

# Image Tag Core Generation

Olga Kanishcheva<sup>1</sup>[0000-0002-4589-092X], Olga Cherednichenko<sup>1</sup>[0000-0002-9391-5220]  
and Natalia Sharonova<sup>1</sup>[0000-0002-8161-552X]

<sup>1</sup> National Technical University “Kharkiv Polytechnic Institute”, Kharkiv, Ukraine  
kanichshevaolga@gmail.com, olha.cherednichenko@gmail.com and  
nvsharonova@ukr.net

**Abstract.** In this paper, we explore the task of tag aggregations or merge of tags meanings for the video and image. In our work, based on our previous research we try to merge tag meanings of video files. We present the result of our experiments using word2vec and clustering algorithms. For our experiments, we use the auto-tagging program from Imagga company as the generating program. As data, we use 5 videos which were split into shots for future processing. Our experiments showed that such clustering algorithms as k-means and Affinity propagation could not be used for aggregation tag meanings. We used word2vec model from spaCy software library and combined the similarity score with score from the auto-tagging program. Our results are not very excellent but better than for clustering algorithms. We received  $F_{\text{mean-measure}} = 0.62$ . For a detailed analysis of this task, we need to create a dataset with human annotations. It will help to evaluate the  $F_{\text{mean-measure}}$  of our approach more precision.

**Keywords:** Image Description, Video Description, Image Tags, Video Tags, Natural Language Processing, Aggregation of Video Tags, Aggregation of Image Tags Meanings, Word2vec, Clustering.

## 1 Introduction

There is a lot of content variety on the Internet and it grows drastically. Nowadays we can discover a large number of video content and various images that are provided by social networks, professional stock image marketplaces, scientific communities, and other sources. The presence of a large number of video content and various images causes interest in the tasks of automatic text generation from images or video series. Popular tasks include creating subtitles, as well as creating a sentence or phrase based on certain visual or image information. In this context, image processing and video processing are very close to each other and can use similar approaches because the video can be divided into slots where each slot represents an image.

Generating images into text is an important topic in artificial intelligence, which is associated with pattern recognition, computer vision, and natural language processing. From the point of view of natural language processing, such tasks as image tagging,

selecting keywords, evaluating the weight and relevance of keywords, generating sentences and text, etc. are of quite an interest.

One of our goals is the construction of a system that optimizes the number of tags describing video resources, without any loss of sense. We have started our research by analyzing systems that generate descriptions for video and images and explored the main problems of this task [1]. In our previous work [1], we concentrated on the problem of keywords aggregation into a single description of the object. Multimedia collections integrate electronic text, graphics, images, sound, and video. Tags that characterize, describe or refer to categories in certain classifications usually annotate their objects. These tags help to distinguish the objects and often form folksonomies: user-generated categories for organizing digital content. In the work [1], we showed how works the preprocessing stage for tag optimization of keywords sets for video fragments works, using NLP techniques, lexical resources to tag aggregation.

The main purpose of this paper is to investigate the key factors that influence the similarity of the keywords, which describe an image or video slot. In order to achieve our goal we make the experiment with tag core creating based on using the auto-tagging program, the semantic words distance and clustering algorithm.

The paper is organized as follows: Section 2 discusses related work and similarity metrics for aggregation of word meaning, similarity measures and algorithms applied in our experiments. The results and evaluation using different metrics and algorithms are reported in Section 3. Finally, in Section 4 we briefly sketch future work and present the conclusion.

## **2 Background and Related Work**

Recent years are characterized by the development of research in the field of creating descriptions and keywords or tags for images and videos. Both large companies, such as Google and Microsoft, and small ones that work in certain areas, for example, Clarifai (clarifai.com) or Imagga (imagga.com), are engaged in this task. It can also be noted that certain prerequisites have been created in this area and preliminary studies are being conducted, which determines such an intensive development. For example, special image collections were created (e.g. ImageNet, Microsoft COCO, etc.). All this has allowed achieving by Google Brain researchers automatically create captions that can accurately describe images. The authors of [2] provide a number of successful examples of the operation of this algorithm. Microsoft also has excellent results in this area.

