A Supervised Approach for Text Illustration

Harsh Jhamtani
Adobe Systems
jhamtani@adobe.com

Shubham Varma
IIT Varanasi
varmashubham2@gmail.com

Midhun Gundapuneni
IIT Kharagpur
g.midhun95@gmail.com

Siddhartha Kumar Dutta
IIT Bombay
siddhartha_dutta@live.com

ABSTRACT

In this paper we propose a novel method to illustrate text articles with pictures from a tagged repository. Certain types of documents, like news articles, are often accompanied by a few pictures only. Prior works leverage topics or key phrases from the text to suggest relevant pictures. We propose a supervised model based on features like readability, picturability, sentiment polarity, and presence of important phrases, to identify and rank key sentences. The proposed method then suggests some relevant pictures based on the top ranked sentences thus identified.

CCS Concepts

•**Information systems** → *Multimedia content creation;*

Keywords

Text Illustration; Supervised Approach;

1. INTRODUCTION

Since pictures often draw reader's attention, illustrating text articles with pictures is a positive way to engage readers of an article. Our work focuses on automatically illustrating a text article with limited number of relevant pictures from the available repository of pictures.

We propose a novel method which uses supervised learning approach to score sentences based on factors like presence of important phrases or keywords, 'picturability'[1] of sentence, sentiment polarity of sentence, and readability of sentence. These scores are then leveraged to retrieve relevant pictures. The major contribution of our work is as follows. We propose a novel method for text illustration which uses supervised learning to score and rank sentences. Our method identifies importance of above mentioned features in determining the pictures to accompany a text article.

2. RELATED WORK

Our problem involves various aspects spread over areas such as image retrieval, and natural language processing. There are some existing works focusing on translating short textual sentences to a structured series of pictures ([22, 6, 21]). However, our problem is different as we aim to illustrate articles, and not representing a single sentence through pictures.

Many existing works identify important keywords and phrases from text to retrieve relevant pictures ([3], [2]) . Zhu et al. ([22]), extract 'picturable' key phrases from a given text. Lu et al. ([14]) have proposed models for illustrating travelogues by extracting location specific topics from travelogues and finding relevant pictures from tagged picture repository. In contrast to this, we take a supervised approach to identify sentences to picturize based on a variety of textual features.

Some research efforts focus on retrieval and ranking of pictures for given text article. In the work by Aletras and Stevenson ([3]), a graph is created with pictures as nodes and similarity of visual features between pictures as edge weights. Then a graph-based algorithm is used to rank the pictures. Joshi et al. ([9]) use the keywords extracted from text for retrieving pictures from an annotated database. Similarity between the pictures is then calculated using integrated region matching and annotation based similarity. Finally, the pictures are ranked using the principle of mutual reinforcement.

Delgado et al. ([5]) propose a method where sentences are expressed as a word vector and pictures as a vector of tags, and the relevant pictures are found for each sentence using cosine similarity value between the two vectors. However, in our problem setting, only few pictures are to be added to an article - and not one picture for every sentence.

3. DATA SETS

To build our model and conduct our experiments, we used 120 news articles from Reuters (www.reuters.com) and CleanLeap (www.cleanleap.com) (60 from each). The news categories covered are politics, international news, and energy. There are a total of 5169 sentences in these articles. Apart from the pictures already present in these articles, we crawled pictures from other news domains like BBC (www.bbc.com) and CNN, from articles similar to the 120 articles in our data set. We also crawled pictures from about 1000 randomly chosen news articles from various news domains. We considered only those pictures which had accompanying descriptions.

[1] Picturability of a sentence or a phrase is the degree to which that semantics of a sentence or phrase can be represented through a picture

MM '16, October 15-19, 2016, Amsterdam, Netherlands
© 2016 ACM. ISBN 978-1-4503-3603-1/16/10...$15.00
DOI: http://dx.doi.org/10.1145/2964284.2967214

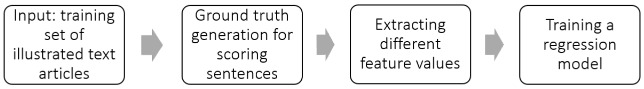
Input: training set of illustrated text articles
Ground truth generation for scoring sentences
Extracting different feature values
Training a regression model

Figure 1: Steps in training a supervised model to score sentences

4. SOLUTION DESCRIPTION

In this section we describe our proposed approach. Figure 1 shows the major steps in building a supervised model to score sentences in a text article. Figure 2 shows the steps in ranking sentences of a text article, and finally obtaining the relevant pictures.

