TVPulse: Improvements on detecting TV highlights in Social Networks using metadata and semantic similarity

Afonso Vilaça
Instituto de Telecomunicações
Aveiro, Portugal 3810-193
Email: afonso.vilaca@av.it.pt

Mário Antunes
Instituto de Telecomunicações
Aveiro, Portugal 3810-193
Email: mario.antunes@av.it.pt

Diogo Gomes
Instituto de Telecomunicações
Universidade de Aveiro
Aveiro, Portugal 3810-193
Email: dgomes@ua.pt

***Abstract*—Sharing live experiences in social networks is a growing trend. That includes posting comments and sentiments about TV programs. Automatic detection of messages with contents related to TV opens new opportunities for the industry of entertainment information.**

This paper describes a system that detects TV highlights in one of the most important social networks - Twitter. Combining Twitter's messages and information from an Electronic Programming Guide (EPG) enriched with external metadata we built a model that matches tweets with TV programs with an accuracy over 80%. Our model required the construction of semantic profiles for the Portuguese language. These semantic profiles are used to identify the most representative tweets as highlights of a TV program. Measuring semantic similarity with those tweets it is possible to gather other messages within the same context. This strategy improves the recall of the detection. In addition we developed a method to automatically gather other related web resources, namely Youtube videos.

I. Introduction

The number of social networks users worldwide is increasing every year, with 2.078 billion active accounts in 2015[1]. Twitter is a microblogging network, characterized by its 140-character messages, called tweets. It has more than 302 million active users (May 2015)[2]. The simplicity by which users can share their experiences live, including TV watching, makes it ideal for our use case.

Tweets content recognition and detection of messages related to TV programs would provide a sense of the impact of those TV shows on the network. This is of particular interest for the TV industry[3]. The frequency of tweets related to a given TV show can tell one not only the most popular programs, but also the highlights of those programs. Web and interactive TV applications can provide to the users an experience that joins television with social networks. The automation of the detection makes this service applicable on many scenarios, with real time responses. A particular use of content recognition from tweets is the summarization of TV programs into videos with the highlighted moments and the tweets with more impact. Relevant tweets offers a textual report of the show from the users reactions. Other option is to offer web resources related to the detected contents.

A simple way to get TV programs information is from the Electronic Program Guide (EPG). That information can be enriched with information from external sources.

In this paper we describe a system built to extract and store information from Twitter and from publicly available EPG's, build a semantic profile for the extracted contents and match tweets with TV programs. It is important to state that our work focus is on the Portuguese language, which ultimately distances ourselves from the state of the art. In that way, this becomes a novel work, including the construction of semantic profiles for Portuguese language.

II. State of the Art

The first step in text mining projects is text processing[1]. Several techniques are reported in literature. One of the first procedures is usually to lower-case all words. In microblogging messages, grammar rules for capitalization and accentuation are not followed by users, and this technique is justifiable. For the same reason, accents are also usually removed. Terms removal is also applied in many applications, and what is removed depends on the goal. The most common words, also called stop words, act as noise in sentences and their removal is advisable. Those words may be identifiable from the collected dataset or can be retrieved from a stored list built for the specific language. Other terms may or not have interest to the application being developed. Examples of those terms are swear words, short words, hyper links, punctuation, numbers or *emoticons*. Typically, terms are also stemmed in order to get a homogenization of their stemmed versions. Many stemming algorithms exists, some more general, other optimized to specific languages. In Portuguese language a known stemming algorithm is the RSLP stemmer[2]. All text filtering options should be optimized for the specific case study.

[1]http://wearesocial.net/blog/2015/01/digital-social-mobile-worldwide-2015

[2]http://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

[3]https://www.ibm.com/developerworks/community/blogs/025bf606-020a-48e9-89bf-99adda13e9b1/entry/by_the_numbers_social_media_impacts_the_entertainment_industry

A second choice to be made in text mining, after the filtering process, are the features to be used[1], [3]. An obvious choice are the terms that passed through the filtering process. Other features can be sequence of consecutive terms, like bigrams (two) or trigrams (three). Social networks have their own syntax, where some characters in the beginning of the word give a particular meaning to it. The most popular in Twitter are *hashtags* (started by '#') and user names (started by '@'). Using these special terms as features is usually valuable for many machine learning problems.

