Using Textual and Visual Processing in Scalable Concept Image Annotation Challenge

Alexandru Calfa, Dragoș Silion, Andreea Cristina Bursuc, Cornel Paul Acatrinei, Răzvan Iulian Lupu, Alexandru Eduard Cozma, Cristian Pădurariu, Adrian Iftene

UAIC: Faculty of Computer Science, "Alexandru Ioan Cuza" University, Romania
{alexandru.calfa, dragos.silion, andreea.bursuc, paul.acatrinei, razvan.lupu, eduard.cozma, cristian.padurariu, adiftene}@info.uaic.ro

Abstract. This paper describes UAIC[1]'s system built for participating in the Scalable Concept Image Annotation challenge 2015. We submitted runs both for Subtask 1 (Image Concept detection and localisation) and for Subtask 2 (Generation of Textual Descriptions of Images). For the first subtask we created an ontology with relations between concepts and their synonyms, hyponyms and hypernyms and also with relations between concepts and related words. For the second subtask, we created a resource that contains triplets (*concept*$_1$, *verb*, *concept*$_2$), where *concepts* are from the list of concepts provided by the organizers and *verb* is a relation between concepts. With this resource we build sentences in which *concept*$_1$ is subject, *verb* is predicate and *concept*$_2$ is complement.

Keywords: Text processing, Visual Processing, Text Generation.

1 Introduction

In 2015, UAIC group participated again in CLEF labs [1] in few ImageCLEF tasks [2] and in this way we continued our previous participation from 2013 when we participated in Plant Identification task [3]. Like in the 2014 campaign, the Scalable Concept Image Annotation challenge (from Image CLEF 2015 - Image Annotation[2]) task in 2015aims to develop systems that receive as input an image and produce as output a prediction of which concepts are present in that image, selected from a predefined list of concepts. In addition, this year the participants must describe images, localize the different concepts in the images and generate a description of the scene. This year the task was composed by two subtasks using a data source with 500,000 web page items. For each item we have a corresponding web page, an image and the keywords extracted from the web page. The participants must annotate and localize concepts and/or generate sentence descriptions for all 500,000 items. More details about challenge from 2015 are in [4] and details about challenge from 2014 are in [5].

[1]University "Alexandru Ioan Cuza" of Iasi, Romania
[2]ImageCLEF 2015 – Image Annotation: http://www.imageclef.org/2015/annotation

In 2014, three teams [6, 7 and 8] based their system on Convolutional Neural Networks (CNN) pre-trained using ImageNet [9].Also, most of the teams proposed approaches based on classifiers that need to be learned [6] or based on classification with constructed ontologies [10].

The rest of the paper is structured as follows: Section 2 details the general architecture of our system, Section 3 presents the results and an error analysis, while the last Section discusses the conclusions.

2 System components

In 2015, UAIC submitted runs both for concept identification (Subtask 1) and for generation of a description of the scene (Subtask 2). For that, we built a system, consisting in modules specialized for *text processing* and *visual processing*.

2.1 Subtask 1 - Textual processing

2.1.1 Google Translate

This module calls the Google Translation service[3] and it uses a file which contains the most frequent words in English and a cache file with already translated words. This cache file contains pairs of words in a foreign language and their translation in English.

Thus, for translating a current word from the initial file, we consider the following cases:

- *Case 1*: If the current word is in the file with most used words in English then the program uses this English form and then it skips to the new word from the initial file. If the new word isn't in this file then we search for it in the cache file (Case 2).
- *Case 2*: If the current word is in the cache with translated words we take the translated form and we skip to the new word from initial file. If not, we call the translation service (Case 3).
- *Case 3*: Before calling the translation service, we identify the language of the word and then we translate this word from the identified language in English.

Example:

Other language:... **portada** 1104 toronto 1080 **por** 1029 **vuelve** 990 **sorprender** 987 **cada** 974 **escenario** 970 pisa 961 **concierto** 933 …

English: … **home** 1104 toronto 1080 **by** 1029**back** 990 **surprise** 987 **each** 974 **scenario** 970 pisa 961 **concert** 933 …

[3] Google Translation service: https://cloud.google.com/translate/docs

The file with English words contains around 50.000 words and the file with cache contains 343.121 pairs like (*initial_word*, *translated_word*) (until last submission). From what we see, the files were accessed an average of 400 times per minute.

