AUTOMATIC IMAGE ANNOTATION

Conference Paper · September 2014

DOI: 10.13140/2.1.2633.9842

Adrian Iftene
Universitatea Alexandru Ioan Cuza

AUTOMATIC IMAGE ANNOTATION

ANDREEA-ALICE LAIC, ADRIAN IFTENE

"Alexandru Ioan Cuza" University of Iaşi, Faculty of Computer Science
{andreea.laic, adiftene}@info.uaic.ro

Abstract

In the recent years, multimedia content has grown increasingly over the Internet, especially in social networks, where users often post images using their mobile devices. In these networks such as Flickr, the content is later used in search operations when some users want to find something using a specific query. Nowadays, searching into these networks is primarily made using the title and the keywords associated to resources added by users that have posted the content. The problem we face comes from the fact that in many cases, the title or the related keywords are not relevant to the resource and only after we analyse the image, can we conclude what it contains in reality. The project that we want to present in this article proposes that each image is connected to relevant keywords according to its content. In order to do this, the first step was to create a collection of images that was annotated by human annotators, while the second step was to expand this collection of images performing search on the Internet using keywords associated to the initial collection of annotated images. Currently, for a new picture, we can identify similar images in our collection of images and based on the keywords associated with them, we can determine what keywords characterize this new image. The evaluation of this system has demonstrated that our approach works efficiently for images for which we can find similar images in our collection.

Key words — Image retrieval, automatic image annotation, Flickr.

1. Introduction

The domain of image retrieval is dedicated to systems which deal with browsing, indexing and searching for images in a large context (Datta et al., 2008). Typically, this search is done by keywords, metadata and descriptions of images. The volume of data has significantly increased during the past years, which has led to the development of algorithms performing image processing, the *Image Retrieval* domain being in continuous expanding. Big companies like Google[1], Bing[2], Yahoo[3] have developed tools in time and they have optimized algorithms to be efficient while searching for images, a proof of this is the option "*Image Search*" that they offer.

Content-Based Image Retrieval is preferable because usual keyword search depends on the quality and accuracy of annotations (Eakins et al., 1999). Until now, Google has had the most complex system for automatic recognition of image elements.

[1] https://support.google.com/websearch/answer/1325808?hl=en
[2] https://www.bing.com/?scope=images&nr=1&FORM=NOFORM
[3] http://images.search.yahoo.com/

Google proposes a new type of image search, the one through similar images[4] (images that have similar content, both in color and texture, and the components of the image) to user data. This option is available only in the browser, allowing the user to drag-and-drop an image, enter the URL of the image or make a simple image upload. The advantage of this option (compared to what is now on the market) comes from the fact that the image database from Google is impressive (~100,000,000 gigabytes[5] of indexed pages). The disadvantage of this option regarding programmers is that Google still does not provide an API for application developers.

Similar to what Google offers, TinEye[6] developed a framework that allows you to perform a *reverse search* by image. There is a Web application where the user can enter an URL, drag-and-drop or upload an image and get similar results with the image inserted by him. Different from Google, TinEye offers an API for application developers, but the process of integration into an application development is chargeable.

RevIMG[7] is a search engine of images through other images. It provides a library for JavaScript and one for Android mobile applications. This engine is intended only to certain image categories like pictures, monuments, famous people, flags, etc.

Besides these applications, there are a series of platforms (*Lire*[8], *pHash*[9]) which are able to extract the content items (color, texture, etc.) of the image. The application that we developed uses *Lire*, a library corresponding to the application requirements in terms of type of search (which is done by image content) and speed of rendering the results (which is small).

2. *System architecture*

The first development step was to build a collection of images that were manually annotated with keywords, collection which was expanded, using the Internet, by searching the keywords associated to the initial collection of annotated images. Next, for a new picture offered by the user, we can identify similar images in our collection and based on the lists of keywords associated with them, we can determine what list of keywords characterizes this new image.

2.1. *Creation of gold collection with annotated images*

The initial collection of images consisted of 100 images, from different areas. The images were categorized in the following proportions: 30% images with peoples, 15% images from nature, 20% images with animals and the remaining images were from various categories (art, furniture, sport, other, etc.).

