International Journal of Engineering Inventions
ISSN: 2278-7461, www.ijeijournal.com
Volume 1, Issue 2 (September 2012) PP: 08-14

Enriching Image Labels by Auto-Annotation: Simple but Effective and Efficient

Weilei Wang, Song Han ,Ying Zhou, Yong Jia, Yong Chen and Qingxiang Cao
Roboo Inc.

Abstract—Image search is popular and welcome currently, however the rareness of labels available may influence the performance of most commercial search engines built on traditional vector space model. To bridge the semantic gap between text labels and text queries, we propose one simple approach to extend existing labels using thesaurus. Different from naive approach where synonyms of tokens contained in a label are simply added, we employ three metrics to filter out those non-appropriate candidate labels (combination of synonyms): user log, semantic vector and context vector. The experiment results indicate that the proposed method has impressive performance, in term of effectiveness and efficiency.

Keywords— auto-annotation, label extension, effective, efficient.

I. INTRODUCTION

The amount of multimedia resource search request, like image, video or music, has been increasing on most commercial search engine. For example, based on our own data, the PVs (page-view) contributed by image search occupy over 10% of our total flow on Roboo® (http://m.roboo.com, a popular mobile search engine in China). The ratio will be much higher if we count all non-text resource retrieval service. This is exactly consistent with what we understand about the users of mobile search: they prefer entertainment resource during their non-office time [11]. In the remaining text, we will focus the discussion on image annotation, but what proposed here can be applied to video or music as well.

However, there is one critical challenge posed on us regarding these multimedia resources, that is their labels are normally quite short compared with Web document. One study on our own image repository indicates that the average length of image labels is 5.18(characters). Besides, the average query length on image is about 3.52 [11], which worsen the performance of image search further since obvious there is a gap between image labels and textual queries. Since manually annotating images is a very tedious and expensive task, image auto-annotation has become a hot research topic in recent years.

Although many previous works have been proposed using computer vision and machine learning techniques, image annotation is still far from practical[12]. One reason is that it is still unclear how to model the semantic concepts effectively and efficiently.

The other reason is the lack of training data, and hence the semantic gap can not be effectively bridged [12]. Therefore, till now, almost all known running commercial image search engines depend on labels attached to images, including ours. The direct result of this search model is that:

- Hit missing is not avoidable due to the short labels available;
- One image can only serve very few queries. For example, with the example shown in Fig 1., the image has label "性感车模" (Sexy auto model). Based on current VSM-based (Vector Space Model) search model, the image will and only will be retrieved upon query of "性感", "车模", or "性感车模". Though "性感车模" normally means implicitly "美女" (pretty girl, one quite popular query as indicated by our log), this example image will not appear in the query result of "美女" due to the short of label "美女".

Fig. 1. One image example with label "性感车模" (Sexy auto model)

Generally, these two aspects together construct the so-called semantic map between target resources and users' initiatives, as shown as Path 3 in Fig 2. Ideally, we hope the query can directly be matched by some resources, like Path 3, but we are not always so fortunate in real world. Two possible options to solve the problem:

1. Query extension (Path 1)[1,8].
2. Label extension (Path 2).

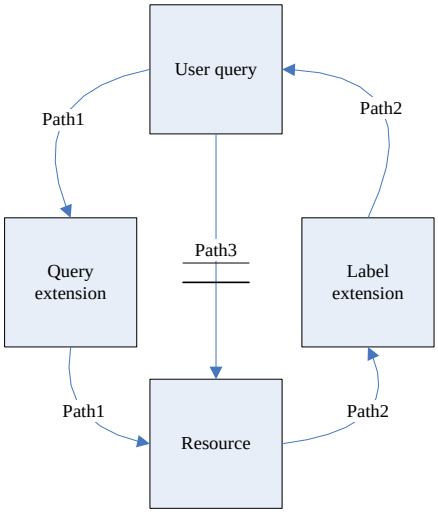

Fig. 2. The path connecting query and resource

In this paper, we choose Path 2 since it is believed to more "secure" approach than Path 1 based on the following considerations:

1. The extension work normally is done offline, so efficiency is not so critical a factor while we choose candidate approaches. Normally, better methods require more computing resource based on common sense;
2. Given those zero-hit queries, we may "attach" them purposely to some resources via some appropriate extension algorithm;
3. By preparing the extended labels offline first, the online response speed would not be influenced;
4. No matter how well one query extension may work, it is still built on the labels; Rich labels play as the basis for more applications.
5. Extended labels may evolve, added or removed from an image, with time on to meet specific application requirements.

