AMRITA-CEN@FIRE2015: Automated Story Illustration using Word Embedding

Sanjay S.P
Centre for Excellence in Computational Engineering and Networking,
Amrita Vishwa Vidyapeetham
Ettimadai, Coimbatore. India
sanjay.poongs@gmail.com

Nivedhitha Ezhilarasan
Centre for Excellence in Computational Engineering and Networking,
Amrita Vishwa Vidyapeetham
Ettimadai, Coimbatore. India
e.nivedhitha@gmail.com

Anand Kumar M and Soman K P
Centre for Excellence in Computational Engineering and Networking,
Amrita Vishwa Vidyapeetham
Ettimadai, Coimbatore. India
m_anandkumar@cb.amrita.edu

ABSTRACT

Story books are copiously filled with image illustration in which the illustrations are essential to the enjoyment and understanding of the story. Often the photos themselves turn out to be more important than content. In such cases, our principle job is to locate the best pictures to show. Stories composed for kids must be improved with pictures to manage the enthusiasm of a tyke, for words usually can't do a picture justice. This system is built as a part of shared task of Forum of Information Retrieval and Evaluation (FIRE) 2015 workshop. In this system we provide a methodology for automatically illustrating a given Children's story using the Wikipedia ImageCLEF 2010 dataset, with appropriate images for better learning and understanding.

Keywords

Automated Story Illustration; Story Picturing Engine; Image Ranking; word-embedding, WordNet; Machine Learning; TF-IDF; Image Retrieval;

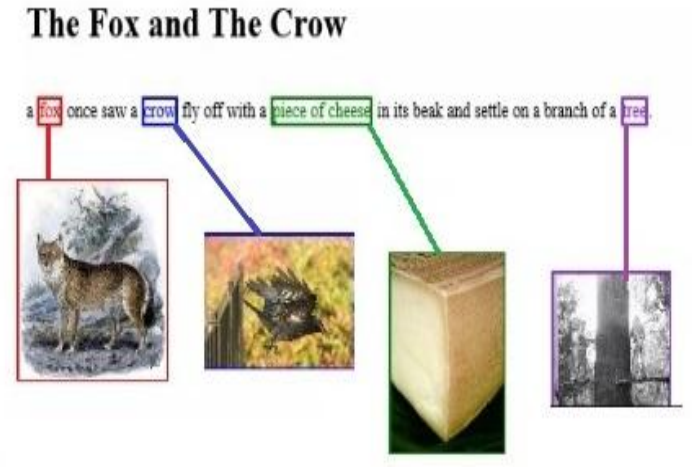

Figure 1: An image generated by our engine

1. INTRODUCTION

A kid is touchy to pictures even before he/she can talk. This is not shocking in the event that we think about that as an infant effectively recognizes its mom's face and outsiders. The kid's mom, sister, sibling and the outsider can all be viewed as living and moving pictures. In the same way, a kid will perceive a most loved toy or pet.Stories are always preferred when they tag along with beautiful images depicting the content. This clearly gives us the importance of giving image Illustration in Children's short stories. But we don't have many systems that can automatically convert any general textual information into pictorial representation. This paper describes our system for FIRE 2015 Automated story Illustration using the Wikipedia ImageCLEF 2010 dataset. This task focuses on automatically illustrating the story with corresponding images thereby making reading and understanding better. One can understand the core content of the story, just by looking at the images.

A sample output (Figure 1) for the story "THE FOX AND THE CROW" is shown. An image corresponding to each entity is produced. Our novel story Illustration system automatically generates a picture that aims to convey the gist of the content of general natural language text.

2. RELATED WORK

Our present work is inspired and triggered by a rich resource work done prior to us. A similar system called story picturing engine, is built using the techniques designed for content-based image retrieval and textual information retrieval [5]. In recent years, learned statistical models have been widely used in linguistic indexing of pictures [7]. Also image annotations can be modeled using latent Dirichlet allocation (LDA) [10].

