrect words. The meaning of Synset(‘cooking.n.01’) is best represented by the word “cooking” not by the word “preparation”, as the word “preparation” could have other ambiguous meanings. Synset(‘movie.n.01’) generated the output “picture” instead of “film”, which is incorrect. Table 7 shows results of the lemma identifier component. The whole idea of introducing the lemma identifier component was to find out the most commonly used word that best describes the meaning, based on the frequency distribution of the synonymous words, but still it gives incorrect results of 28%. Some of the results imply that after disambiguating the sense correctly, during the lemma identifier we again ambiguously name the keywords by incorrect wording. For example the Synset(‘book.n.02’) contains synonymous values ‘book’ and ‘volume’. The lemma identifier identifies ‘book’ as the correct wording, but still, if we had used a different corpus there is a possibility of identifying the word ‘volume’ as the lemma name for Synset(‘book.n.02’). We believe the solution for this would be to have different corpuses for different domains.

Table 6. Sample values that were used for the lemma identifier

Input: Synset	**Output: Identified Word**	**Correct Word**
cooking.n.01	preparation	cooking
movie.n.01	picture	film
class.n.06	year	class
prize.n.01	award	
ocean.n.02	sea	
photograph.n.01	picture	

Table 7. Test results of lemma identifier

	No. of words
Correctly identified lemma names	72
Incorrectly identified lemma names	28

Next we performed an evaluation of the overall system(Figure 3). We have taken the output of hundred short articles and evaluated the sense tagging outcome of the system. We have considered the number of keywords that are given by the external system and the number of senses that were correctly tagged. The overall accuracy of the system is 66%.

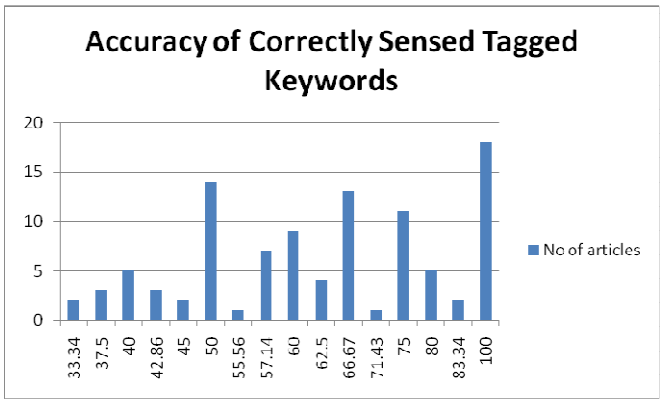

Figure 3. Test results on accuracy of the sense tagged system

Further investigating we found out that most of the articles that had correctly tagged keywords contained about two to four keywords in the initial pool. This could mean that the initial keyword list might have contained unnecessary keywords, such as 'sustainable high quality', 'square miles', etc, were removed by the system as inappropriate keywords.

Another issue that we came across is that during the verb identifier for a noun, correct verbs of the sentences were not identified. The main reason for this is that some of the verbs are contained in the stopwords list. The stopword list contains verbs, such as 'has', 'be', etc, and these verbs do appear as main verbs of sentences. For example, consider the two sentences 'I had a guitar' and 'I am playing a guitar'. The system would correctly identify 'playing' as the main verb of the second sentence, but will be unable to identify 'had' as a main verb, as it has already been removed from the stopword list.

IV. Conclusion and Future Work

According to the evaluation results, the text illustrator system performs well even with some identified shortcomings. Therefore, we believe that the system could be further enhanced in order to give better results. One essential drawback of the system was the pre-compiled file of (noun, verb) pairs. This resulted mainly for the inaccuracies that were identified in the results. The file was generated based on the existing Brown corpus. Therefore, it would be useful if the (noun, verb) pairs were accurate and up to date. A manual validation of the entries of the file would be appropriate. In order to solve the above problem, we could extend our system to extract (noun, verb) pairs from articles according to domains, such as sports, politics, etc, and to save them in different files according to domains. So during the verb identification period, the system would only utilise the file that is relevant to the domain, and identify verbs that are relevant to the domain. We could also use these domain specific files for the lemma identifier to get the correct synonymous word. We believe that using domain specific entries will drastically increase the accuracy of the system. Currently, the system does not handle keyword extraction. It uses an external system to identify keywords of the system. Therefore, we could further extend this system to handle keyword extractions.