In this paper, label extension is based on thesaurus. However, different from naive approach, we propose three metrics to filter out those non-appropriate labels. In Section 2, the proposed method is explained in detail. In Section 3, experiment results are presented to demonstrate the effectiveness and efficiency. Short conclusion and future work are shared in Section 4.

II. LABEL EXTENSION

In this section, how labels are extended by us will be discussed in detail. The overall procedure is shown in Fig 3, and each step will be discussed in the following sub-sections.

2.1 Word Segmentation (Tokenization)

In English, there is space between word and word. However, in Chinese, one word may contain one or more than one Chinese character, and all characters appear in concatenation in a sentence. Therefore, Chinese text has to be segmented or tokenized first to extract words before they can be analyzed or indexed [2,3]. In Fig 3, the input "Label" refers to the original label available to an image, and "Word Segmentation" module outputs a series of tokens appearing as their relative order in the original label string. Repeated tokens are left unchanged to keep the original semantic meaning. These tokens play as the input for next step – Get candidate using synonyms.

2.2 Collect Candidates Using Thesaurus

Thesaurus[4,7] plays as an important role here because we have to select the candidates very carefully to keep the original semantic meaning unchanged. Synonym is believed an ideal candidate considering that normally thesaurus is constructed in a serious way normally. The appropriateness of synonym is believed as a solid basis for the success of our method. Actually, thesaurus technique is widely applied in information processing and information retrieval, such as detecting text similarity over short passages [5], Query expansion using synonymous terms [6], et al.

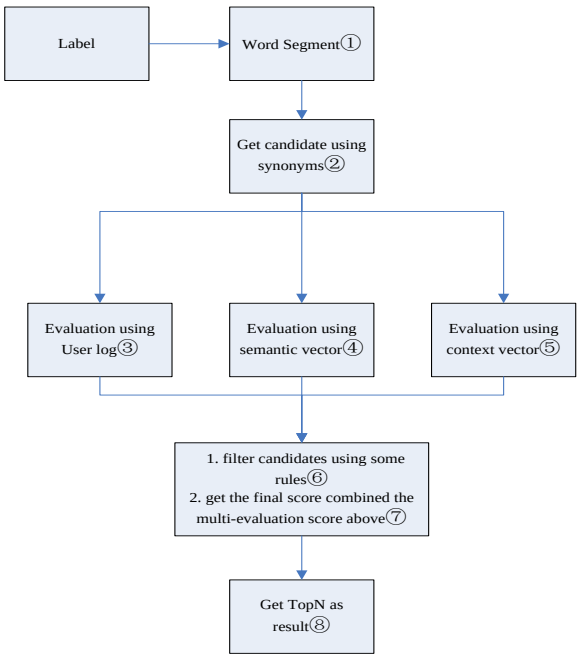

Fig. 3. Overall procedure of label extension proposed by us

Here, given the sequence of tokens, the replacement with synonym is applied to each individual token separately and independently.This candidate construction procedure can be formalized as below:

1. With one label L, its corresponding token sequence as output by word segmentation component is denoted as $TS_L = \{T_1, T_2, ..., T_k\}$, and T_i represents one token.
2. With each T_i, we get its synonyms via reference to one open thesaurus edited and maintained by[7]; all the synonyms plus T_i itself construct one set denoted as $Syn(T_i)$;
3. With one instance selected from each $Syn(T_i)$, and concatenate them together, we get one candidate label. All possible candidates are denoted with $CL = \{CL_i = C_1C_2...C_k \mid C_j \in Syn(T_j), i = 1...k; j = 1..L\}$ where $L = \Pi_{i=1}^{k} \mid Syn(T_i) \mid$, and $\mid Syn(T_i) \mid$ is the cardinality of set $Syn(T_i)$.

With this method, many more possible labels are generated. For example, assuming one label “性感车模”, it is segmented into two tokens “性感”+“车模” first. By referring to thesaurus, we notice that there are five synonyms corresponding to “车模” , and four synonyms corresponding to “车模”. Then, we may have 30 labels ((5+1) * (4+1) = 30), e.g. “性感模特”, 美丽车模 etc. However, some labels may be meaningless by combining the candidate synonyms, though each synonym token itself is acceptable. For example, one possible extension of “我的女人” (my girl), like “我的女性” (my female) may not be appropriate although “女性” (female) may be a acceptable synonym of “女人” (girl or woman). From the example here, it is noticed that deciding whether one synonym alone is appropriate or not may not be easy or possible. The more practical and reasonable approach is to evaluate each candidate label separately, though it results with more computing work. How it can be achieved will be discussed in the following three sub-sections.