The images were selected by six human experts and then were manually annotated by human annotators. Some of the images have words in their visual content to see how

[4] https://support.google.com/websearch/answer/1325808?hl=en

[5] http://www.google.com/insidesearch/howsearchworks/crawling-indexing.html

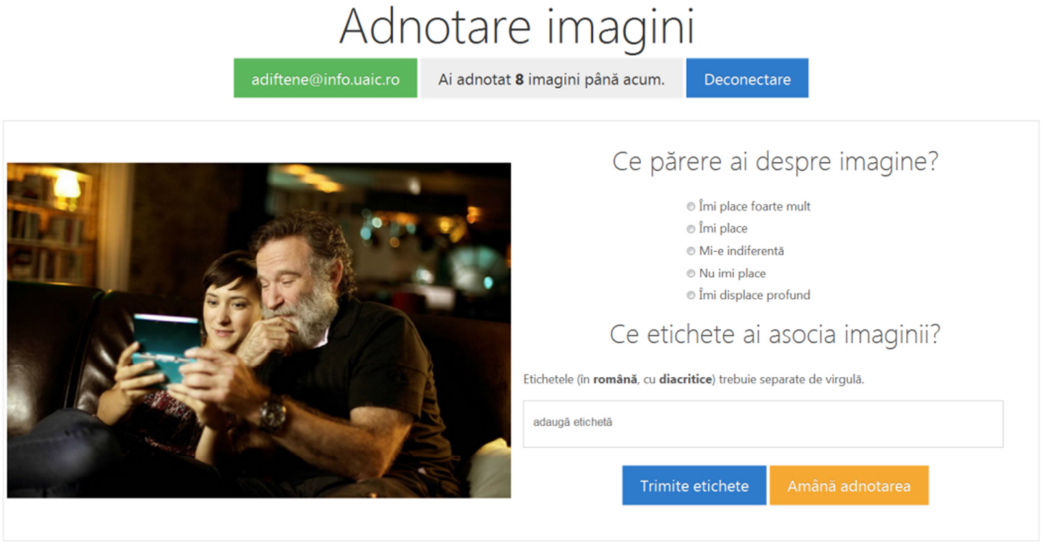
[6] https://www.tineye.com/

[7] http://www.revimg.net/

[8] http://www.semanticmetadata.net/lire/

[9] http://www.phash.org/

this can influence the process of annotation. In Figure 1, you can see how a logged user can annotate an image.



Adnotare imagini

Ai adnotat **8** imagini până acum.

Ce părere ai despre imagine?

○ Îmi place foarte mult
○ Îmi place
○ Mi-e indiferentă
○ Nu imi place
○ Îmi displace profund

Ce etichete ai asocia imaginii?

Etichetele (în **română**, cu **diacritice**) trebuie separate de virgulă.

adaugă etichetă

Figure1: Application interface where users can annotate images

In the section "*Ce părere ai despre imagine*?" (English: *What is your opinion about this image*?), the user can select how much she/he liked the image shown. We record these opinions in our database and this action allows us to build profiles for users who have annotated images and also to build a recommendation system for them.

In the section "*Ce etichete ai asocia imaginii*?" (English: *What keywords would you associate to the image*?), the user can indicate a series of Romanian keywords that she/he considers suitable for the image. Besides the simple words, they can also write expressions which they consider appropriate for the image.

2.1.1. Experiments

In the process of annotating, there were 28 volunteers in third-year and master students of the Faculty of Computer Science from Iaşi. They had to annotate 100 images; the only criterion was to write keywords in the Romanian language, criterion that was established from the beginning.

Comparing the keywords entered by users for the same picture, it was seen that there were small differences among the words entered, most of them were from the same lexical family or they were synonyms. Each user was able to annotate how many pictures she/he wanted, but in the analysis entered only keywords by 21 users who have annotated all 100 images.

Performing an analysis on what users annotated over a period of two weeks, it can be said that their tendency was to introduce, on average, 3.41 keywords per image, with a minimum of 2 keywords for an image and a maximum of 12 keywords for an image. Looking further into the keywords that they have entered, it can be said that most users have opted for simple words and not phrases. As a general rule, they have chosen to annotate the content of the image that quickly appears in sight. In the end, the 21 users have entered a total of 1,514 keywords for 100 images.

