Relevant Tag Extraction Based on Image Visual Content

Nancy Fazal(✉) and Pasi Fränti

School of Computing, University of Eastern Finland, Joensuu, Finland
{nancyf,pasi.franti}@uef.fi

Abstract. Social media web services like Flickr allow users to share and freely annotate images with textual tags. These tags play a crucial part for text-based social image retrieval and browsing tasks. However, these tags are usually irrelevant and incomplete which limits their effectiveness and use. One fundamental problem is to interpret the relevance of tags with respect to the image visual content. Existing solutions have targeted either image visual content or user contributed tags separately to address the issue. Our proposed method *Tag-tag*, however, combines both aspects and exploit their semantic relationship.

We use state of the art pretrained machine learning models in Tensorflow for object detection and NLP (Natural Language Processing) for semantic analysis. Our experiments on a dataset of 219 randomly collected Flickr images demonstrates the applicability of our method. Images with missing tags were excluded from the experiments. We identified two reasons where relevant tag was not found as: **(1)** relevant tag itself was missing from user contributed tag list, **(2)** relevant tag got skipped because it was either not found in the Wordnet dictionary or has no pretrained vector in word2vec-google-news-300 model. We identify this as one limitation of the proposed method.

Keywords: Social Tagging · Tag Relevance · Flickr · Object Detection · Semantic Similarity

1 Introduction

The popularity of several digital imaging devices and with the advancement of Internet technologies, digital images can be easily created and distributed. Social media tagging, a process where images, videos and text objects are mostly assigned with tags or keywords by common users, is reshaping the way people generate, manage and search multimedia resources [1]. Services like Flickr which cumulates 10 billion images with around 3.5 million new uploads per day are flourishing [33]. Besides their general use, these rich multimedia databases have triggered many innovative research domains such as tag recommendation [2], landmark recognition [3], tag ranking [4], concept similarity measurement [5], automatic image annotation [6] and personalized information delivery [7].

D.-S. Huang et al. (Eds.): ICAI 2023, CCIS 2015, pp. 1–13, 2024.
https://doi.org/10.1007/978-981-97-0827-7_25

The major concern, however, is that the multimedia objects are usually not annotated properly and includes irrelevant, and often incomplete tags. Some objects are left completely unlabeled also. Liu et al. [4] reported that only 50% of the tags are related to the image content, which poses a great challenge for typical web image search approaches. To address this issue, various tag refinement techniques have been reported by the researchers recently, which can improve the quality of tags [1, 5, 8–11]. Tag refinement improves the quality of tags but does not answer the tags which best represent the visual content of an image [12]. Kennedy et al. [13] reports that the tags associated with images contain many noises and not only describe the image contents but a broad spectrum of semantic space such as location, time, and subjective emotion [14, 15]. Liu et al. [3] claims that the automatic detection of tags which are content-related can support more smart use of the social images and tags such as ease the task of browsing, retrieval and indexing of large-scale image repositories [16].

Li et al. [1] proposed a voting algorithm to find a relevant tag from the tagging behavior of visual neighbors of that image. They conducted three experiments one for image ranking and two tag ranking experiments for verification purpose. Their experiments on 3.5 million Flickr images improved upon baselines and demonstrated the usability of algorithm for both social image retrieval and image tag recommendation. Liu et al. [4] reports that the tags are almost in random order in terms of their relevance to the associated tags. They propose a tag ranking method to automatically rank tags for a given image, according to their relevance to image visual content. Their experiments on 50,000 Flickr photo collection show that the proposed method is both effective and efficient. They further applied tag ranking into three applications i.e., tag-based image search, tag recommendation, and (3) group recommendation, which demonstrates that the proposed tag ranking approach really boosts the performances of social-tagging related applications.

Zhao et al. [16] proposed PoCR a data driven method to assess the probability of a tag relevancy to its corresponding image. Their experiments on 149,915 Flickr images demonstrated that PoCR achieved best performance by obtaining 59.8%, 26.6%, 29.3%, and 20.4% relative improvements compared to Baseline, LiCR g, LiCR l, and LiCR f respectively. Liu et al. [36] proposed the pixel voting method to choose the visual neighbors for seed image to find a representative tag. Their experiments on MIR Flickr dataset show the effectiveness of the method in tag de-noising and tag ranking. They also stated that the concern of tag relevancy learning could not be resolved completely because of the semantic gap between the images and tags.