2.1.2 Stop-words Elimination

This component receives a file with 500,000 lines and tries to remove every stop-word from every line with its associated number which represents its frequency. We consider additional elimination of classical stop-words (*the*, *from*, *it*, *he*, *be*, *is*, *has*,…) and the elimination of all words with one or two characters. The number of classical stop-words was 667.

For example, for input line:

```
000bRjJGbnndqJxV 100 timepiece 7988 the 4823 time 3252 or 3100
of 2664 that 2169 thesaurus 1595 for 1578 device 1569 timepieces
1513 its 1397 measuring 1286 instrument 1285 clock 1268 wheel
1232 from….
```

after using this component we obtained:

```
000bRjJGbnndqJxV 58 timepiece 7988 time 3252 thesaurus 1595
device 1569 measuring 1286 instrument 1285 clock 1268 wheel 1232
…
```

In the end, from a total of 43.448.058 words in the input file, 16.802.111 stop words were eliminated. From 2 files summing up 457.1 Mb we obtained one file with 299 Mb.

2.1.3 Concept Identification

At this step, we start to build an ontology with relations between the initial 250 concepts and related words to them. For that we used the WordNet[4] [11] and we extracted in average around three synonyms and an average of five words that are somehow related to the concepts (automatically extracted from WordNet (hyponyms or hypernyms) and manually verified by human annotators or manually added by human annotators).

For example, for a concept we have the following information:

Concept: bicycle
Synonyms: bike cycle, two-wheeler, mountain bike, ten-speed, racing bike, recumbent fixie, penny-farthing, ordinary velocipede
Lexical family: park garden, tree, flower, wood, grass, path, kid, child, sport, sun, marathon, equipment, outfit, protection, street, saddle, handle bar

[4] WordNet: http://wordnetweb.princeton.edu/perl/webwn

With this file we execute on the processed file with 500.000 lines (after steps 2.1.1 and 2.1.2) a module which identifies related concepts for every line. For that, for each word, we try to find a way to connect it to concepts. That implies searching for it in the list of concepts (case 1) or in list of synonyms (case 2) or in list related to lexical family (case 3). If a match is found, the word is replaced with its related concept and placed in the output file along with its initial number (case 1 and case 2) or with a lower value (case 3).All words that could not be associated with any concept have been eliminated along with their number.

At the next step we sum the frequencies for the same concept and we put in the output only one value, with a unique appearance of this concept ID and with the individual sum value. Next, we normalize all the values on lines and we replace the individual sum value for every concept with a percentage value obtained after we calculate a global sum with all individual sum values.

For example, an input line looks like:

```
timepieces 81 timepiece 7988 time 3252 thesaurus 1595 device
1569 measuring 1286 instrument 1285 clock 1268 wheel 1232 noun
1170 balance 1162 legend 1105 dictionary 1095 …
```

This is how the same line of data looks like at output:

```
n03046257 0.527 n04555897 0.1802 n02866578 0.0769 n03249569
0.063 n04574999 0.0604 n06410904 0.0147 …
```

Additional, to this we built a file with relations between concepts and in case that one concept is present, we consider also the related concepts with a smaller percentage. For example, we consider a relation between “ear” and eye (because they are both body parts), and if we identify in an image the concept “eye” with score 0.0941, we consider also the concept “ear” but with a lower score equal with 0.0001. Although both the input file and the output file for this module are very close in size, they differ a lot due to the lack of concept-related words on some lines of the initial file.

Because, it would have taken us 400 minutes to parse all 500.000 line of data single-threaded but our program takes around 45 second per 50.000 lines because we decided to work with 10 execution threads. We need some additional time for the reconstruction of the final file with 500.000 data, but in the end we reduced the execution time from 400 minutes to about 15 minutes. All programs were run on an 8-core i7 Intel processor.