Here we try to analyze the research issues related to Image illustration mentioned in this section. The task of automatically

generating the words that describe the picture is called Image annotation. Image annotation is used for image search and retrieval applications [6]. Story Illustration, on the contrary, aims in substituting the set of images that best describes the given text. In the proposed work it is the story. So we can call these two problems the inverse of each other. To rank pictures in for a given story, in this system, we have used an unsupervised algorithm. Our system is entirely based on the concept of Word Embedding. Word embedding is a mapping of a word to an n-dimensional vector space. This real valued vector representation captures semantic and syntactic features. Gensim is an open source tool for python which is used for implementing the concept of word embedding. Our ranking scheme based on TFIDF using gensim. We have represented one ranked set of images for one entity. So when the entire story is queried, the list of images that are produced can effectively produce the story line. Other variations and implementation are explained in detail in section 3.

3. SYSTEM ARCHITECHTURE

The task and working methodology is as follows:

In the development phase initially a set of five children's short stories with important entities and events, that needs illustration were already provided. Our system provides one ranked list of images corresponding to each important entity and event in a story. At a later stage a set of 22 children's short stories were given for illustration. We have provided a unique image ranking methodology that effectively computes the importance of each picture and outputs a ranked list of images which aptly describes the story.

3.1 Dataset analysis

In development data set input Query is constructed and annotated with their label by using Python. It contains a set of five short stories. The most important entities and events that effectively summarize the story were already provided which reduced the overhead of finding them. This information serves as the input to query the image database to retrieve the pertinent image. In this task we use the ImageCLEF Wikipedia Image Retrieval 2010 dataset. This dataset consists of 237,434 images along with their captions. Captions are available in English, French and/or German. Complete language statistics, image files and their captions are found in the ImageCLEF website. Metadata are provided as a single metadata.zip archive which is split into 26 directories (from 1 to 26). "metadata/1" contains XML files from 0.xml to 9999.xml, "metadata/2" contains images from10000.xml to 19999.xml etc. We have used only English dataset for developing our system. We extracted important information such as caption, description comments etc., from the database and created a file, which is then queried to obtain the ranked list of images corresponding to each story. The extracted information is stored in a model file. Figure.2 depicts the procedure of how the information from ImageCLEF 2010 is extracted.

The main components in the .xml files are:

<Image>

< name >

<description>

<comment>

<caption>

<comment>

<license>

Given a query, the search methodology retrieves relevant pictures by analyzing the image caption textual descriptions found adjacent to the image, and other text-related factors such as the file name of the image which are already extracted and stored in model file. This extraction process involves converting all the xml files that contains information about the images into a text file named as "Text model of image database". Only necessary information about the image, from the XML file is filtered out.

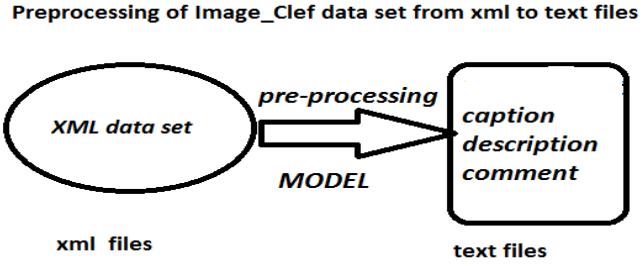

Figure 2: An image depicting our preprocessing ImageCLEF dataset

3.2 Model Description

The input data, story was in XML format and is pre-processed and converted into a simple ".TXT" named STORY.TXT file for fast searching of the data . This contains information about entity, event and the entire story. This is called story entity event block in the figure 3.1 .The whole set of information is passed to a extraction unit where the important key words such as entity and events are extracted .The Image extraction unit will search through the MODEL.TXT files which is also called 'Text model of image data base block'. The extracted information is then stored for processing in the local variable which is referred as Local Data Base. The extracted text in the local data base is then given to the training model of TFIDF from GENSIM. It will create a word2vec features form the documents. These model files are created for each Entity in the story which is called the model Block in the figure 3.1