Lindstaedt et al. [17] proposed a research prototype *tagr* which makes use of data from Flickr group “fruit & veg” and electronic lexical database WordNet. Techniques from the various fields such as image analysis, social network analysis and statistical text analysis were used to develop the service. The results revealed that the despite of low precision and recall values test users did find the *tagr* useful. Zhuang et al. [12] proposed a novel two-view learning approach for social image tag ranking by exploiting both visual and textual contents. To evaluate the method efficiency, extensive set of experiments on automatic image annotation task and text-based social image retrieval were conducted. Encouraging results were reported as proposed method outperformed the conventional approaches. See Table 1 for the summary of the existing methods.

Table 1. Existing Tag relevance methods comparison.

Reference	Method	Datasets Used	Data Context	Properties
[1]	Tagging behavior of visually similar neighbors	3.5 M	Images and tags	Does not require any model training for any visual concept but visually similar neighbors for every seed image are needed
[4]	Probability density estimation and Random walk	50,000	Tags	Does not require model training
[16]	Data-driven method	149,915	Image and tags	Does not require any model training and limited to 270 popular Flickr tags only
[17]	*Tagr*	14,000	Tags, images, and users	Offline analysis and limited to Flickr group *fruit & veg*
[12]	Data-driven method	1 M	Tags and images	No parametric model relevance between images and tags
[36]	Pixel voting method	MIR Flickr	Images and tags	Visually similar neighbors for every seed image required to focus on the local features of an image

In this paper, we propose a new method for extracting a tag which best describes the image visual content. Instead of relying on the existing tags alone, or use image visual content as such, we apply them jointly. We refer this method as *Tag-tag*, which exploits the semantic relationship between objects identified on a given image and associated user tags. The method is completely independent of visually similar images, their associated tags and is not restricted to a certain set of tag groups. It further relies on state of the art pretrained models, thus does not require building anything from the scratch.

2 Representative Tag Extraction Using Tag-Tag

We define the representative tag as the one which best describes the objective aspects of the visual content of an image (see Fig. 1). Our method to select a representative tag for a photo comprise of two approaches i.e., computer vision techniques and natural language processing. We use TensorFlow object detection, a computer vision technique that detects, locates, and traces an object from a still image or video and NLTK (Natural Language Toolkit) to perform semantic analysis of textual tags.

The proposed method consists of three steps as 1) Object detection using TensorFlow, 2) Semantic analysis of textual tags using NLTK (WordNet) and 3) Deriving representative tag (Wu & Palmer similarity and Word2Vec). The thorough processing of the *Tag-tag* method is shown in Algorithm.

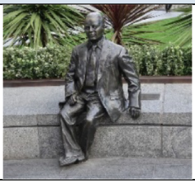
Photo	User-provided tags	Representative tag
	London, Paddington, UK, Sir Simon Milton, statue, sculpture, deputy mayor, official	sculpture / statue

Fig. 1. Representative tag extraction based on the image visual content.

ALGORITHM: REPRESENTATIVE TAG SELECTION USING WORDNET SIMILARITY AND WORD2VEC SIMILARITY

Input: Image *i* and its user contributed list of tags *t;*

Output: A representative tag *rTag* which best describes the visual content of an image;

```
Obtain unique object detections d using Tensorflow object detection model FasterRCNN+InceptionResNet V2
Extract set of tags t and object detections d present in the Wordnet dictionary and word2vec-google-news-300 model using wordnet.synsets(word) and word not in google_news_vectors.key_to_index respectively
Perform Part of Speech (POS) tagging using nltk.pos_tag(t)
Filter named entities using NLTK and extract tags with NN, NNS, NNP and NNPS, POS labels
for detection in d do
    for tag in t do
        WORD2VECSIMILARITY(d, t)
// For representative tag extraction using Wordnet Similarity call WORDNETSIMILARITY(d, t)
        scores ← append(similarityScore)
    matrix ← append(scores)
maxScore ← [sum(col) for col in zip(*matrix[1:])]
rTag ← matrix[0][ maxScore.index(max(maxScore))]
return rTag
function WORD2VECSIMILARITY (d, t)
    similarityScore ← google_news_vectors.similarity(d, t)
    return similarityScore
function WORDNETSIMILARITY (d, t)
    w1 ← wordnet.synsets(t)[0]
    w2 ← wordnet.synsets(d)[0]
    similarityScore ← w1.wup_similarity (w2)
    return similarityScore
```

2.1 Object Detection Using Tensorflow

The object detection framework in TensorFlow works on trained models, so it does not require building anything from scratch. Prioritizing accuracy over speed, we chose FasterRCNN + InceptionResNet V2 [19] module for object detection task. This model is trained on Open Images V4 with ImageNet pre-trained Inception Resnet V2 as image feature extractor. The model is further publicly available as part of the TensorFlow object detection API.