4.1 Picturizing Score

In our proposed approach, a score is assigned to every sentence in the article, which we shall refer to as ***picturizing score*** . A sentence with a higher *picturizing score* should be prioritized over a sentence with a lower *picturizing score* score, when identifying pictures to accompany text. To build a model to predict this score, we consider already illustrated news articles to obtain the ground truth data. The assumption here is that the content author or marketer is an expert, and has, through manual efforts, selected the relevant and significant pictures to accompany the text. To obtain the ground truth, we define 'picturizing score' (S_{ij}) of a sentence i to be the number of pictures (n_{ij}) found corresponding to the sentence in the original article j, divided by a normalizing factor. The formula used is as follows:

$$S_{ij} = \frac{n_{ij}}{\log_2 (N_j + 1)} \tag{1}$$

Here, N_j refers to the total number of pictures corresponding to text in the j^{th} article. Normalization is required since different articles have different number of accompanying pictures. Logarithm of the number of pictures in the document is used to prevent over penalization of heavily illustrated documents.

To obtain n_{ij} values, we designed an annotation task. We recruited 3 annotators for this purpose. The annotators were asked to identify the number of pictures in an article corresponding to each sentence in the same article. They were provided with task description and some examples. We considered only those annotations for which at least two annotators agreed, considering the agreed value as the final annotation. Since this was an easy task, there was high agreement among the annotators. Out of 5169 annotations, only 273 (5.28%) were discarded.

We build a regression model to predict the picturizing score of a sentence, which represents the degree to which the sentence should be associated with pictures in the article. The features, on which independent variables are based, will be described in Section 4.3.

4.2 Vector space representation of text

We use a term frequency-inverse document frequency based vector space based model ([18]) for representation of sentences as well as picture descriptions. Each sentence and each picture description is considered as a document. Inverse document frequency of a term (a 'word' in our case), is calculated as follows:

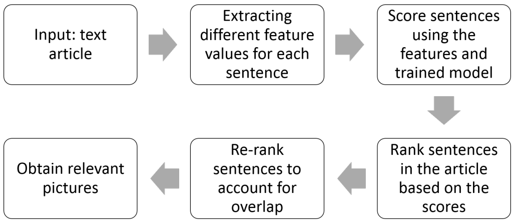
Input: text article
Extracting different feature values for each sentence
Score sentences using the features and trained model
Rank sentences in the article based on the scores
Re-rank sentences to account for overlap
Obtain relevant pictures

Figure 2: Steps in obtaining pictures after scoring sentences of a text article using the trained model

$$IDF(t) = log_2(\frac{N_t}{|D|}) \tag{2}$$

Here, N_t is the number of documents containing the term t, and $|D|$ is the total number of documents. Then, TF-IDF of a term t for a document d is calculated as:

$$tfidf(t, d) = n_{t,d} * IDF(t) \tag{3}$$

Here, $n_{t,d}$ is the the count of term t in the document d normalized by the total number of terms in the document d. In order to find similarity between two sentences or a sentence and an image, one can compute certain measures like cosine similarity between corresponding vector representations.

4.3 Feature extraction from text

We consider following four types of features: (1) Readability of sentence (2) Picturability of sentence (3) Presence of keywords and important phrases (4) Sentiment polarity of sentence

4.3.1 Readability of sentence

Here we try to capture the readability of a sentence i.e. how easy will someone find reading and understanding the sentence. The intuition here is that sentences with low readability should be accompanied with pictures, to enhance the understanding. We consider following three features values corresponding to this:

Fog Index: Fog index computes a readability score of a sentence using the length of sentence and number of complex words with 3 or more syllables [1]. Higher fog index means lower readability. The formula used is as follows:

$$fogindex(sentence) = 0.4 \times number\_of\_words(sentence) + 40 \times \frac{no\_of\_complex\_words}{total\_no\_of\_words} \tag{4}$$

Sentence length: For a given sentence this feature value is equal to the number of words in the sentence. To break a sentence into words, we used white spaces as delimiters.

4.3.2 Picturability

Leong et al. ([12]) proposed a measure of 'picturability' of a sentence based on the tags of pictures in the picture repository . For our model, we measure the picturability of a sentence based on the maximum similarity it has across all the picture descriptions in repository, calculated using cosine similarity between corresponding vector space representations. The underlying intuition here is to identify sentences which can be easily picturized through pictures present in the repository. Note that *picturability* is not to be confused with *picturizing score*.