2.2 Subtask 1 - Visual Processing

2.2.1 Face recognition

After downloading all pictures from the URL file using a script, we ran the JJIL[5] for face recognition on all of them. The information we got from this part was merged with the information obtained by the textual processing component and converted in the output form.

Fig. 1. An example of image that contains faces

For example for image from Fig. 1, the output after we use the JJIL API is:

```
n05538625 0.5:165x180+220+273,0.5:155x167+439+227
```

It's worth mentioning that in around 10% of images faces were detected (in 48.000 images), though only 75% of images where put through the face recognition JJIL API (375.000 files) since only this many links were valid.

For example from the following line from the input file:

```
n03046257 0.7206    n02782093 0.1077    n02866578 0.1071
n04199027 0.0374    ….
```

This is how the corresponding output looks:

```
1 000bRjJGbnndqJxV n03046257 0.7:128x126+0+0 n02782093
0.1:128x126+0+0 n02866578 0.1:128x126+0+0 n04199027 …
```

[5] JJIL Face Recognition: http://www.richardnichols.net/2011/01/java-facial-recognition-haar-cascade-with-jjil-guide/

Downloading about 25.000 images took around 4 hours and after that, detecting faces in those images took another 2 hours. All programs were run on a single Thread on multiple computers to reduce the overall time.

2.2.2 Body parts identification

Based on results obtained at 2.2.1 and on image size (width and height), we tried to approximate the position of the face features using basic human proportions. If the resulting bounding boxes where inside the image then we used them as they were. However if one of them was exiting the image boundaries we would cut the outside part or, if necessary, removed them totally. We used this process for eyes, nose, lips, head, legs and feet.

2.2.3 Subtask 2 – Text Generation

For Subtask 2 we built a matrix with relations between concepts. In fact we built triplets in form ($concept_1$, *verb*, $concept_2$), where concepts are from concepts provided by the organizers and verb is a relation between concepts. With this resource we build sentences in which $concept_1$ is subject, verb is predicate and $concept_2$ is complement. For example, in our matrix are the following types of triplets:

- body_part – wearing – accessories;
- animal – drinking – drink;
- insect – in on – land vehicle;
- animal – near – man made object;
- animal – playing – sport_item_or_toy.

The rate of success in creating sentences for the given images was greater for the clean track in comparison with noisy track. The program used was single threaded and took about 4 hours to complete all 500.000 data on a dual core processor.

This is how lines of input data looks like:

```
1. 000bRjJGbnndqJxV n03046257 n04555897 n02866578 n03249569
n04574999 n06410904 n05600637 n02778669 n05564590 n03479952
n06277280 n02958343
2. 006fmXbGJW3UhmjI n03623556
3. 00DIvt1Zik2Vo1yY n07739125 n05254795 n04100174 n02849154
n03135532 n07848338 n05563770 n02801525 n09328904 n10287213
n03479952 n07747607 n04197391 n05311054 n05598147 n04379243
```

This is how the same lines of data looks like at output:

```
1. 2 000bRjJGbnndqJxV The clock is near a ball, the drum is in a
hallway and the wheel is near a clock.
2. 2 006fmXbGJW3UhmjI Empty sentence.
3. 2 00DIvt1Zik2Vo1yY The apple is on a table, the blanket is on
a table and the cross is near a table.
4. 2 00k_Jt7GwBTWPDIP The tower has a door, the pen is on a
table and the radio is on a table.
```

3 Results and Evaluation

For the 2015 task, our team submitted 4 runs for Subtask 1 and 4 runs for Subtask 2. The description and duration for every run is presented in bellow Table.

Table 1: Description of runs for Subtask 1

	Description	Duration
Run 1	Based on basic textual processing, like stop-words elimination, lemmatization, using of ontology	3 hours
Run 2	Additional to Run 1 we used the Google Translation service	5 days
Run 3	Additional to Run 2 we used visual processing for face recognition	12 hours
Run 4	Additional to Run 3, based on rectangle associated to face, we to add concepts related to body, arms, foots, etc.	4 hours

For Run 2, we used the Google Translation service[6] and because we were limited to a number of requests per second, we inserted some delays between successive calls of the service. Also, we created a cache with already translated words and before calling the translation service we checked to see if we have the current word in our cache. If we did, we skipped the current word and we used the translation from cache. Because in the input file were millions of words, this component for translation runs more than 5 days, and in the end we didn't succeeded to translate all words from the initial file. For the Run 3 and Run 4 we used only the created cache and for this reason the duration is lower than that of Run 2. For Run 3 we run the face recognition component on all images, on a distributed architecture with 10 different threads and in the end we concatenated the results. For Run 4 we used the partial results from runs 2 and 3 and for this reason it took less hours.