The Inception ResNetV2 feature extractor was trained on ImageNet and fine-tuned with FasterRCNN head on OpenImages V4 dataset containing 600 classes. Open Images is a dataset of ~9M images annotated with image-level labels, object bounding boxes and visual relationships. The training set of Open Images V4 [20] contains 14.6 million bounding boxes for 600 object classes on 1.74 million images, making it the largest existing dataset with object location annotations. The images used are diverse and often contain complex scenes with several objects. The hierarchy for 600 boxable classes can be viewed and download as JSON file [21, 34].

For a given image, we first perform the object detection and extract all the identified objects. Next, we apply 2 level filtering to find unique detections made, and extracting only those which are present in the Wordnet dictionary [22] and pretrained vectors of the word2vec-google-news-300 [23] model for deriving representative tag (see Fig. 2).

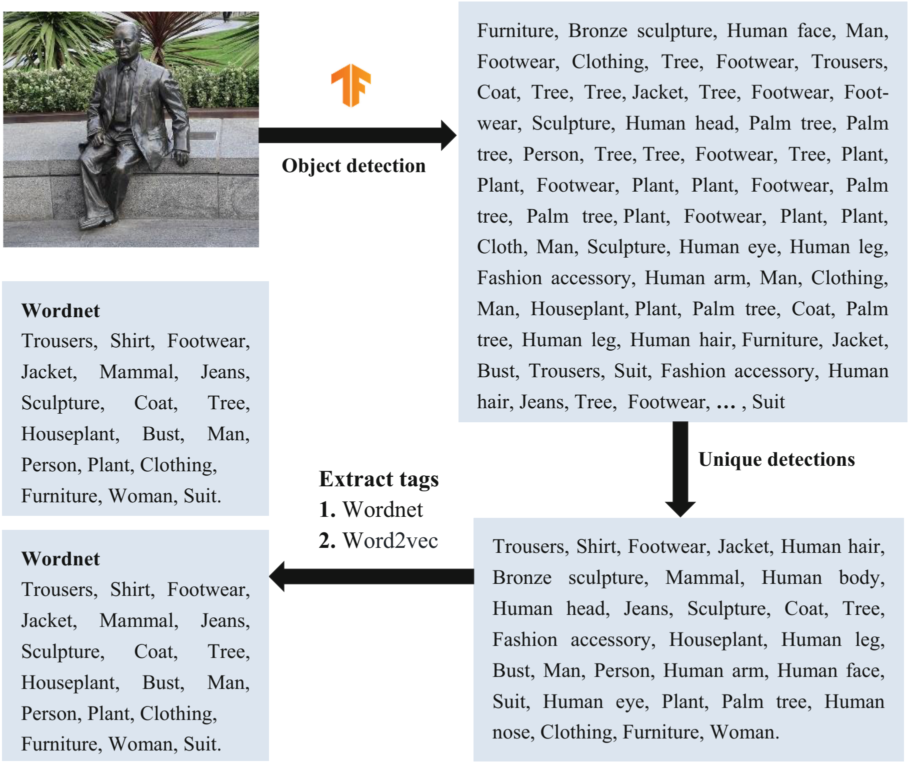

Object detection
Furniture, Bronze sculpture, Human face, Man, Footwear, Clothing, Tree, Footwear, Trousers, Coat, Tree, Tree, Jacket, Tree, Footwear, Footwear, Sculpture, Human head, Palm tree, Palm tree, Person, Tree, Tree, Footwear, Tree, Plant, Plant, Footwear, Plant, Plant, Footwear, Palm tree, Palm tree, Plant, Footwear, Plant, Plant, Cloth, Man, Sculpture, Human eye, Human leg, Fashion accessory, Human arm, Man, Clothing, Man, Houseplant, Plant, Palm tree, Coat, Palm tree, Human leg, Human hair, Furniture, Jacket, Bust, Trousers, Suit, Fashion accessory, Human hair, Jeans, Tree, Footwear, ... , Suit
Unique detections
Trousers, Shirt, Footwear, Jacket, Human hair, Bronze sculpture, Mammal, Human body, Human head, Jeans, Sculpture, Coat, Tree, Fashion accessory, Houseplant, Human leg, Bust, Man, Person, Human arm, Human face, Suit, Human eye, Plant, Palm tree, Human nose, Clothing, Furniture, Woman.
Extract tags
1. Wordnet
2. Word2vec
Wordnet
Trousers, Shirt, Footwear, Jacket, Mammal, Jeans, Sculpture, Coat, Tree, Houseplant, Bust, Man, Person, Plant, Clothing, Furniture, Woman, Suit.
Wordnet