The text extraction block will join the story, entity and events .The expanded text is now passed to extract the hypernyms which will get the example sentences of the hypernyms while the expanded text is also passed to WSD block which will extract the sense of the entity used in the story and related example text is extracted from WORD NET. The extracted hypernyms and WSD text are added to the extracted text this is called Text Expansion block. The expanded text is now passed as a query to the model file created by the GENSIM which will map the query and then rank all Images according to the TFIDF weights. The weight of the Images will be in 0-1 scale then

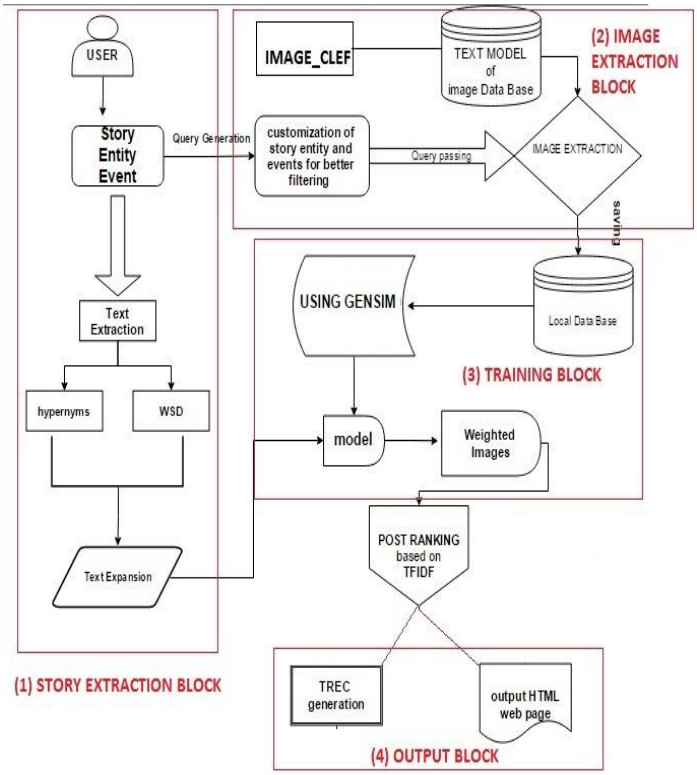

Figure 3: A block diagram showing the entire system

it is mapped to 0-100 scale. The mapped images are ranked and then extracted images with the rank are converted to ".TREC" file which is then evaluated by the FIRE 2015. The output is also generated in the form of HTML page with the best mapped images illustrating the story with pictures.

4. RANKING ALGORITHM

The result from TFIDF is in a scale of 0-1. Images have to be ranked based on the results obtained. We have divided the values into a 5 sub category form 0-4 based on below mentioned method. To compute this, we first compute a value called Range. Once when the range is computed, we assign the rank based on it. If the value is below the range, we assign "0". If the value is below twice the range, then we assign rank to be "1", for the range below thrice the rank, we assign "2" and so on. This computation goes on till we assign the rank "4", which is the maximum possible rank. The pseudocode is as follows:

```
Range=(Max(all.val)-Min(all.val))/5

If val < Range:
        Rank =0
Else if val < Range*2:
        Rank=1
Else if val < Range*3:
        Rank=2
Else if val < Range*4:
        Rank=3
Else:
        Rank=4
Return Rank
```

Figure 4: Algorithm describing the ranking methodology

5. TOOLS AND METHODS

Word embedding is a mapping of a word to a d-dimensional vector space. This real valued vector representation captures semantic and syntactic features. We have used gensim to implement this. For Vector Space modeling we have used Gensim toolkit. It is implemented in python and one can improve the performance using NumPy, SciPy etc. Efficient online algorithms are used in dealing with huge text data with the help of Gensim. Gensim has packages included for TF-IDF, latent semantic evaluation (LSA) and latent Dirichlet allocation (LDA), along with allotted parallel variations random projections, Google's word2vec and document2vec algorithms, etc. It finds its application in commercial as well as academic areas [8]. We have imported gensim from NLTK. NLTK is a major platform for building Python applications related to text analytics. Libraries for tokenization, stemming, tagging, parsing, and much more are included in here [9].