Words and features can be grouped according to their grammatical properties. Part-of-speech (POS) tagging on chats and microblog messages is a big challenge of Natural Language Processing (NLP), considering the large use of typing errors, abbreviations, dialect variations and use of slang words and peculiar vocabulary. There is however good progresses mainly for English. We highlight the work of Owoputi et al.[4]. Their model of POS tagging is a first-order maximum entropy Markov model (MEMM) that achieves an accuracy of 93%.

Both information retrieval and clustering techniques are necessary to measure distances between words, sentences or documents. These measures can be just the count of equal words, an edit distance, that represents the cost that takes to transform a word into another. Or even semantic distance, based on statistics of co-occurrences of words in the same document, sentence or window (sequence of words with a fixed length). A popular edit distance is the Levenshtein distance[5]. There are other posterior measures derived from this one. They are useful in string matching where words with small edit variations between them are assumed to belong to the same family or the difference be caused by a typing error. To measure similarity of words in terms of their meaning it is necessary to associated them to concepts. Mohammad and Hirst[6] propose a semantic distance using distributional profiles (DP) of concepts. They not only build to each word a DP, but they also group words to represent concepts and create profiles from those concepts using bootstrapping. Concepts are inferred from the context and decisions are made based on concepts distances. They showed that distributional concept-distance measures outperformed word-distance measures on ranking word pairs.

Other approaches can be followed to tag tweets. Classification algorithms can be used when there is a tagged dataset. Cremonesi et al.[7] collected tweets from specific TV programs and movies pages and trained 1-class-SVM models. They achieved results on associating tweets to programs and movies with precision of 92% and recall of 65%.

Another work on detection of events on TV programs from Twitter messages, in Japanese, is Nakazawa et al. [8]. They associate tweets to TV programs based on hashtags and inspect their content to identify hot moments. They extract people names and keywords from the terms co-occurring with those names. Their success on tagging events is above 66.8%.

Semantic characterization of tweets is commonly used to classify or cluster them: Genc et al. [9], Abel et al. [10] and Ozdikis et al.[11]. The first work associate tweets to Wikipedia pages, the second extract contexts from messages to construct user profiles, the last one also focus on event detection. All of them show that semantic similarity is a better procedure to relate text messages in terms of their context, comparing with the standard methods.

In a preliminary work we present a first version of our system with a successful highlights detection[12]. We had however some limitations, mainly in the used of manually inserted metadata. Even so, the achieved precision was higher than 80%. We also inspected with a positive perspective the potential of the use of semantic similarity on relating tweets.

III. System's architecture

Fig. 1 is a diagram of the system's architecture. Tweets are collected from Twitter Search API[4] using a JAVA agent. Since we are interested in the Portuguese community, we only collect tweets created in Portugal, meaning that they're almost all written in Portuguese. A tweet is represented not only by its text, but also by other information, like its id, user information, time of its creation, place, number of re-tweets, *hashtags*, if it is a reply to someone, etc. TV programs are also collected programmaticaly from Sapo Services [5]. A TV program is represented by its title, channel name and acronym, start and end times, a description and a short description. Tweets are stored in a Cassandra database (for scalability) and TV programs in a MySQL database (for easier information retrieval). Additionally we add to TV programs related metadata to the database, extracted from Wikipedia pages. To do it we use some python modules: *wikipedia*[6], *requests*[7] and $beautifulsoup4$[8]. Searches are done on the Portuguese language, using programs titles and the word *Portugal*, since homonym programs exist in other countries. The information is extracted from pages tables and from specific sections with interesting indexed contents, like *Elenco* (cast) or *Apresentadores* (presenters). This procedure automates and eases the quest for extra information. The only human dependent part is the association of hashtags not directly derived from the program title. For the most popular TV programs we manually insert a list of related hashtags.

Each tweet or program information is processed using a text processing pipeline implemented in Python with the following transformations. The text is converted from upper cases to lower cases, removal of accents, punctuation, numbers, stop words and swear words, finally a stemming transformation based on the RSLP algorithm [2]. Stemming allows the homogenization of words from the same family. Each processed document (tweet or program) is tokenized into a bag of words (BOW).