Table 2: Description of runs for Subtask 2 Noisy track

	Description	Duration
Run 5	Using file with triplets (concept1, verb, concept2), we build sentences based on first two concepts received as input	75 minutes
Run 6	Similar to Run 5, where we improve the number of triplets	2.5 hours
Run 7	Similar to Run 6, with an improved rule for selection of concepts	2.5 hours
Run 8	Similar to Run 7, with a new version for file with triplets and with new rules used in selection of most relevant concepts	5 hours

In the case of Subtask 2 noisy track, from run to run we completed our resource file with triplets of type ($concept_1$, *verb*, $concept_2$), and similar we add more new rules for selection of the most relevant concepts. Of course, the time duration for execution

[6] Google Translation service: https://cloud.google.com/translate/docs

increased from run to run. Similar with noisy track, we submitted five runs from R9 to R13 for Subtask 2 clean track.

3.1 Evaluation for Subtask 1

Table 3 below gives the results for the runs from Subtask 1 described above. More details are in [4].

Table 3: Results of UAIC's runs from Subtask 1

% Overlap with GT labels	R1	R2	R3	R4
50 %	0.020719	0.020009	0.021134	0.055917
0 %	0.185071	0.185288	0.18522	0.265927

As we can see from Table 3, the better run is the R4, where we use all created components (translation, stop-word elimination, concept identification, face recognition, body components identification, etc.). Also, in this run we used the final versions for our resources files with English words, with translated pairs, with concept synonyms, hyponyms and hypernyms, and related words. The most important component is component related to the identification of body parts (which is used only in R4). We can see how results for R1, R2 and R3 are much closed, but the results for R5 are more than twice as good. Because, between runs (R1 and R2), (R2 and R3) and (R3 and R4), we improve continuous all our resources (resources for translation, ontology, list with stop-words, etc.) it is hard for us to say what was the impact of every step performed by us.

3.2 Evaluation for Subtask 2

Tables 4and 5 from below give the results for the runs from Subtask 2noisy track and Subtask 2 clean track described above.

Table 4: Results of UAIC's runs from Subtask 2 noisy track

	R5	R6	R7	R8
MEAN	0.0409	0.0389	0.0483	0.0813
STDDEV	0.0310	0.0286	0.0389	0.0513
MEDIAN	0.0309	0.0309	0.0331	0.0769
MIN	0.0142	0.0142	0.0142	0.0142
MAX	0.2954	0.2423	0.2954	0.3234

Table 5: Results of UAIC's runs from Subtask 2 Clean track

	R9	R10	R11	R12	R13
MEAN	0.1709	0.2055	0.2080	0.2093	0.2097
STDDEV	0.0771	0.0589	0.0654	0.0661	0.0660
MEDIAN	0.1762	0.2078	0.2082	0.2082	0.2085

	R9	R10	R11	R12	R13
MIN	0.0258	0.0290	0.0290	0.0290	0.0290
MAX	0.7246	0.3850	0.7246	0.7246	0.7246

We can see how from run to run the results are improved. For noisy track R8 is much better than R5, R6 and R7 which have almost similar values. For clean track results are more closed, but we can see how these results are much better than results from noisy task. This mean that the selection of most relevant concept is the hardest part and by this step will depend the final result of this track. Similar to runs R5 to R8 were from run to run we improve our resource with triplets or our rules for building sentences, we obtained our runs from R9 to R13. For these runs we start from resources obtained for R8 and then from run to run we analyse our results and we changed our resources in order to obtain better results.

4 Conclusions

This paper presents the system developed by UAIC for the Scalable Concept Image Annotation Challenge from ImageCLEF 2015.This system contains components for Subtask 1 (Image Concept detection and localisation) and for Subtask 2 (Generation of Textual Descriptions of Images).