Trousers, Shirt, Footwear, Jacket, Mammal, Jeans, Sculpture, Coat, Tree, Houseplant, Bust, Man, Person, Plant, Clothing, Furniture, Woman, Suit.

Fig. 2. Object detection and tags grouping

2.2 Semantic Analysis of Textual Tags

Natural language processing (NLP) is a subfield of Artificial Intelligence (AI). This is a widely used technology which deals with the interaction between computers and humans in natural language. It processes and analyses the natural language data such as text and speech with the goal of understanding the meaning behind the language. Some common techniques used in NLP includes tokenization, part-of-speech tagging, named entity recognition, sentiment analysis, machine translation and text classification. In this paper, we have used the Natural Language Toolkit (NLTK) a python package to work with NLP for carrying out the semantic analysis of textual tags. NLTK acts as a toolbox for NLP algorithms and provides easy-to-use interfaces to over 50 corpora and lexical resources such as Wordnet [24].

In this step, the user-provided tags of the photo are refined before computing the semantic similarity. Firstly, we extract two sets of tags i.e., tags which are present in the Wordnet dictionary and the ones which are present in the word2vec-google-news-300 model. Secondly, we apply Part-of-Speech (POS) tagging, a process where each word in a text is labeled with its corresponding part of speech. This may include nouns, pronouns, verbs, adverbs, adjectives, and other grammatical categories. Thirdly, we filter the named entities [35] followed by extracting tags which are identified as Singular Common Nouns **(NN)**, Plural Common Nouns **(NNS)**, Singular Proper Nouns **(NNP)**, and Plural Proper Nouns **(NNPS)** (see Fig. 3).

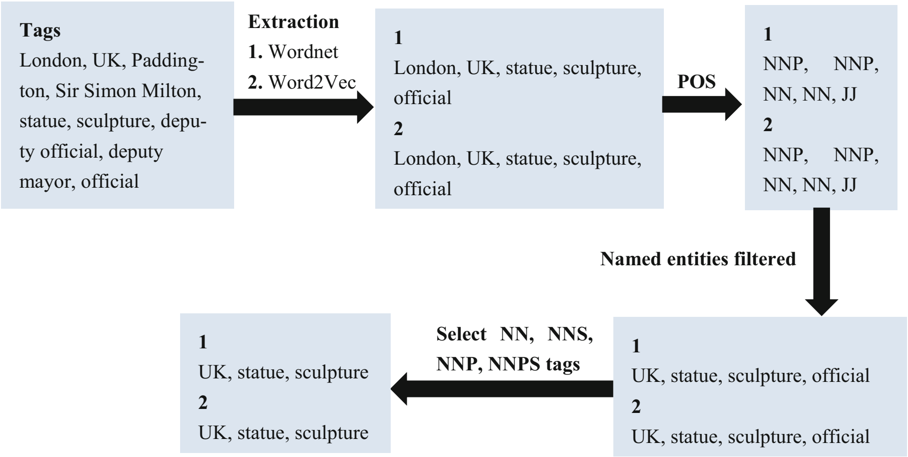

Tags
London, UK, Paddington, Sir Simon Milton, statue, sculpture, deputy official, deputy mayor, official
Extraction
1. Wordnet
2. Word2Vec
1
London, UK, statue, sculpture, official
2
London, UK, statue, sculpture, official
POS
1
NNP, NNP, NN, NN, JJ
2
NNP, NNP, NN, NN, JJ
Named entities filtered
1
UK, statue, sculpture, official
2
UK, statue, sculpture, official
Select NN, NNS, NNP, NNPS tags
1
UK, statue, sculpture
2
UK, statue, sculpture