2.3 Evaluation Using User Log

User log records the users' queries and the corresponding frequencies, and it not only does full justice to the interests and behaviors of users but is able to allow us to evaluate the appropriateness of one candidate label:

1. $CL = \{CL_i = C_1C_2...C_k \mid C_j \in Syn(T_j), i = 1...k; j = 1..L\}$ contains all the candidate labels as output by the end of last step.

$Log = \{(q_i, f_i) \mid i = 1..N\}$. represents the log corpus, where q_i is the query and f_i is its frequency being issued;

2. Given the log data and generated candidate labels, we filter those not found in The log, with $CL' = \{CL_i \mid C_i \in CL \text{ and } C_i \in Log\}$ left. In the current version, we require COMPLETE match while determining filtering one candidate out or not. Actually, this filtering is also required by Step 4 and Step 5 as indicated in Fig 3;

3. With each $CL_i \in CL'$, one score is assigned, denoted as $SLog(CL_i)$ and $SLog(CL_i) = f_i / \sum f_j$.

Therefore, $SLog(CL_i)$ is the result of Step 3 in Fig 3. Based on our experiments,this is effective to filter some non-appropriate candidates, and give a reasonable score for each accepted candidate with the heuristics that the more frequently queried, the more important one candidate is. However, still there is "noise" exist in the remaining candidates, and we need do further filtering.

One bonus advantage of this method is that we may "create" images for some queries originally having no matched results by appending some more labels.

2.4 Evaluation Using Semantic Vector

Given a set of candidate labels, it is hard, or impossible, to simply tagging "right" or "wrong". For instance, "女人"(woman) has synonyms like "女性"(female) or "老婆"(wife). Though both of are acceptable, it does NOT mean that they are "equal" in meaning as compared to the original label "女人". In this section, we propose to rank the remaining candidates based on their semantic similarity relative to the original label.

Given two strings, like "春节愉快" and "新春快乐", they have the same meaning even though they share no any common token, which results in 0 similarity score in traditional vector space. To address this problem, a more reasonable as well as reliable measure about the closeness among labels is desired, and it is expected to use more comprehensive background information for reference, instead of simple term-wise happenings. Fortunately, the Web provides a potential rich source of data that may be utilized to address this problem, and modern search engine helps us to leverage the large volume of pages on the web to determine the greater context for a query in an efficient and effective manner.

Our solution to this problem is quite simple, and it is based on the intuition that queries which are similar in concept are expected to have similar context. For each query, we depend on the search engine to seek for a group of related and ranked documents, from which a feature vector is abstracted. Such feature vector contains those words that tend to co-occur with the original query, so it can be viewed as the corresponding context of the query, providing us more semantic background information. Then, the comparison of queries becomes the comparison of their corresponding feature, or context, vectors. Actually, we found that there were similar application of this method to determine the similarity between more objects on the Web [8,10].

Here, one candidate label or someone token contained is viewed as a query, and be submitted to a search engine to retrieve relevant documents from which the so-called "semantic background" knowledge could be abstracted and referred later. The procedure is formalized as follows:

Let q represent a query, and we get its feature vector (FV), denoted as $FV(q)$, with the following sequential steps:

1. Issue the query q to a search engine; for now, let us ignore the actual algorithms of the search engine and assume that the approach is generalizable to any engine that provides "reasonable" results;
2. Let $D(q)$ be the set of retrieved documents. In practice, we only keep the top k documents, assuming that they contain enough information, so $D(q) = \{d_1, d_2, ..., d_k\}$.
3. Compute the TF-IDF term vector tv_i for each document $d_i \in D(q)$, with each element as

$$tv_i(j) = tf_{i,j} \times \log(\frac{N}{df_j})$$

where $tf_{i,j}$ is the frequency of the jth term in d_i, N is the total number of documents available behind the search engine, and df_j is the total number of documents that contain the jth term. The TF-IDF is widely applied in the information retrieval (IR) community and has been proved work well in many applications, including the current approach.