4.3.3 Importance phrases/keywords

Here we describe features capturing presence of various important phrases and keywords in a sentence. Such features have been used in many text mining tasks ([11],[7],[8]). The underlying intuition is that the sentences bearing more important words and phrases should be prioritized while ranking sentences to picturize.

Text Ranking: Text Rank [16] builds an undirected graph using sentences as vertices. Edges are a measure of similarity between the sentence vertices, for which we use cosine similarity between vector representations of corresponding two sentences. Once the graph is constructed, it is used to form a stochastic matrix, combined with a damping factor, and the scores of vertices are obtained using PageRank. Using this we are able to obtain score for each sentence, such that the score denotes the importance of the sentence.

LSA score: Latent semantic analysis (LSA) is a technique of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms [20]. The method assigns significance scores to various concepts or topics, which is then used to determine sentence ratings. This is achieved by summing scores of various concepts present in a sentence.

TF-IDF based score: A score is obtained for each sentence by summing up the the TF-IDF values of unigrams, bigrams, and trigrams of words in the sentence. TF-IDF scores are obtained as described earlier in section 4.2.

4.3.4 Sentiment polarity

In many cases, we would not want to picturize a sentence comprising negative sentiment phrases and/or hate words. To capture this intuition, we generate a feature value for each sentence, which is the sentiment score of the sentence, as provided by Alchemy API [13]. The sentiment score is a real number between -1 and 1 (both inclusive). 1 represents the extreme positive sentiment polarity, while -1 represents the extreme negative polarity.

4.4 Training of model

Consider an **example** sentence: “A crewless cargo space ship burned up in the Earth’s atmosphere after a communication failure, and a Proton-M carrier rocket carrying a Mexican satellite crashed in Siberia.” Corresponding feature values are: lsa_score=5.71, fogindex=15.33, tfidf_score=0.63, sentence_length=36, picturability=0.43, and sentiment_score =0.0.

Normalization of feature values is done to bring the values to a scale of [0-1]. (Note that *picturizing score* is not to be confused with *picturability*). Regression models were trained using the normalized sentence features as independent variables and the sentence ‘picturizing score’ as dependent variable. The training involves learning various parameter values of the regression model using the annotated data.

4.5 Discounting for overlap among sentences

Since after ranking, the top sentences would be leveraged to obtain relevant pictures, overlap among top sentences should be minimized to avoid obtaining pictures depicting very similar ideas. We use Marginal Relevance (MMR) [4] to re-rank the sentences by combining picturizing scores and degree of overlap of the sentence with already selected sentences, using a linear combination function.

The following formula is used to obtain the best item from yet unselected items at any iteration:

$$\max_{D_i \epsilon R-S}[\lambda * Picturizing\_score_{D_i} - (1-\lambda)\max_{D_j \epsilon S} sim(D_i, D_j)] \qquad (5)$$

Where R is the set of all sentences in the current text article. S is the subset of sentences in R already selected; $R-S$ is the set of as yet unselected sentences in R; $sim(D_i, D_j)$ is the cosine similarity score between vector space representations of D_i and D_j. Parameter λ takes a real value between 0 and 1. Higher the value of λ, more would be the importance given to *picturizing score* as compared to overlap between sentences. We used $\lambda = 0.5$ for our experiments.

4.6 Obtaining illustrated article

For a given sentence, the picture in the repository, corresponding to which the maximum similarity score is obtained, is selected. To compare similarity between a sentence and a picture, we calculate the cosine similarity between the vector representations of the sentence and the corresponding picture description. We inquire user about the number of pictures he wants to add, represented by K. We obtain a ranked list of sentences as explained above. Thereafter, we select one relevant picture for each of the top K sentences. The retrieved pictures are placed near the corresponding sentences to obtain the illustrated article.

5. EXPERIMENTS AND RESULTS

Here we describe various experiments we performed and report the obtained results. We have performed following two experiments: (1) Experiments to evaluate regression models to score sentences (2) Experiments to evaluate overall performance of the proposed text illustration system

5.1 Experiments to evaluate regression models

In Section 4.1, we had described how ground truth values of *picturizing score* are obtained. For evaluating the performance of the regression models, we randomly select 80% text articles, and the sentences in these articles are used to generate the training data. Sentences in the remaining articles constitute the test data. We experimented with Linear Regression, and Support Vector Regression (with linear and RBF kernels). Average MSE (Mean square error) values under five-fold cross validation for these models on the training set are **0.031**, 0.064, and 0.069 respectively, while average MSE values on test set are **0.033**, 0.046, and 0.035 respectively. We use Scikit [17] implementation of these regression models. Linear regression model performs the best. The mean square errors are low, and moreover, the testing errors are comparable to the training errors. Thus, it can be inferred that the regression model fits quite well on the data. The goodness of fit represented by R^2 is **0.87** for the Linear regression model, showing that 87% percentage of variance is explained by the chosen features.