Fig. 3. Semantic analysis and tags refinement

2.3 Extracting Representative Tag

Wu & Palmer Similarity. To select a representative tag for a given photo, we compute the semantic similarity between a unique set of object detections (Fig. 2) and refined tags (Fig. 3). For this purpose, we have considered two NLP techniques i.e., Wu & Palmer similarity [26] and Word2Vec [27]. Wu & Palmer similarity returns a score denoting how similar two-word senses are, by considering depths of two synsets (groups of synonymous words expressing the same concept) in the WordNet taxonomies, along with the depth of the LCS (Least Common Subsumer).

$$Wu - Palmer = 2 * \frac{\mathrm{depth(lcs(s1,\, s2))}}{(depth(\mathrm{s}1) + depth(s2))} \tag{1}$$

For a given tag and object detected, we first retrieve the list of available synsets. Some words may have only one synset and some may have several. We, however, use the first available synset. The list of synsets can be retrieved using wordnet.synsets(word). For computing Wu & Palmer similarity using NLTK, we use wup_similarity function as follows:

$$synset1.wup\_similarity(synset2)$$

Next, we derive a matrix of $d \times t$ dimensions. Where d represents the number of objects detected and t represents the number of tags. Each element in the matrix represents a wup_similarity score between d and t. Finally, we perform the summation of matrix columns and a tag with maximum column sum is acknowledged as representative tag for a photo (see Fig. 4).

Word embedding – Word2Vec. Word Embedding in NLP is an important aspect which connects a human language to that of a machine. This technique transforms the words into a numerical representation of words (vectors). These vectors try to capture the different characteristics of words regarding the overall text, including semantic relationships of words, definitions, and context etc. [28]. One Hot Encoding, TF-IDF (Term Frequency-Inverse Document Frequency), Bag of Words, Word2Vec, FastText and GloVe (Global Vectors for Word Representations) are frequently used Word Embedding methods. Word Embedding find its applications in music/video recommendation systems, analyzing survey responses, verbatim comments, and others [29].

We chose Word2Vec method to find the semantic similarity between user contributed tags and objects detected. It was developed by Thomas Mikolov in 2013 at Google. It is a popular word embedding technique which embed words in a lower-dimensional vector space using shallow neural network. This results in a set of word vectors where vectors close together in a vector space are semantically related and word vectors distant in vector space have different meanings. For example, *clean* and *tidy* would be close together as compared to the *clean* and *season* [29]. Embeddings learned through Word2Vec has proven to be efficient with learning high-quality vector representations and capturing semantic and syntactic information [30]. It has two neural network-based variants: Continuous Bag of Words (CBOW) and Skip-gram.

While one can train their own Word2Vec embeddings, we take advantage of the pre-trained vectors on part of Google News dataset of about 100 billion words. This model (**word2vec-google-news-300**) contains 300-dimensional vectors for 3 million words and phrases. The model is available through *Gensim*, which is a free open-source python library for unsupervised topic modeling and natural language processing [29, 31]. For a given list of tags and object detections, we compute the cosine similarity using built-in *similarity* function as shown below: [32]

$$google\_news\_vectors.similarity('Statue', 'Clothing') \tag{2}$$

Next, we derive a matrix of $d \times t$ dimensions as discussed above. Where, d represents the number of objects detected and t represents the number of tags (see Fig. 4). Each element in the matrix represents a cosine similarity score between d and t. Finally, we perform the summation of matrix columns and a tag with maximum column sum is acknowledged as representative tag for a photo (see Fig. 4).