For example, for Figure 2, the users have chosen keywords such as *câine, cățel, copil, pat, cerceaf, puritate* (in English: “dog”, “puppy”, “baby”, “bed”, “bed sheet”, “purity”), elements that can be easily seen in the image, and not keywords like *lemn* (in English: “wood”), which can hardly be seen in the background.

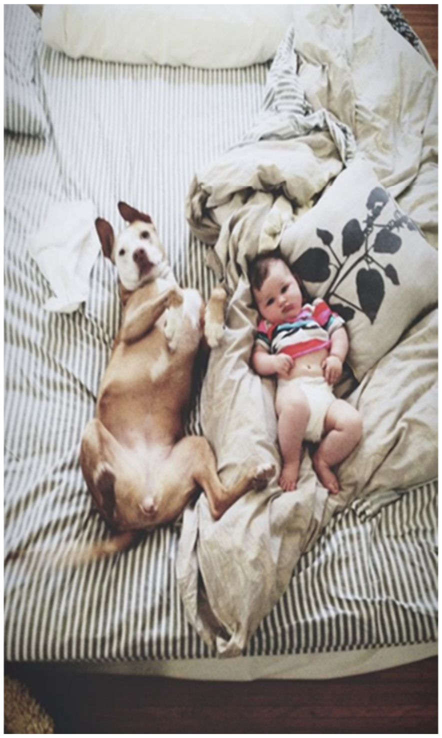


Figure 2: One of the images annotated by the users

Furthermore, we have implemented an algorithm which, for each image, counts the frequency of lemmas of the keywords associated by users and keeps those with a frequency of at least 4. Besides frequency, we considered the relation of synonymy using the Romanian WordNet (Tufiș et al., 2004). Among all synonyms, we kept the keyword which appears more often at the users who have annotated the image.

For expressions, we used the division into component words, and then we calculated the frequency of word components based on lemma and synonymy. If all components of the expression had an occurrence frequency over 4, we decided to keep the expression and give up the words which appeared in the expression. In the end, we considered for every image a list of keywords in a descending order of frequency (of course, for frequencies over 4).

In addition to the score calculated for each keyword based on frequency, we decided to calculate a score for each user who annotated all images. Furthermore, regarding the way we calculated the score for keywords, for the user’s score we took into account the order of the entered keywords. For example, the user’s score was calculated as a product between the number of users who entered that keyword and its quota, given from the formula (1).Thus, we could identify the reliable and the less reliable annotators.

$$(1)\quad \frac{1}{abs(index\_of\_keyword\_in\_final\_list - index\_of\_keyword\_in\_user\_list)+1}$$

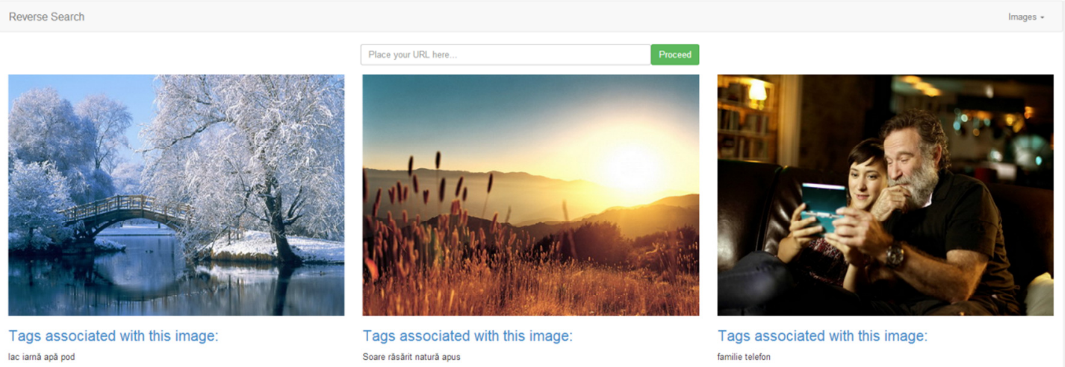
Each image initially contained around 30-40 different keywords from all users, and afterwards, we applied the algorithm, the number of keywords was reduced to approximately 3-4 keywords per image. The average remained of 3.32 keywords per image, with a minimum of 1 and a maximum of 7. It can be seen that filtering was done quite rigorously.

After we performed the steps explained above for the image from Figure 2, we were left with the following keywords: *cățel*, *copil, pat* (in English: “dog”, “child”, “bed”).