The task of evaluating the word similarity is important in the semantic processing of image-related texts. Based on state-of-the-art we found out that researchers study this problem from two perspectives. Firstly, this is the problem of generating text from an image [3]. Secondly, it is the problem of images generating from natural language [4-7]. Analysis of publications shows the relevance of the problem statements [3, 6, 7]. Many authors note that existing approaches generate text descriptions from a sequence of images automatically. However, such construction of sentences bases on texts roughly concatenation, which leads to the problem of generating se-

mantically incoherent content. We can underline that the image-to-text generating is still an unsolved problem.

Another task that many researchers are working over is generating an image based on a part of the text. Despite some progress in this area, a number of issues are still open. The authors of [4] study the existent state of the art of models. They note that recent progress has been made using Generative Adversarial Networks (GANs). Generative adversarial networks, driven by simple textual descriptions of images, are capable of generating realistic-looking images [5]. However, current methods still struggle to generate images based on complex image captions from a heterogeneous domain. In addition, quantitatively evaluating these text synthesis models is a real challenge due to most assessment metrics only evaluate image quality and do not evaluate the correspondence between the image and its caption. The authors [5] propose the approach to solve the issue based on a new evaluation metric.

Several papers studying particular semantic similarity evaluation metrics [8, 9]. Semantic similarity between word pairs has become the most common evaluation benchmark for word embeddings [10, 11]. A large amount of research on semantic textual similarity is focused on creating modern embeddings. In paper [8] is figured out that the inclusion of semantic information in any similarity measures improves the efficiency of the similarity measure and provides human interpretable results for further analysis. Authors [9] note that little attention was paid to similarity measures. The cosine similarity is used in the majority of cases. Paper [9] illustrate that for all common word vectors, cosine similarity is essentially equivalent to the Pearson correlation coefficient, which provides some justification for its use. In the paper [10] report experiments with a rank-based metric for word embeddings, which performs comparably to vector cosine measure. Researchers suggest that rank-based measures can improve clustering quality. The analysis shows that many authors note the shortcomings of the cosine measure in solving problems of assessing the similarity of words and texts.

The study of state-of-the-art shows that in tasks of semantic proximity succeeded the vector models. Lacking standardized evaluation methods for vector representations of words, the NLP community relies on word similarity tasks. The paper [12] notes that the recent methods perform in capturing semantic and syntactic regularities using vector arithmetic, but the origin of these regularities has remained opaque. They analyze and make explicit the model properties needed for such regularities to emerge in word vectors. Paper [13] presents several problems associated with the evaluation of word vectors on word similarity datasets and summarize existing solutions. The study suggests that the use of word similarity tasks for evaluation of word vectors is not sustainable and calls for further research on evaluation methods [13]. Authors of [14] conduct an evaluation of a large number of word embedding models for language processing applications. Based on the six models of word embedding they provide experimental results and estimate the performance. The paper [15] is devoted to neural language models for word embeddings that capture rich linguistic and conceptual information. In paper [16] an unsupervised method to generate Word2Sense word embeddings is considered. Authors conclude that on computational NLP tasks, Word2Sense embeddings compare well with other word embeddings generated by

unsupervised methods. As a result of the literature review, we found out the impact of word embeddings based methods on the word similarity evaluation. The most popular similarity metric in semantic models is the vector cosine. Compared to Euclidean distances, the cosine measure is normalized and is robust to the scaling effect. However, the limitation of this metric is that it does not take into account that some dimensions might be more relevant for the semantic content. This leads to the necessity of using and studying alternative metrics.

In our work, we try to apply different clustering algorithms based on different metrics. One of the most popular technique word2vec to unification image tags meanings problem is also used. We consider our experiments and results for image tag aggregation with using all these methods.

### 3 Experiments

#### 3.1 Data Set Description

We used five fragments of films for our experiments, they are *Batmobile*, *FC Barcelona*, *Hunger Games*, *Meghan Trainor*, *Remi Gaillard*. All these films were divided into shots. The structure of these files you can see on the Table 1. We received sets of tags for all video shots using the auto-tagging program from Imagga company (<https://imagga.com/>).