References

[1] Susmitha Dey, Siby Abraham, "User Interface For A Search Engine: A Customized and Multi-domain Approach" International Journal of Computer Information Systems and Industrial Management Applications(IJCISM), ISSN 2150-7988 Volume 4, pp. 169-179, 2012.

[2] Olga Werby, "Using Digital Storytelling to Advance Scientific Comprehension and Retention in a Middle School Science Course", International Journal of Computer Information Systems and Industrial Management Applications(IJCISM), ISSN 2150-7988 Volume 4, pp. 018-026, 2012.

[3] Dhiraj Joshi, James Z. Wang and Jia Li "The Story Picturing Engine - A System for Automatic Text Illustration," ACM Transactions on Multimedia Computing, Communications and Applications, 2006.

[4] X. Zhu, A. B. Goldberg, M. Eldawy, C. R. Dyer, and B. Strock, "A Text-to-Picture Synthesis System for Augmenting Communication," Proceedings of the 22nd Conference on Artificial Intelligence: Integrated Intelligence Track, 2007.

[5] K. Barnard and D.A. Forsyth. "Learning the Semantics of Words and Pictures." International Conference of Computer Vision, pp. 408-415, 2001.

[6] A. Yamada, T. Yamamoto, H. Ikeda, T. Nishida and S. Doshita. "Reconstructing Spatial Image from Natural Language Texts." Proceedings of the COLING, vol. 4, pp. 1279-1283, 1992.

[7] B. Coyne and R. Sproat. "WordsEye: An Automatic Text-to-Scene Conversion System." Proceedings of SIGGRAPH, pp. 487-496, 2001.

[8] R. Johansson, A. Berglund, M. Danielsson and P. Nugues. "Automatic Text-to-Scene Conversion in the Traffic Accident Domain." Proceedings of the 19th IJCAI, pp. 1073-1078, 2005.

[9] Miller G.A. "WordNet: An on-line lexical database", International Journal of Lexicography, 3(4): pp. 235-312, 1990.

[10] Savindhi Samaraweera, Ravindra Koggalage, "Automated text Illustrator Based on Keyword Sense Tagging", International Conference on Affective Computing and Intelligent Interaction (ICACII 2012), Taipei, Taiwan, China, Feb 2012.

[11] Li, Xiaobin Szpakowicz, Stan, and Matwin, Stan, "A WordNet-based algorithm for word sense disambiguation." Proceedings, 14th International Joint Conference on Artificial Intelligence, Montreal, August 1995.

[12] A. Budanitsky and G. Hirst, "Semantic Distance in WordNet: An Experimental, Application-Oriented

Evaluation of Five Measures", Proc. Workshop WordNet and Other Lexical Resources, Second Meeting North Am. Chapter Assoc. for Computational Linguistics, June 2001.

[13] Resnik, P., Using information content to evaluate semantic similarity in a taxonomy. In Proceedings of IJCAI-95, pages 448–453, Montreal, Canada, 1995.

[14] Jiang, J., Conrath, D. "Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy", Proceedings of the 10th International Conference on Research on Computational Linguistics, Taiwan, 1997.

Author Biographies

First Author Professor Koggalage was born in Sri Lanka in 1966. He is presently the Deputy Vice Chancellor - Academic of Kotelawala Defence University, Sri Lanka. He also serves as a council member of the University of Moratuwa. He has obtained his PhD from the University of Melbourne in 2004, Australia in Mechatronics Engineering. He got his MEng from the Nangyang Technological University, Singapore from the school of EEE and MSc from the National University of Singapore. He has obtained His BSc (Hons) in Computer Science & Engineering from the University of Moratuwa, Sri Lanka. He has work experience in many countries such as Australia, Japan, Germany, Singapore and Sri Lanka. He was the Chief Technology Officer(CTO) of ANCL and Director-Projects at UOM-Dialog Mobile communication research lab, University of Moratuwa. His research interests include image processing, pattern recognition and classification, communication, artificial intelligence, computer security, data mining, optimization, language processing, management, and mind development.