$$\left[\begin{array}{cccc} & UK & Statue & Sculpture \\ Shirt & 0.32 & 0.53 & 0.55 \\ Jeans & 0.30 & 0.50 & 0.53 \\ Tree & 0.30 & 0.40 & 0.42 \\ Coat & 0.30 & 0.50 & 0.53 \\ Jacket & 0.29 & 0.48 & 0.50 \\ Plant & 0.33 & 0.55 & 0.59 \\ Suit & 0.32 & 0.53 & 0.55 \\ Footwear & 0.33 & 0.55 & 0.58 \\ Person & 0.36 & 0.47 & 0.50 \\ Bust & 0.12 & 0.27 & 0.29 \\ Mammal & 0.30 & 0.40 & 0.42 \\ Trousers & 0.32 & 0.53 & 0.55 \\ Woman & 0.33 & 0.42 & 0.44 \\ Houseplant & 0.33 & 0.44 & 0.48 \\ Clothing & 0.33 & 0.59 & 0.63 \\ Furniture & 0.33 & 0.55 & 0.59 \\ Man & 0.33 & 0.42 & 0.44 \\ Sculpture & 0.32 & 0.84 & 0.88 \\ \mathbf{Sum} & \mathbf{5.56} & \mathbf{8.97} & \mathbf{9.47} \end{array}\right] \left[\begin{array}{cccc} & UK & Statue & Sculpture \\ Shirt & 0.07 & 0.05 & 0.09 \\ Jeans & 0.11 & 0.06 & 0.17 \\ Tree & 0.03 & 0.12 & 0.22 \\ Coat & 0.07 & 0.15 & 0.09 \\ Jacket & 0.09 & 0.12 & 0.10 \\ Plant & 0.05 & 0.14 & 0.13 \\ Suit & 0.07 & 0.09 & 0.01 \\ Footwear & 0.06 & 0.20 & 0.19 \\ Person & -0.02 & 0.39 & 0.71 \\ Bust & 0.04 & 0.05 & 0.04 \\ Mammal & 0.09 & 0.09 & 0.14 \\ Trousers & 0.13 & 0.09 & 0.07 \\ Woman & 0.09 & 0.15 & 0.17 \\ Houseplant & 0.01 & 0.09 & 0.09 \\ Clothing & 0.08 & 0.19 & 0.21 \\ Furniture & 0.08 & 0.03 & 0.11 \\ Man & 0.07 & 0.07 & 0.04 \\ Sculpture & 0.07 & 0.07 & 0.10 \\ \mathbf{Sum} & \mathbf{1.19} & \mathbf{2.15} & \mathbf{2.68} \end{array}\right]$$

Fig. 4. Representative tag calculation using Wu & Palmer similarity (left) and Word2Vec (right).

3 Experiments

We collected a random sample of 169 Flickr images against 6 different locations around the world for experimental purpose (see Table 2). Flickr standardizes the user-provided tags by removing the space between words and converting the letters into lowercase. A user tag “My Helsinki” would become “myhelsinki” [18]. However, for research purpose we rely on the raw tags. Images with missing tags were excluded. It is worth mention that the authors collected the ground truth data by hand. On average, each photo had 6 to 7 tags ranging from name of the place, content of the photo, weather details, camera information and time information. Our results show that representative tag derivation using Wu & Palmer similarity (Wordnet similarity) outperforms cosine similarity measures in Word2Vec by 11%.

For images, where representative tag was not correctly identified, we observed the following two main reasons:

1. For a given list of user tags, representative tag itself was missing (see Fig. 5)
2. Representative tag existed in the user tags but got filtered because it was either not found in the Wordnet dictionary or has no pretrained vector in the word2vec-google-news-300 model (see Fig. 6)

Table 2. Representative tag selection results.

Locations	Images Inspected	Average number of tags	Images without tags	Wu & Palmer correct prediction	Word2Vec correct prediction
Helsinki Cathedral	27	7	14	38%	23%
Stonehenge	41	6	11	40%	3%
Leaning Tower of Pisa	101	13	22	41%	29%
Koli	1	9	0	100%	100%
Hyde Park	42	7	14	46%	32%
Mont des Arts Garden	7	6	0	14%	28%

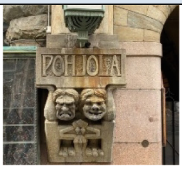
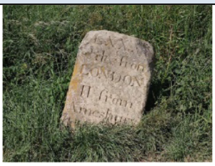
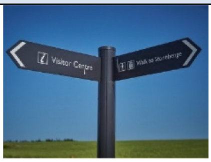
<table>
<tr><th colspan="3">Photos</th></tr>
<tr><td> </td><td></td><td> </td></tr>
<tr><th colspan="3">User-provided Tags</th></tr>
<tr><td>финляндия, q, finland, helsinki</td><td>stonehenge, world heritage site, photographie, landscapes, wbayer.com</td><td>stonhenge, stone, circle, neolithic, wiltshire, uk</td></tr>
<tr><th colspan="3">Representative tags Wordnet / Word2Vec</th></tr>
<tr><td>helsinki / helsinki</td><td>Stonehenge / landscapes</td><td>stone / stone</td></tr>
<tr><th colspan="3">Ground truth Tag</th></tr>
<tr><td>Pohjola</td><td>Graffitied Stone</td><td>Sign</td></tr>
</table>