**Table 1.** Information about test data sets.

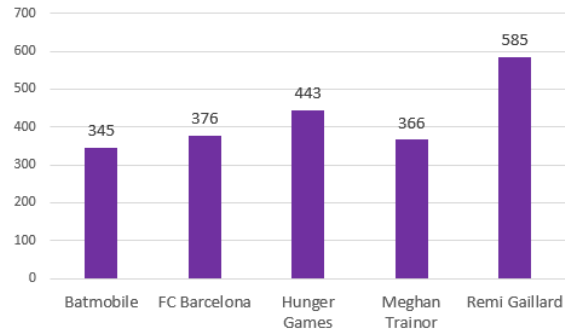
Name of film	Number of shots	Number of tags
<i>Batmobile</i>	24	1,524
<i>FC Barcelona</i>	57	1,570
<i>Hunger Games</i>	60	1,555
<i>Meghan Trainor</i>	154	6,161
<i>Remi Gaillard</i>	58	1,936

After removing all duplicate tags, we receive the set of tags that are shown in Fig. 1. On the stage of removing all duplicate tags, we delete only repeating words without any pre-processing or semantic analysis.

#### 3.2 Experiments with Clustering

Our task was to create a core of tags for each video without sense missing. Initially, we present our experiments with clustering algorithms. In our case we don't know the number of clusters therefore we need to use a clustering algorithm that can take into account this feature. For our experiments we use Affinity propagation algorithm. It is a clustering algorithm based on the concept of "message passing" between data points [17]. Unlike clustering algorithms such as k-means propagation does not require the number of clusters to be determined or estimated before running the algo-

rithm. Affinity propagation finds "exemplars" members of the input set that are representative of clusters [17].



**Fig. 1.** A set of tags for five films after removing all duplicate tags.

For defining the similarity measure we use Levenshtein distance. It's a string metric for measuring the difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits required to change one word into the other.

Table 3 shows some centroids and corresponding tags for them. We show only five centroids for each film. Table 4 indicates the final number of core tags and examples of core tags.

**Table 3.** The list of centroid and tags in these centroids (on the example of five clusters).

Name of film	Clusterization	
	Centroid	Tags
<i>Batmobile</i>	light	bright, light, night
	vehicle	convertible, device, mechanical, office, recycle, vehicle
	colour	club, collar, color, colorful, colour, computer, ecology
	partners	happiness, letters, partner, partners, partnership, patriotism
	30s	20s, 30, 30s, 3d, 40s
<i>FC Barcelona</i>	friends	field, friends, friendship, greenhouse
	curtain	cartoon, currency, curtain, fountain, portrait, urban
	celebrate	beverage, celebrate, celebration, corporate
	cloud	child, close, closeup, clothes, cloud, crowd, tagcloud
	active	active, activity, attractive, autumn, fantasy
<i>Hunger Games</i>	broom	bathroom, broom, brush, room
	print	drink, grain, parquet, plant, pretty, print, spring, think
	health	adult, health, healthcare, healthy
	businesswoman	businessman, businesspeople, businesswoman
	blind	basin, bird, blind, blonde, lines, smiling
<i>Meghan Trainor</i>	person	expression, person, season, yellow
	fashion	family, fashion, fashionable, passion

	hat	cap, coat, fit, hair, happy, hat, head, hot, lab, shape, two
	health	health, healthcare, healthy, heart, vitality
	hairpiece	hairpiece, happiness, timepiece
<i>Remi Gaillard</i>	mobile	automobile, couple, mobile, model, movable
	minivan	ibizan, minibus, minivan
	man	dane, doberman, german, human, lab, lawn, man, men, tan
	sand	bend, giant, hand, hands, island, plant, sand, sandbar
	terrier	barrier, retriever, tennis, terrier