The regression coefficient values of the features, corresponding to the Linear Regression model, are shown in Table 1. We also report significance of various features, which is calculated using hypothesis testing. The ‘Null Hypothesis’ is that the coefficient in question is 0. Note that a coefficient

Table 1: Feature Coefficients and Significance levels for Linear Regression Model

Feature name	Coefficient	Significance Level ($Pr > \|t\|$)
LSA Score	-0.03	0.510
Textrank Score	-0.03	0.670
Fog Index	0.09	0.037
TFIDF	0.23	3.46e-07
Picturability	0.08	0.002
Sentiment Value	0.05	0.049
Sentence Length	0.02	0.011

value of zero will mean that the corresponding feature is not a predictor of the *picturizing score*. We use Student's t-test ([15]) at 0.95 confidence level. Based on the test, those features will be significant for which $Pr > |t|$ is less than 0.05 TF-IDF, picturability score, fog index, sentiment score, and sentence length are found to be statistically significant. A surprising result is that LSA based score and Text Rank were found to be insignificant (All significant features are marked in bold in Table 1). One possible explanation for TextRank being insignificant is that those sentences score high for which there are other similar sentences in the the article, which may not be desirable.

For each article forming up the test data sentences, we rank the sentences as per the scores given by the regression model. We already have ground truth rank list for each article - obtained by sorting the sentences by the annotated score. We then compare two ranked lists for each article in the test set. Kendall Tau rank correlation coefficient τ [10] and Spearman's rank correlation coefficient ρ [19] for the Linear Regression model were found to be **0.856** and **0.699** respectively. This means that our method is able to provide a rank list very similar to the ground truth rank list.

5.2 Experiments to evaluate overall performance of the proposed system

For the baseline approach, we use Text Rank [16] to identify a ranked list of important terms. We build a graph with words (excluding stopwords) as vertices, such that an edge exists between two vertices if they occur within a window size of 8. Then PageRank is run on the graph to obtain top T significant vertices(words). Thereafter, contiguous significant words are coalesced, with score being the sum of scores of coalesced words. If the total number of pictures desired are K, then we consider the top K terms(words or coalesced set of words). For each term, we find a picture from repository containing the words in the term.

We created 24 sets of three articles each, of which one is illustrated using baseline method, one is illustrated using our method, and one is article without pictures. We recruited a total of 24 people, such that each one of them is assigned one set, and is then asked to rate each of the articles in the set on a scale of 1-7 on different aspects. The questions that were asked are shown in Table 2.

Table 2: Questionnaire for evaluation

Ques. no.	Question text (H/L) [2]
Q.1	The article was engaging (H)
Q.2.	The text of the article was easy to understand (H)
Q.3.	The pictures in the article were useful to understand the article (H)
Q.4.	The pictures were unhelpful in maintaining my interest in reading the article (L)
Q.5.	The pictures were relevant to the content of the article (H)
Q.6.	The pictures did not capture the main points of the article (L)
Q.7.	The two pictures in the article conveyed overlapping information (L)
Q.8.	Please take a look at the two pictures again. Then answer the following question: By themselves the two pictures told the story in the article (H)

Table 3: Mean rating values with corresponding standard deviations

Ques. no.	Our method: Mean (σ)[3]	Baseline: Mean (σ)	Without pictures: Mean (σ)
Q.1 (H) [2]	5.42 (0.71)	4.72 (0.41)	3.85 (0.75)
Q.2 (H)	5.05 (1.02)	4.20 (0.43)	3.35 (0.75)
Q.3 (H)	5.52 (0.79)	4.63 (0.63)	-
Q.4 (L)	2.62 (0.83)	3.95 (1.31)	-
Q.5 (H)	5.40 (0.98)	4.30 (0.58)	-
Q.6 (L)	2.78 (1.01)	3.88 (0.48)	-
Q.7 (L)	2.73 (0.60)	4.63 (0.39)	-
Q.8 (H)	4.85 (0.41)	3.85 (0.39)	-

Analysis of responses: Table 3 shows the mean ratings for different questions for our approach, baseline approach, and non-illustrated articles (articles without pictures). Note that for non-illustrated articles, we only asked first two questions. It can be seen that our method received favorable scores in most cases, and outperforms the baseline approach. Moreover, we report the standard deviation for rating for different questions for our method as well as for baseline method in Table 3. On considering the standard deviation values, we observe that the mean value of ratings for our method differs from the baseline method by more than unit standard deviation for most cases.