From the BOW we extract their features. Unigrams, bigrams and trigrams are computed for tweets text and programs title,

[4]https://dev.twitter.com/overview/api
[5]https://store.services.sapo.pt/en/cat/catalog/other/meo-epg
[6]https://pypi.python.org/pypi/wikipedia/
[7]http://www.python-requests.org
[8]https://pypi.python.org/pypi/beautifulsoup4/4.4.0

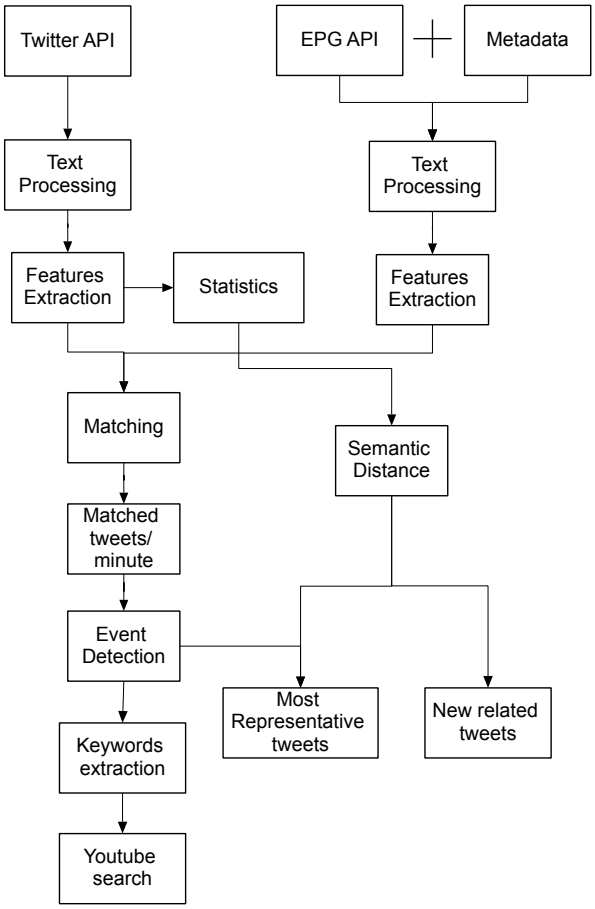

Fig. 1. *System diagram.*

description, short description, channel name and acronym and metadata. Tweets hashtags are also considered as features.

Portuguese distributional profiles are being extracted continually from each tweet. Terms frequencies and number of co-occurrences are stored in a MySQL database. From these statistics it is possible to build semantic features based on terms distributional profiles. To measure the distance of two words we use the cosine distance[6]:

$$Cos(w_1, w_2) = \frac{\sum_{w \in C(w_1) \cup C(w_2)} (P(w|w_1) \times P(w|w_2))}{\sqrt{\sum_{w \in C(w_1)} P(w|w_1)^2} \times \sqrt{\sum_{w \in C(w_2)} P(w|w_2)^2}} \quad (1)$$

where $C(w_i)$ is the set of words that co-occur with word w_i. The conditional probabilities are the relative frequencies on the word profile. Considering the high-dimensionality of the co-occurrences matrix, its reduction speeds up the process and may improve the accuracy, too. Two reduction methods can be used: the power law[13], also known as 80-20 rule, and the elbow method [14]. We apply both to study their influence.

To check if a tweet is related to a program or not, we developed a matching procedure. Co-occurrences of previously computed features in both sides are counted. A special treatment is given to *hashtags*. It is checked if the text of a tweet *hashtag* is in the program title (with spaces removed) and vice versa. This is worthy in the cases where there is no auxiliary *hashtags* table for the program. Some cases have required very specific functions such as the one that detects if a program is a football match. If it is the case, it looks in the tweet for a regular expression that represents a goal (*golo* in Portuguese): $\hat{}g+o+l+o+\$$. We empirically built a tree that gives a score to the tweet-program pair based on the matching results of each feature. That score discretely varies from 0 to 1, where 0 means that the tweet and the program are not related and 1 means that they are very related. 0.5 is used as minimum acceptable value to attribute a tweet to a program.

One of the goals of this work is to detect relevant events on TV programs with impact in Twitter. In order to achieved this we analyze matching results minute by minute during the program duration. Besides counting the number of matched tweets per minute, we compute a second derivative of that frequency ($f(m_{i+1}) - 2f(m_i) + f(m_{i-1})$). The following two measures allow us to decide if on that minute occurs a relevant event or not:

$$Event(m_i) = \begin{cases} True, & f''(m_i) \leq \overline{f''(m)} - \sigma_{f''(m)} \\ & \wedge f(m_i) > 5 \\ False, & otherwise \end{cases} \quad (2)$$

where $f(m_i)$ is the number of matched tweets for the minute m_i and $f''(m_i)$ the value of its second derivative. $\overline{f''(m)}$ is the average for all the minutes of the program and $\sigma_{f''(m)}$ the standard deviation.