**Table 4.** The list of centroid and tags in these centroids (on the example of five clusters).

Name of film	Finally tags (Total number of core tags/examples of core tags)
<i>Batmobile</i>	<b>54</b> / light, vehicle, digital, people, backdrop, decoration, style, paper, man, businessman, auto, traffic, sport, automotive, adult, hand, friends, relationship, finance, partners, cart, etc.
<i>FC Barcelona</i>	<b>58</b> / ball, metal, celebrate, advertise, cloud, fare, association, packet, contestant, soccer, tree, outdoor, grass, pole, active, ring, cuisine, eating, team, friends, boy, looking, place, fence, sit, etc.
<i>Hunger Games</i>	<b>76</b> / black, space, flower, decoration, element, style, water, cereal, agriculture, crop, country, health, sun, land, cloud, old, grunge, glass, ice, advertise, package, ornament, association, print, businesswomen, etc.
<i>Meghan Trainor</i>	<b>65</b> / light, color, person, hat, health, hands, sensual, dress, child, style, internet, boy, suit, gold, relaxing, cream, eating, water, clothing, girl, spring, active, dance, desire, eating, house, etc.
<i>Remi Gaillard</i>	<b>110</b> / mobile, trailer, tow, tree, man, mountain, horizon, natural, water, card, dog, holiday, active, sport, children, sun, animal, sea, terrier, swimming, enjoyment, health, romantic, rest, destination, etc.


As Table 3 shows the results are not very good, the algorithm merges such words as *retriever* and *tennis* or *bird* and *blond*. Therefore, this algorithm is not appropriate for our task.

But for good quality evaluation, we need etalon to which we can compare our results. Unfortunately, we don't have a dataset with correct core tags for each image from our collection. However, we can take one image and humans will evaluate tags and separate only the most important tags for this image. The image with initial and human tags is presented in Table 5.

The clustering results were confirmed by our example. Only 3 words of 17 match with the opinion of experts. These words are like *outdoor*, *fashion* and *face*. To improve the results of clustering, we used the word2vec model to represent tags and then clustered using the Euclidean metric. The results were also pretty bad.

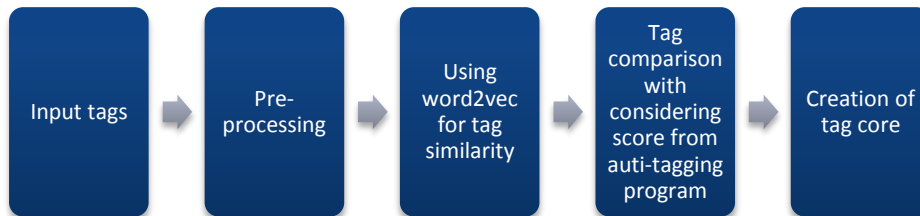
As a result of the experiments, we decided not to use clustering but proposed our own algorithm for combining tags within the meaning. The description of the algorithm and the results are shown below.

**Table 5.** The image with initial, human tags and tags from clustering algorithm.

	<i>Initial tags from auto-tagging program</i> tourist 52.69%, person 48.47%, traveler 31.54%, pedestrian 31.50%, attractive 30.69%, people 30.41%, adult 30.39%, street 28.71%, pretty 26.57%, cute 25.67%, smile 25.25%, outdoor 24.92%, business 23.29%, city 23.02%, building 22.73%, urban 22.57%, happy 21.12%, fashion 20.25%, lifestyle 19.78%, women 18.96%, man 18.68%, lady 18.58%, professional 18.05%, ... (total 102 tags)
	<i>Human tags (core of tags)</i> tourist, person, attractive, street, outdoor, business, fashion, women, student, face, bag, walking, communication (total 13 tags)
<i>Tags from clusterization algorithm</i> pretty, outdoor, fashion, man, face, model, walking, businessman, coat, education, style, architecture, successful, university, phone, shopping, travel (total 17 tags)	

### 3.3 Experiments with Similarity Measure

The most effective metric for determining the similarity of words is the word2vec model. We use it as a base for our algorithm. The whole algorithm is shown in Fig. 2