After completing this step, we increased the initial collection with 100 images as it follows. For each image from the initial collection, we added 10 new images to our collection, thus increasing the image collection to 1,000 images. For this, we searched for Google images using lists of keywords associated with each image. For the first 10 results, we initially associated the list of keywords used in the search process, followed by a process of verification, corrections, additions to this list; this process was done with human annotators.

2.2. Reverse Image Search

This module uses the 1,000 collection of images with related keyword lists obtained at the previous step. Regarding this collection, we know that the list contains relevant keywords associated with images.



Tags associated with this image:
lac iarnă apă pod

Tags associated with this image:
Soare răsărit natură apus

Tags associated with this image:
familie telefon

Figure 3: Reverse Image Search application

The main purpose of this module is to generate a list of keywords that characterize an image given by the user. This is done as follows:

- The user introduces an URL and presses the **Proceed** button (See Figure 3).
- Behind the applications, we use the LIRE[10] library (Lucene Image REtrieval) (Lux and Marques, 2013), which compares the new image with the images from our collection. It establishes a set of 20 images most closely to the new inserted image in terms of texture and color. LIRE uses many low-level characteristics in the

[10] LIRE: http://www.semanticmetadata.net/lire/

indexing processing such as Color Layout[11], Edge Histogram (Park et al., 2000), CEDD[12] (Color and Edge Directivity Descriptor), FCTH (Fuzzy Color and Texture Histogram) (Chatzichristofis and Boutalis, 2008), etc. and then it uses the Euclidian distance for finding similar images. We used, in the first instance, the FCTH characteristic, but we also made experiments, with the other values.

- To establish the list of keywords that we associate with an image and their order in this list, we apply the algorithm from section 2.1.1. For that, the input that we use is represented by lists of keywords from 20 similar images, and then we use lemmatisation, the synonymy relation and the processing of expressions.
- An exception to the above is the case when all Euclidean distances between the new image and all images from the collection are below 0.2. This value was found experimentally and it tells us that the new image is too different in comparison with the existing images from our collection. In this case, we cannot associate keywords to the new image.

2.2.1. Use-cases

To illustrate the two cases described above, we carried out two searches to see how the application behaves.

2.2.1.1. There are similar images in the collection

This is the case when the application works as we wanted and it is able to build a list of relevant keywords to be associated to a new image.

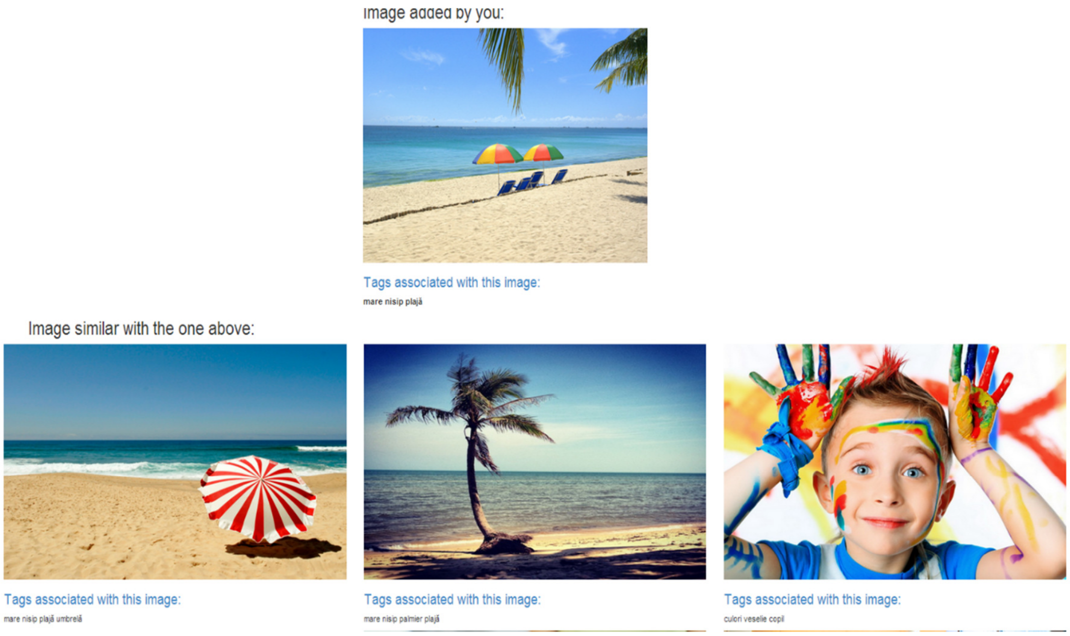


Image added by you:

Tags associated with this image:
mare nisip plajă

Image similar with the one above:

Tags associated with this image:
mare nisip plajă umbrelă

Tags associated with this image:
mare nisip palmier plajă

Tags associated with this image:
culori veselie copil

Figure 4: Reverse Image Search when there are similar images in the collection

[11] ColorLayout: http://en.wikipedia.org/wiki/Color_layout_descriptor

[12] CEDD: http://www.itec.uni-klu.ac.at/lire/nightly/api/net/semanticmetadata/lire/imageanalysis/CEDD.html

In Figure 4, it can be seen that there are similar images in the database with the one introduced by the user, and the list of keywords contains relevant keywords for this image.

2.2.1.2.There are no similar images in the collection

In this case, because of the limit imposed by the Euclidean distance, the user will receive a negative response. This means that the algorithm didn't find similar images with the user image in the collection of images, and thus it is unable to create a list of keywords that characterize it.

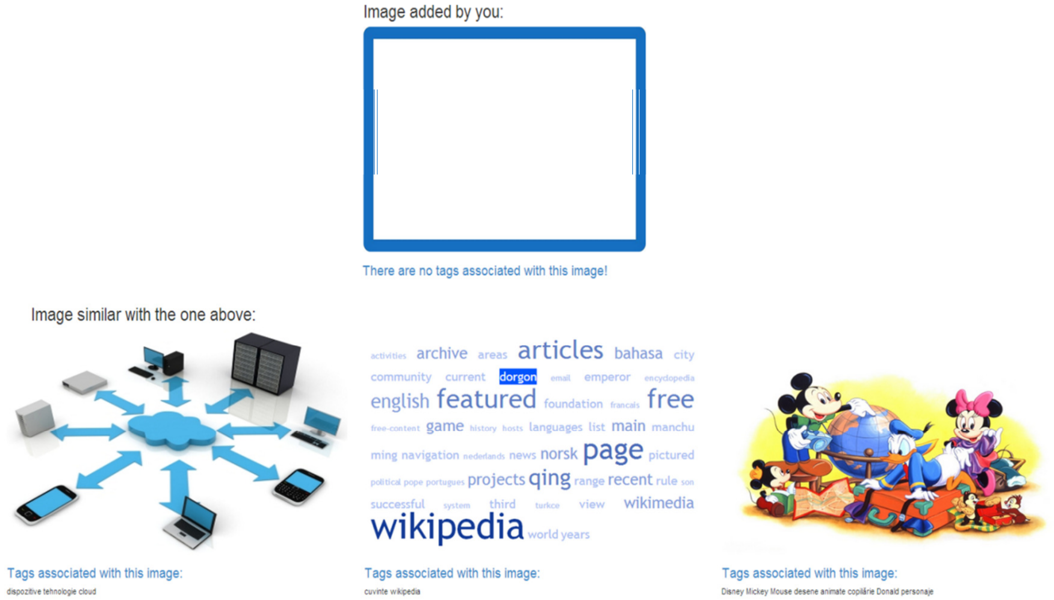


Image added by you:

There are no tags associated with this image!

Image similar with the one above:

activities archive areas articles bahasa city
community current dorgon email emperor encydopedia
english featured foundation francais free
free-content game history hosts languages list main manchu
ming navigation nederlands news norsk page pictured
political pope portugues projects qing range recent rule son
successful system third turkce view wikimedia
wikipedia world years

Tags associated with this image:
dispozitive tehnologie cloud

Tags associated with this image:
cuvinte wikipedia

Tags associated with this image:
Disney Mickey Mouse desene animate copilărie Donald personaje

Figure 5: Reverse Image Search when there are not similar images in the collection

In Figure 5, it can be seen that in the collection of images there are no similar images with the one introduced by the user (the images shown in the second row have the Euclidian distance below the limit imposed by us). We also note that these images do not contain common keywords, which may characterize the new image inserted by the user.

The conclusion that can be drawn from the analysis of the two cases: the more images we have in our collection of annotated images, the more chances of finding similar images. This conclusion is also strengthened by the experiments that we perform in the next section.