4. Sum up $tv_i, i = 1..K$, to get a single vector. Here, for the same term, its IDF,i.e. $\log(\frac{N}{df})$, is known as identical in different tv_i , so the sum of their corresponding weights can be calculated;
5. Normalize and rank the features of the vector with respect to their weight (TF-IDF), and truncate the vector to its M highest weighted terms. What left is the target $FV(q)$.

After we get the feature vector for each query of interest, the measure of semantic similarity between two queries is defined as:

$$Sim(q_i, q_j) = FV(q_i) \bullet FV(q_j);$$

In our label extension, we evaluate the similarity between the original label L and someone candidate label CL from two aspects:

1. One is from the overall, with a score denoted as $Score_{All}$:

$$Score_{All} = Sim(L, CL)$$

2. The other is from the partial view with corresponding score denoted as $Score_{Partial}$:

$$Score_{Partial} = \max_i (Sim(T_i^L, T_i^{CL})\}$$

where T_i^L refers to the ith token in original label L , and T_i^{CL} refers to

the ith token in the candidate label CL (The candidate construction algorithm ensures that L and CL have same number of tokens, so the calculation here(has no problem);

3. Given a candidate label, CL , its final score is determined by

$$S_{sv}(CL_i) = \max(Score_{All}, Score_{Partial}) .$$

$S_{sv}(CL_i)$ therefore, is the output of Step 4 in Fig 3.

2.5 Evaluation Using Context Vector

Till now, two dimensions have been introduced to measure if one candidate label is acceptable or not, including query log and semantic information. However, it is noticed that due to the feature of our index repository and queries received (most about entertainment), the selection procedure is more or less biased. To weaken this influence so that those more serious or formal labels would not be filtered out, one additional dimension called Context Vector (CV) based on People's Daily news Corpus[9] is introduced. How a context vector is got and how to determine the similarity score based on context vectors are formalized as follows:

1. Given a label L , we find all the "contexts" containing L , $\{(T_i, L, T_{i+1}), \ldots\}$. In the current version, we limit the context length as 1, that is we only abstract $\{(T_1, L, T_2)\}$;

2. Given all contexts ready, $\{(T_1, L, T_2)\}$, we summarize all the happenings of someone token T_i that appears in the triple above, ignoring its position in the triple before or after L . This allows to get the so-called context vector of L , $CV(L) = \{tf_1, tf_2, ..., tf_m\}$ where tf refers to someone token's frequency and they are ranked in a decreasing order in term of frequency;

3. The similarity between the original label *L* and candidate label CL_i is $Sim(L, CL_i) = CV(L) \bullet CV(CL_i)$. For easy reference, this score is denoted as $S_{CV}(CL_i)$.

So, the score by Context Vector is similar to that by Semantic Vector, both depend on more some 3rd-party corpus to gain deeper understanding of the labels of interest. The only difference exists on how to construct the vectors for comparison. Table 1 lists two examples where the similarity scores based on two different vectors are listed for different pair of labels, and it indeed indicates that the two scores are complementary to some extend.

Table 1. **Two examples about Semantic Vector vs. Context Vector**

Example pair	Similarity based on Semantic Vector	Similarity based on Context Vector
愉快 vs 快乐 (both refer to happiness)	0.433	0.00
美女 vs 尤物 (pretty girl vs. stunner)	0.00	0.268

2.6 Reach the Output

Till now, we have introduced how to construct candidates given a label, and how to ranking the candidates based on different scores, S_{Log} , S_{SV} and S_{CV} . To get the final score of someone label, the three scores need to be combined in some manner. In the current version, weight is assigned to each score, and they are added linearly as:

$$Score = a_1 \times S_{Log} + a_2 \times S_{sv} + a_3 \times S_{cv}$$

To make them comparable and addable, each individual score is normalized before be summarized. Based on observation via lots of experiments and manually study, $a_1 = 0.8$, $a_2 = 1.3$ $a_3 = 1.0$ are chosen since satisfactory results are produced then.