Fig. 5. Representative tag itself was missing in user's contributed list of tags

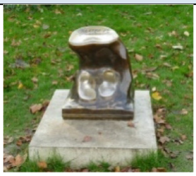
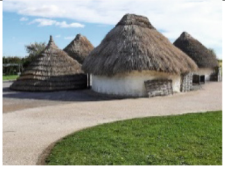
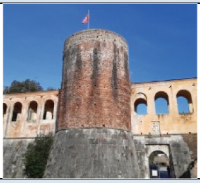
Photos		
		
User-provided Tags		
London, Camden, Regent's Park, London Parks, Frieze Sculpture, Frieze Sculpture 2022, Public Sculpture, Tim Etchells	Stonehenge, Wiltshire, Neolithic Village	tuscany, pisa, giardino scotto
Representative tags Wordnet / Word2Vec		
Camden / -	Stonehenge / Stonehenge	tuscany / -
Ground truth Tag		
Frieze Sculpture	Neolithic Village	Giardino scotto

Fig. 6. Representative tag was neither present in the Wordnet dictionary nor in Word2Vec model

4 Conclusions and Future Work

Social tagging is subjective, orderless and noisy which restrict the use of tags in many related applications. In this paper, we propose a method named *Tag-tag* to derive a representative tag which best describes the visual content of a given image. For this purpose, we exploit the semantic relationship between visual content of a given image and its tags. Our method is completely independent of visually similar images, their associated tags and is not limited by any set of Flickr tag groups. We further use the existing state of the art pretrained machine learning models for object detection, thus does not require building anything from the scratch.

Our experiments on a set of 169 Flickr images, demonstrate the efficiency of proposed method. We identify two possible reasons where representative tag was not found as: 1) Representative tag itself was missing in the user given list of tags 2) Representative tag got filtered because it was not found in Wordnet dictionary or had no pretrained vector in word2vec-google-news-300 model, which we recognize as the one limitation of our method. For future work, we aim to extend our dataset of images, conduct comprehensive comparison with existing methods and check the applicability of proposed method with other social tagging services.

References

1. Li, X., Snoek, C.G., Worring, M.: Learning social tag relevance by neighbor voting. IEEE Trans. Multimedia **11**(7), 1310–1322 (2009)