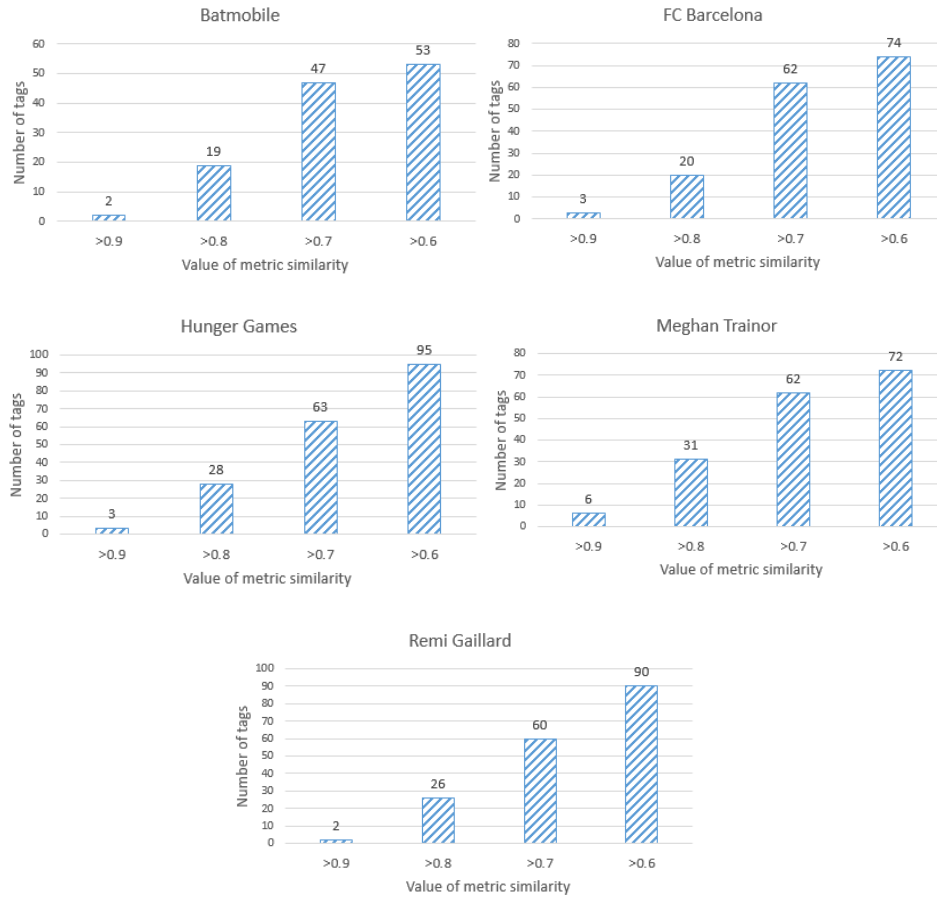


**Fig. 2.** The algorithm for a finding of tag core with saving the meaning of tags.

The proposed algorithm is not complicated, but it takes into account the semantic similarity of words using word2vec and the weights that the tags have after the auto-tagging program.

We take the word2vec model from spaCy software library and compare each tag with others on the list. If a tag does not have strong links with other words, we delete it. Otherwise, we keep tag with a high score in the final set.

The results of these experiments are presented on Fig. 3. For further experiments, we took a similarity value of more than 0.8. It was selected based on an analysis of the tags received, as well as their number. As Fig. 3 shows, we receive a fairly short list of tags when the value of the similarity measure is more than 0.8. The top of the tag lists with a similarity measure of more than 0.8 is presented in Table 6.



**Fig. 3.** The results after using similarity measure for tags.

**Table 6.** The top of core tags (for similarity value > 0.8).

Name of film	Top of tags (Top 10)	
<i>Batmobile</i>	black	work
	men	money
	success	tasty
	sport	smiling
	hand	clothing
<i>FC Barcelona</i>	color	clothing
	women	working
	playing	child
	hand	interior
	smiling	dinner
<i>Hunger Games</i>	black	cereal



	flower	agriculture
	element	summer
	hands	yellow
	wheat	meal
<i>Meghan Trainor</i>	color	women
	blond	lovely
	smile	child
	eyes	girls
	glamour	male
<i>Remi Gaillard</i>	walk	tree
	smile	tranquil
	women	summer
	outdoor	scenic
	playing	sport

From the analysis of our tag list, we defined that these lists need to refinement. For example, we can use a part-of-speech tagger for the determiner part of speech and stay only nouns for the core tag set. Also, such tags as *playing* and *sport* could be merged into one concept.

For the evaluation, we used METEOR metric. Unigram precision  $P$  is calculated as

$$P = \frac{m}{w_a}, \quad (1)$$

where  $m$  is the number of unigrams in the candidate for tag core which are also found in the human list of tag core, and  $w_a$  is the number of unigrams in the list from our algorithm. Unigram recall  $R$  is computed as:

$$R = \frac{m}{w_h}, \quad (2)$$

where  $m$  is as mentioned above, and  $w_h$  is the number of unigrams in the human list of tag core. Precision and recall are combined using the harmonic mean in the following fashion, with recall weighted 9 times more than precision:

$$F_{mean} = \frac{10PR}{R+9P}. \quad (3)$$

For example in Table 5 we received  $P=0.58$ ,  $R=0.54$ , and  $F_{mean}=0.62$ . The final list of tag core for this example is “*smile, outdoor, business, women, work, student, face, bag, one, success, fashion, education*”.