III. EXPERIMENTAL STUDY

In our experiment, given each label, we apply the procedure shown in Fig 3 to get candidate labels, and some summary statistics information is presented here:

Table 2. **Summary information of experiments**

Total # of images in test repository	1,957,090
Total # of different labels	270,192
The ratio of image # to label #	7:1
Average length of label	5.18
Total time to process all images	3,296(sec.)
Total # of labels being extended finally (with at least one new candidate label output)	1,992
Total # of images influenced, i.e. with new label generated	132,5816

From Table 2, we could conclude that:

1. Many images share same labels (the ratio of image # to label # is 7:1);
2. Labels available are short;
3. The running performance is acceptable, costing less than 1 hour to process about two millions of images on a common PC (CPU 1.6 GHz; 2.0GB RAM);
4. Although there are only 1,992 labels been extended actually, about 1.3 millions of images (over two third) are influenced, which furthermore confirms the first point, i.e. many images have same labels. Actually, these labels are also frequently queried, which reflects the common preference among human beings, no matter image annotators or query submitters.

With the 1,992 labels got, four editors are employed to do manual checking. Finally 1405 labels are thought as appropriate, about 70.5% (1405/1992) acceptance ratio. Two primary causes may explain the non-appropriate extensions:

1. The quality of thesaurus relative to our application. Some synonyms may be acceptable in other fields,but not to us in mobile search application. It may be worthy of effort to build one thesaurus customized for our field on long terms;
2. Another problem is about polyseme. Although we carefully evaluate the appropriateness of one candidate based on query log, semantic vector and context vector, extra effort to further disambiguation is still desired.

IV. CONCLUSION

Image search is one popular service online. However, the rareness of annotation prevents is widely known as a challenge. In this paper, we propose a simple architecture is proposed to enrich the labels available for an image. Firstly, we construct a series of candidates for a given label based on simple rule and thesaurus. Then, three different dimensions of information are employed to rank the candidates, including user log, semantic similarity and context similarity. The experiments indicate that over 70% of generated labels are regarded as reasonable and acceptable. Although our discussion is based on image, the proposed method obviously can be applied to other multimedia resources suffering the similar problem, e.g. video or theme to stall on mobile phone. There is still much space to improve the underlying performance of our work. For example, we admit that filtering candidates by the fact if they appear in query log is somewhat too restrict. Partial containment relation or similarity between queries and labels may bring us more labels. Other information or metrics may be introduced and be included in selection also in applications.

REFERENCES

1. Mitra M, Singhal A, Buckley C. Improving automatic query expansion. In: Proceedings of the 21st Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval. Melbourne: ACM Press, 206-214 (1998)
2. Stefan Riezler, Yi Liu, Alexander Vasserman: Translating Queries into Snippets for Improved Query Expansion, In: Proceedings of the 22nd International Conference on Computational Linguistics (COLING'08) (2008).
3. Jianfeng GAO, Joshua Goodman, Mingjing Li and Kai-fu Lee: Toward a unified approach to statistical language modeling for Chinese. ACM Transactions on Asian Language Information Processing (TALIP), 1(1):3 – 33 (2002).
4. 梅家驹,竺一鸣，高蕴琦等编.同义词词林.上海：上海辞书出版社，1983.
5. Hatzivassiloglou V, et al.: Detecting text similarity over short passages: exploring linguistic feature combinations via machine learning. Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora. College Park, Maryland, 203-212 (1999).
6. Guo, Y., Harkema, H., & Gaizauskas, R: Sheffield University and the TREC 2004 genomics track: Query expansion using synonymous terms. In Proceedings of TREC, (2004).
7. http://blog.csdn.net/ganlantree/archive/2007/10/26/1845788.aspx
8. Kamvar, M. and Baluja, S.: Query suggestions for mobile search: understanding usage patterns. In: Proceedings of the SIGCHI conference on Human Factors in computing systems (CHI), (2008).
9. http://icl.pku.edu.cn/icl_groups/corpus/shengming.htm
10. Shunkai Fu, Michel C. Desmarais, Bingfeng Pi, Ying Zhou, Weilei Wang, Gang Zou, Song han, and Xunrong Rao. Simple but effective porn query recognition by k-NN with semantic similarity measure, In: Proceedings of APWeb/WAIM, LNCS 5446, pp. 3-14, (2009).

11. Fu, S.-K., Han, S., Zhuo, J.-H., Pi, B.-F., Zhou, Y., Demarais, M.C. and Wang, W.-L.: Large scale analysis of Chinese mobile query behavior. In: Proceedings of 7th International Conference on Information and Knowledge Engineering, Las Vegas, USA (2009).
12. Wang, X.-J., Zhang, L., Jing, F. and Ma, W.-Y.: AnnoSearch, Image auto-annotation by search. In: Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR), Near York, USA (2006).