2.2.2. Evaluation

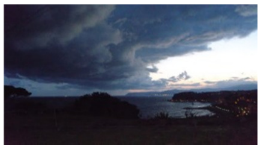
To see how accurate the above system is, we conducted a series of experiments on two different sized collections of images with related keyword lists. The first collection has 100 images and the second collection has 200 images (100 from the first collection plus 100 similar to those). In the following pages, we present the experiments that we have done to evaluate the system.

We considered 20 new images taken from the Internet and then we used the application separately on two collections of images. For each of the 20 new images taken from the Internet, we have made processing using the system created and we monitored the following values:

- How many keywords are added, on average, to an image;
- How long the processing of an image takes;
- How many keywords added to an image are incorrect.

In Table 1, one can see the obtained results: the number of keywords added to a picture is, on average, 1.89 keywords for a collection with 100 images and 2.74 for a collection with 200 images. Of course, these values depend on the number of keywords associated to the images from our collections (where the average number was around 3.32).

Table 1: System evaluation

Image from database	**How many keywords are added, on average, to an image**		**The average length of the processing (seconds) per imagine**		**How many keywords added to an image are incorrect**	
Number of images from database	**100**	**200**	**100**	**200**	**100**	**200**
	1	1	52	101	0	0
	3	3	50	115	0	0
	2	4	27	240	0	1
	2	3	26	302	0	0
The average of those 20 images	*1.89*	*2.74*	*40*	*212*	*0.31*	*0.45*

The average duration for application was around 40 seconds for the collection of 100 images, and around 212 seconds for the 200 image collection. This duration varies due to the feature histogram FCTH.

The number of incorrect keywords for a picture is quite small. Wrong keywords appear when the terms of texture and color of an image are very similar to another image from the image collection, showing different elements in the picture.

As a conclusion, after we analysed the results from Table 1, we can say that the system created is a stable one and it offers good results to the user.

3. *Conclusions*

The application presented in this article can be very useful when you have a new image and you want to know what elements it contains. In order to do this, firstly, we need a large collection of annotated images with relevant lists of keywords. Secondly, we need performance algorithms which provide the distance between images in order to find images which are similar to the new one. Thirdly, we assign a list of keywords to the new image by making the intersection of keywords from similar images.

After evaluating the created system, we can say that the system works effectively as long as the requested image finds similar images in our collection. Consequently, it is very important that this collection be very large.

One problem that arises comes from the fact that the application responds slowly when the collection of used images is large (as we can see in Table 1).

Therefore, future directions for improving this application are the following: (1) the first direction is related to optimal and accurate algorithms that can identify specific elements in the new image (such as buildings, trees, people, sky, sea, etc.). These algorithms do not depend on the size of the collection of images and the offered results can be very fast and very accurate. (2) A second direction concerns the increase of the collection of annotated images, but it needs to be combined with the use of cloud platforms in order to have a low response time.

Acknowledgments

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.

References

Chatzichristofis, S., Boutalis, Y. S. (2008). FCTH: fuzzy color and texture histogram, a low level feature for accurate image retrieval. In *Proceedings of WIAMIS'08 Proceedings of the 2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services*, IEEE Computer Society Washington, 191-196.

Datta, R., Joshi, D., Li, J., Wang, J. Z. (2008). Image Retrieval: Ideas, Influences, and Trends of the New Age. *ACM Computing Surveys (CSUR)*, 40:2(5), 1-60.

Eakins, J., Graham, M. (1999). Content-based Image Retrieval. *Library and Information Briefings*, 85, 1-15.

Lux, M., Marques, O. (2013). Visual Information Retrieval using Java and LIRE. *Synthesis Lectures on Information Concepts, Retrieval, and S*, Morgan & Claypool Publishers.

Park, D. K., Jeon, Y. S., Won, C. S. (2000). Efficient use of local edge histogram descriptor.In Proceedings of *the 2000 ACM workshops on Multimedia (MULTIMEDIA '00)*. ACM, New York, NY, USA, 51-54.

Tufiş, D., Barbu, E., BarbuMititelu, V., Ion, R., Bozianu, L. (2004). The Romanian Wordnet. *Romanian Journal of Information Science and Technology*, 7:1-2, 107-124.