We have filtered out important information from the story using the concept of hypernym and hyponym. Hypernyms and hyponyms are semantic classes of words. Hypernyms are more broad in significance (hyper = "over") and hyponyms are more particular (hypo = "under"). Let us try to understand the concepts by some examples.

Example: color is a generalized term for all the colors. We call it the hypernym. Purple, green, red, blue etc. are hyponyms of color. Figure 3

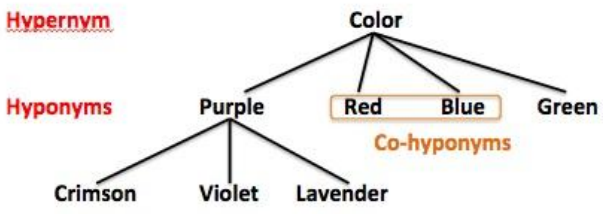

Figure 5: Hypernym and hyponym

6. EVALUATION AND RESULT

Evaluation is conducted by means of precision-at-K (P@K) and mean average precision (MAP) in relation to manual relevance assessments. Each important entity or event in a story will have a relevance list associated with it. P@K and MAP for each annotation are computed against these relevance scores. There were two teams that participated, including us. Our run is based on TFIDF. Our methodology has given better results when evaluated using MAP and B-pref.

Table 1: Showing the evaluation results

Run Name	TFIDF-1	cguj-run1	cguj-run2	cguj-run3
num_ret	6405	92	95	100
num_rel	2068	2068	2068	2068
num_rel_ret	255	16	20	13
MAP	**0.0107**	0.0047	0.0053	0.003
MRR	0.1245	**0.3708**	0.2997	0.2504
B-pref	**0.1241**	0.0074	0.0095	0.0065
P@5	0.0636	0.1273	**0.1545**	0.0909

7. CONCLUSION

In the proposed work we have used TFIDF to generate a sequence of images for the corresponding story. We have successfully implemented a text-to- picture engine that can effectively understand the core content of the story, and produce a set of images that best represents the story. The results are displayed in a web page and for evaluation purpose we have also generated a ".TREC" file. The results were evaluated by FIRE team and it has given considerably good accuracy. In future this can be extended to create gaming units, generate animation based on the story, to educate mentally retarded children an in the rehabilitation of brain-injured patients

AKNOWLEDGEMENT

We would like to thank Dr.Debasis Gangly and to Mr.Iacer Calixto, ADAPT Centre, Dublin City University (DCU), and for FIRE2015 team for organizing such a great event and guiding us through the entire journey.

4. REFERENCES

[1] Tomlinson, Carl M., and Carol Lynch-Brown. *Essentials of children's literature*. Allyn & Bacon, 1996

[2] "The Importance of Illustrations in Children's Books" in Illustrating for Children edited by Mabel Segun. Ibadan: CLAN, 1988. pp 25-27

[3] Goldberg, Andrew B., et al. "Easy as ABC: facilitating pictorial communication via semantically enhanced layout." Proceedings of the Twelfth Conference on Computational Natural Language Learning. Association for Computational Linguistics, 2008.

[4] Zhu, Xiaojin, et al. "A text-to-picture synthesis system for augmenting communication." AAAI. Vol. 7. 2007..

[5] Joshi, Dhiraj, James Z. Wang, and Jia Li. "The Story Picturing Engine---a system for automatic text illustration." *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 2.1 (2006): 68-89.

[6] Feng, Yansong, and Mirella Lapata. "Topic models for image annotation and text illustration." Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2010.

[7] Li, Jia, and James Z. Wang. "Automatic linguistic indexing of pictures by a statistical modeling approach." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 25.9 (2003): 1075-1088.

[8] Řehůřek, R., and P. Sojka. "Gensim–Python Framework for Vector Space Mo delling." *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic* (2011).

[9] Bird, Steven. "NLTK: the natural language toolkit." *Proceedings of the COLING/ACL on Interactive presentation sessions*. Association for Computational Linguistics, 2006.

[10] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." *the Journal of machine Learning research* 3 (2003): 993-1022.