For Subtask 1, the main components of the system are related to text processing (the translation of non-English words, stop-words elimination, and concept identification) and to visual processing (face recognition and body parts identification). From the presented results we can conclude that the most important component is component related to body parts identification which increased significantly our results.

For Subtask 2, the main components are related to applying templates on selected concepts, based on a resource with triplets ($concept_1$, *verb*, $concept_2$). From what we see, the most important part is related to the selection of most important concepts, and from this reason the results for clean track are much better than results for noisy track.

For the future, we aim to use more visual processing in order to identify more concepts from images. Also, on textual processing we want to reduce the time duration for translation, which was the most time consuming component.

Acknowledgement. The research presented in this paper was funded by the project MUCKE (Multimedia and User Credibility Knowledge Extraction), number 2 CHIST-ERA/01.10.2012. Special thanks go to all colleagues from the Faculty of Computer Science, second year, group A1, who were involved in this project.

References

1. Cappellato, L., Ferro, N., Jones, G., and San Juan, E. (editors). CLEF 2015 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings (CEUR-WS.org), Vol. 1391. (2015)

2. Villegas, M., Muller, H., Gilbert, A., Piras, L., Wang, J., Mikolajczyk, K., Seco de Herrera, A. G., Bromuri, S., Amin, M. A., Mohammed, M. K., Acar, B., Uskudarli, S., Marvasti, N., B., Aldana, J. F. and Garcia, M. R. General Overview of ImageCLEF at the CLEF 2015 Labs. Springer International Publishing, Lecture Notes in Computer Science. (2015)
3. Șerban, C., Sirițeanu, A., Gheorghiu, C., Iftene, A., Alboaie, L., Breabăn, M. Combining image retrieval, metadata processing and naive Bayes classification at Plant Identification 2013. Notebook Paper for the CLEF 2013 LABs Workshop - ImageCLEF - Plant Identification, 23-26 September, Valencia, Spain. (2013)
4. Gilbert, A., Piras, L., Wang, J., Yan, F., Dellandrea, E., Gaizauskas, R., Villegas, M., Mikolajczyk, K.: Overview of the ImageCLEF 2015 Scalable Image Annotation, Localization and Sentence Generation task. In CLEF2015 Working Notes – CEUR Workshop Proceedings, Publisher CEUR-WS.org, ISSN: 1613-0073.Toulouse, France, September 8-11. (2015)
5. Villegas, M., Paredes, R.: Overview of the ImageCLEF 2014 Scalable Concept Image Annotation Task. In: CLEF 2014 Evaluation Labs and Workshop, Online Working Notes. (2014)
6. Kanehira, A., Hidaka, M., Mukuta, Y., Tsuchiya, Y., Mano, T., Harada, T.: MIL at ImageCLEF 2014: Scalable System for Image Annotation. In CLEF 2014 Evaluation Labs and Workshop, Online Working Notes. Sheffield, UK, September 15-18. (2014)
7. Vanegas, J.A., Arevalo, J., Otálora, S., Páez, F., Pérez-Rubiano, S.A., González, F. A.: MindLab at ImageCLEF 2014: Scalable Concept Image Annotation. In CLEF 2014 Evaluation Labs and Workshop, Online Working Notes. Sheffield, UK, September 15-18. (2014)
8. Xu, X., Shimada, A., ichiro Taniguchi, R.: MLIA at ImageCLEF 2014 Scalable Concept Image Annotation Challenge. In: CLEF 2014 Evaluation Labs and Workshop, Online Working Notes. Sheffield, UK, September 15-18. (2014)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 248–255. June. (2009), doi:10.1109/CVPR.2009.5206848
10. Reshma, I.A., Ullah, M.Z., Aono, M.: KDEVIR at ImageCLEF 2014 Scalable Concept Image Annotation Task: Ontology based Automatic Image Annotation. In CLEF 2014 Evaluation Labs and Workshop, Online Working Notes. Sheffield, UK, September 15-18. (2014)
11. Fellbaum, C.: WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press. (1998)