## 4 Conclusions

In this paper, we presented the method for the unification of image tag meaning and show that clustering algorithms aren't effective for this task. In this work, we provided how the word2vec works for tag aggregation of keywords sets for video fragments, using the score from auto-tagging program. We presented statistical in-

formation about our experiments and results. The experiments and results showed that we need to improve our approach to tag core creation. For a qualitative analysis of the proposed approach, it is necessary to create a “gold” collection with sets of tags from users and then evaluate the accuracy of the proposed method.

## References

1. Kanishcheva, O., Sharonova, N.: Image and Video Tag Aggregation. In: Supplementary Proceedings of the Seventh International Conference on Analysis of Images, Social Networks and Texts (AIST 2018), pp. 161-172. Moscow, Russia (2018).
2. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and Tell: Lessons learned from the 2015 MSCOCO Image Captioning Challenge. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, 39, 4, pp. 652-663. (2017).
3. Wang, R., Wei, Z., Li, P., Shan, H., Zhang, J., Zhang, Q., & Huang, X.: Keep it Consistent: Topic-Aware Storytelling from an Image Stream via Iterative Multi-agent Communication. arXiv preprint arXiv:1911.04192. (2019).
4. Bodnar, C.: Text to Image Synthesis Using Generative Adversarial Networks. arXiv preprint arXiv:1805.00676. (2018).
5. Hinz, T., Heinrich, S., Wermter, S.: Semantic Object Accuracy for Generative Text-to-Image Synthesis. arXiv preprint arXiv:1910.13321. (2019).
6. Agnese, J., Herrera, J., Tao, H., Zhu, X.: A Survey and Taxonomy of Adversarial Neural Networks for Text-to-Image Synthesis. arXiv preprint arXiv:1910.09399. (2019).
7. Sommer, W., Iosifidis, A.: Text-to-image synthesis method evaluation based on visual patterns. (2019).
8. Sitikhu, P., Pahi, K., Thapa, P., Shakya, S.: A Comparison of Semantic Similarity Methods for Maximum Human Interpretability. arXiv preprint arXiv:1910.09129. (2019).
9. Zhelezniak, V., Savkov, A., Shen, A., Hammerla, N. Y.: Correlation Coefficients and Semantic Textual Similarity. arXiv preprint arXiv:1905.07790. (2019).
10. Santus, E., Wang, H., Chersoni, E., Zhang, Y.: A rank-based similarity metric for word embeddings. arXiv preprint arXiv:1805.01923. (2018).
11. Soler, A. G., Apidianaki, M., Allauzen, A.: Word Usage Similarity Estimation with Sentence Representations and Automatic Substitutes. arXiv preprint arXiv:1905.08377. (2019).
12. Pennington, J., Socher, R., Manning, C. D.: GloVe: Global Vectors for Word Representation, <https://nlp.stanford.edu/pubs/glove.pdf> (2014).
13. Faruqui, M., Tsvetkov, Y., Rastogi, P., Dyer, C.: Problems With Evaluation of Word Embeddings Using Word Similarity Tasks, pp. 30-35. (2016).
14. Wang, B., Wang, A., Chen, F., Wang, Y., Kuo, C.: Evaluating word embedding models: Methods and experimental results. APSIPA Transactions on Signal and Information Processing, 8, E19 (2019).
15. Hill, F., Cho, K., Jean, S., Devin, C., Bengio, Y.: Embedding word similarity with neural machine translation. arXiv preprint arXiv:1412.6448. (2014).
16. Panigrahi, A., Simhadri, H. V., Bhattacharyya, C.: Word2Sense: Sparse Interpretable Word Embeddings. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5692-5705. (2019).
17. Frey, B. J., Dueck, D.: Clustering by passing messages between data points. Science 315 (5814), 972-976 (2007).