6. CONCLUSION

We have proposed a novel approach based on supervised learning to identify key sentences, and then find relevant pictures. An advantage of our approach is that we consider a broad range of features beyond important phrases. Through experiments, we found that features like fog index (readability), picturability, sentence length, sentiment polarity of sentence, and TFIDF based presence of important terms, are significant predictors of key sentences for text illustration. In future, we would like to extend our work to consider features like position of sentence, font, etc.

[2] For questions 1,2,3,5,8 a higher score is more favorable (H) while a lower score is more favorable for questions 4,6,7 (L)

[3] Standard Deviation

7. REFERENCES

[1] The gunning's fog index readability formula. Accessed: 2015-07-19.

[2] Agrawal, R., Gollapudi, S., Kannan, A., and Kenthapadi, K. Enriching textbooks with images. In *Proceedings of the 20th ACM international conference on Information and knowledge management* (2011), ACM, pp. 1847–1856.

[3] Aletras, N., and Stevenson, M. Representing topics using images. In *HLT-NAACL* (2013), pp. 158–167.

[4] Carbonell, J., and Goldstein, J. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (1998), ACM, pp. 335–336.

[5] Delgado, D., Magalhaes, J., and Correia, N. Automated illustration of news stories. In *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on* (2010), IEEE, pp. 73–78.

[6] Goldberg, A. B., Zhu, X., Dyer, C. R., Eldawy, M., and Heng, L. Easy as abc?: facilitating pictorial communication via semantically enhanced layout. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning* (2008), Association for Computational Linguistics, pp. 119–126.

[7] Gupta, V., Varshney, D., Jhamtani, H., Kedia, D., and Karwa, S. Identifying purchase intent from social posts. In *ICWSM* (2014).

[8] Jhamtani, H., Chhaya, N., Karwa, S., Varshney, D., Kedia, D., and Gupta, V. Identifying suggestions for improvement of product features from online product reviews. In *International Conference on Social Informatics* (2015), Springer, pp. 112–119.

[9] Joshi, D., Wang, J. Z., and Li, J. The story picturing engine—a system for automatic text illustration. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 2*, 1 (2006), 68–89.

[10] Kendall, M. G. A new measure of rank correlation. *Biometrika* (1938), 81–93.

[11] Kosala, R., and Blockeel, H. Web mining research: A survey. *ACM Sigkdd Explorations Newsletter 2*, 1 (2000), 1–15.

[12] Leong, C. W., Mihalcea, R., and Hassan, S. Text mining for automatic image tagging. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters* (Stroudsburg, PA, USA, 2010), COLING '10, Association for Computational Linguistics, pp. 647–655.

[13] LLC, O. Alchemyapi, 2009.

[14] Lu, X., Pang, Y., Hao, Q., and Zhang, L. Visualizing textual travelogue with location-relevant images. In *Proceedings of the 2009 International Workshop on Location Based Social Networks* (2009), ACM, pp. 65–68.

[15] Mankiewicz, R. *The story of mathematics*. Princeton University Press, 2000.

[16] Mihalcea, R., and Tarau, P. Textrank: Bringing order into texts. In *Proceedings of EMNLP 2004* (Barcelona, Spain, July 2004), D. Lin and D. Wu, Eds., Association for Computational Linguistics, pp. 404–411.

[17] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research 12* (2011), 2825–2830.

[18] Salton, G., Wong, A., and Yang, C.-S. A vector space model for automatic indexing. *Communications of the ACM 18*, 11 (1975), 613–620.

[19] Spearman, C. The proof and measurement of association between two things. *The American journal of psychology 15*, 1 (1904), 72–101.

[20] Steinberger, J., and Jezek, K. Using latent semantic analysis in text summarization and summary evaluation. In *Proc. ISIM'04* (2004), pp. 93–100.

[21] UzZaman, N., Bigham, J. P., and Allen, J. F. Multimodal summarization of complex sentences. In *Proceedings of the 16th international conference on Intelligent user interfaces* (2011), ACM, pp. 43–52.

[22] Zhu, X., Goldberg, A. B., Eldawy, M., Dyer, C. R., and Strock, B. A text-to-picture synthesis system for augmenting communication. In *AAAI* (2007), vol. 7, pp. 1590–1595.