2. Sigurbjörnsson, B., Van Zwol, R.: Flickr tag recommendation based on collective knowledge. In: 17th International Conference on World Wide Web, pp. 327–336 (2008)
3. Kennedy, L., Naaman, M., Ahern, S., Nair, R., Rattenbury, T.: How flickr helps us make sense of the world: context and content in community-contributed media collections. In: ACM International Conference on Multimedia, pp. 631–640 (2007)
4. Liu, D., Hua, X.S., Yang, L., Wang, M., Zhang, H.J.: Tag ranking. In: 18th International Conference on World Wide Web, pp. 351–360 (2009)
5. Wu, L., Hua, X.S., Yu, N., Ma, W.Y., Li, S.: Flickr distance. In: 16th ACM International Conference on Multimedia, pp. 31–40 (2008)
6. Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: a large data set for nonparametric object and scene recognition. IEEE Trans. Pattern Anal. Mach. Intell. **30**(11), 1958–1970 (2008)
7. Shamma, D.A., Shaw, R., Shafton, P.L., Liu, Y.: Watch what I watch: using community activity to understand content. In: International Workshop on Multimedia Information Retrieval, pp. 275–284 (2007)
8. Jin, Y., Khan, L., Wang, L., Awad, M.: Image annotations by combining multiple evidence & wordnet. In: 13th Annual ACM International Conference on Multimedia, pp. 706–715 (2005)
9. Wang, C., Jing, F., Zhang, L., Zhang, H.J.: Image annotation refinement using random walk with restarts. In: ACM International Conference on Multimedia, pp. 647–650 (2006)
10. Wang, C., Jing, F., Zhang, L., Zhang, H.J.: Content-based image annotation refinement. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
11. Wu, L., Yang, L., Yu, N., Hua, X.S.: Learning to tag. In: 18th International Conference on World Wide Web, pp. 361–370 (2009)
12. Zhuang, J., Hoi, S.C.: A two-view learning approach for image tag ranking. In: Fourth ACM International Conference on Web Search and Data Mining, pp. 625–634 (2011)
13. Kennedy, L.S., Chang, S.F., Kozintsev, I.V.: To search or to label? Predicting the performance of search-based automatic image classifiers. In: ACM International Workshop on Multimedia Information Retrieval, pp. 249–258 (2006)
14. Ames, M., Naaman, M.: Why we tag: motivations for annotation in mobile and online media. In: SIGCHI Conference on Human Factors in Computing Systems, pp. 971–980 (2007)
15. Lindstaedt, S., Mörzinger, R., Sorschag, R., Pammer, V., Thallinger, G.: Automatic image annotation using visual content and folksonomies. Multimedia Tools Appl. **42**, 97–113 (2009)
16. Zhao, Y., Zha, Z.-J., Li, S., Wu, X.: Which tags are related to visual content? In: Boll, S., Tian, Qi., Zhang, L., Zhang, Z., Chen, Y.-P. (eds.) MMM 2010. LNCS, vol. 5916, pp. 669–675. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11301-7_67
17. Lindstaedt, S., Pammer, V., Mörzinger, R., Kern, R., Mülner, H., Wagner, C.: Recommending tags for pictures based on text, visual content and user context. In: IEEE International Conference on Internet and Web Applications and Services, pp. 506–511 (2008)
18. Hu, Y., Gao, S., Janowicz, K., Yu, B., Li, W., Prasad, S.: Extracting and understanding urban areas of interest using geotagged photos. Comput. Environ. Urban Syst. **54**, 240–254 (2015)
19. https://tfhub.dev/google/faster_rcnn/openimages_v4/inception_resnet_v2/1. Accessed 14 Oct 2023
20. https://www.tensorflow.org/datasets/catalog/open_images_v4. Accessed 10 Oct 2023
21. https://storage.googleapis.com/openimages/2018_04/bbox_labels_600_hierarchy_visualizer/circle.html. Accessed 10 Oct 2023
22. https://wordnet.princeton.edu/. Accessed 10 Oct 2023
23. https://radimrehurek.com/gensim/models/word2vec.html. Accessed 10 Oct 2023
24. https://www.nltk.org/. Accessed 10 Oct 2023
25. https://www.geeksforgeeks.org/nlp-wupalmer-wordnet-similarity/. Accessed 10 Oct 2023
26. https://en.wikipedia.org/wiki/Word2vec#:~:text=Word2vec%20is%20a%20technique%20for,words%20for%20a%20partial%20sentence. Accessed 10 Oct 2023

27. https://towardsdatascience.com/word2vec-explained-49c52b4ccb71. Accessed 10 Oct 2023
28. https://www.turing.com/kb/guide-on-word-embeddings-in-nlp. Accessed 10 Oct 2023
29. https://medium.com/nlplanet/text-similarity-with-the-next-generation-of-word-embeddings-in-gensim-466fdafa4423. Accessed 10 Oct 2023
30. https://www.tensorflow.org/tutorials/text/word2vec. Accessed 10 Oct 2023
31. https://radimrehurek.com/gensim/intro.html. Accessed 10 Oct 2023
32. https://tedboy.github.io/nlps/generated/generated/gensim.models.Word2Vec.similarity.html. Accessed 10 Oct 2023
33. https://photutorial.com/flickr-statistics/. Accessed 10 Oct 2023
34. https://storage.googleapis.com/openimages/2018_04/bbox_labels_600_hierarchy.json. Accessed 10 Oct 2023
35. https://en.wikipedia.org/wiki/Named-entity_recognition. Accessed 10 Oct 2023
36. Liu, W., Ruan, Y., Cai, X., Chen, H.: Social image tag relevance learning based on pixel voting. In: International Conference on Computer Science and Application Engineering (CSAE) (2017). ISBN: 978-1-60595-505-6