To understand the topic of the event two approaches are followed. First, consists only in showing the most used words. Second, we try to identify the most representative tweets. This is achieved through semantic similarity. A graph is built, where nodes are tweets and edges are the semantic similarities between them (mean of words similarities). The most representative tweet is that which the sum of its edges is higher, i.e. the one closest to the graph centroid. Again in special cases, such as if the program is a football match, the number of tweets with reference to a goal is also computed to support the goal detection on that event.

An extra feature is added, that meets the goal of finding web resources related to a TV program and its highlights. The most popular unigrams and bigrams are used as keywords on Youtube searches. We do it using the Youtube Data API[9]. Each extracted keyword or keyword pair is joined to the program title and channel name for a search with a proper filter: the time a video was published must be after the program start time; its duration must be less than 20 minutes. For validation, we do a matching between titles from the retrieved videos and the keywords. At the end the remaining videos are rated according to a linear function of their number of views, likes, dislikes, comments and favorites, where each of the variables have a different weight. The proposed videos are the better rated: maximum of 3 for an event and 10 for the entire program. From those videos we also extract their comments. They can be used to enrich the TV program vocabulary and be a resource for future research work.

[9]https://developers.google.com/youtube/v3/?hl=en

Our work is made available to others through a public API. The API implemented using CherryPy [10] responds to queries with matched tweet-program pairs for a given time interval, informing when an event occurs and providing the related Youtube videos. The channel acronym or the program title can be specified. It is also possible to choose only tweets with a specific *hashtag* and to control the system accuracy by specifying the minimum matching score acceptable.

As we have shown in [12], generally tweets related to a same TV program have higher similarity than tweets related to two distinct ones. Sustained on this fact, we applied a second strategy to improve the matching recall, capturing tweets not necessarily with words present on programs EPG and metadata, but still belonging to the same vocabulary. This procedure is divided in two parts. First we collect tweets related to the TV program using the method described above. We use a minimum score of 0.6 to ensure high precision on the matched tweets. All the drawn tweets are used to build a proper semantic for the TV program, based on words frequencies and co-occurrences. A graph of semantic similarities is built, like the ones used after event detection, now for the whole program. A set of representative tweets is selected and the rejected tweets are compared to them in terms of semantic similarity. This part is in being tested. We do not achieve the optimal number of representative tweets and the best threshold on semantic similarity yet. Even so some results are already presented.

IV. Results

The frequency of collected tweets varies during the day and also depends on the day of the week. In a week day 3 distinct periods are distinguishable. From approximately 3 a.m. to 7 a.m. the activity is very low, less than 20 tweets/min. From 8 a.m. to 5 p.m. the frequency is typically between 25 and 60 tweets/min. In the evening the activity is higher, achieving 200 tweets/min, mainly between 9 p.m. and 11 p.m. On the weekend the activity is higher, the frequency varies from 50 to 150 tweets/min between 10 a.m. and 18 p.m. and from 100 to 200 tweets/min between 18 p.m. and 12 a.m. We also notice the existence of sudden peaks in certain minutes. For example, a sudden growth of 200 tweets/min from one minute to other, returning to the same value in two minutes. We realize that some higher peaks coincide with moments in which the Portuguese national team of football scores a goal. We also analyzed the most used *hashtags*. In that top, few were related with TV. Those *hashtags* were mainly about football matches and entertainment shows or contests, typically aired on Sunday evening.

We focus our tweet-program matching analysis on those apparently popular programs, based on the tweets frequency and popular *hashtags*. In Fig. 2 it is presented the number of matched tweets per minute for a program called *Ídolos*. The detected events are marked. Twelve hot moments were identified. Based on the most representative tweets, it was possible to relate each peak with a moment in the program. Analyzing several programs, we observed that the event detection is as successful as the popularity of the program in Twitter. In programs with an average of detected tweets bellow 5 tweets/min there are some small peaks where the relation to an event is dubious. In football matches, goals have a big impact and are always correctly detected.

[10]http://www.cherrypy.org

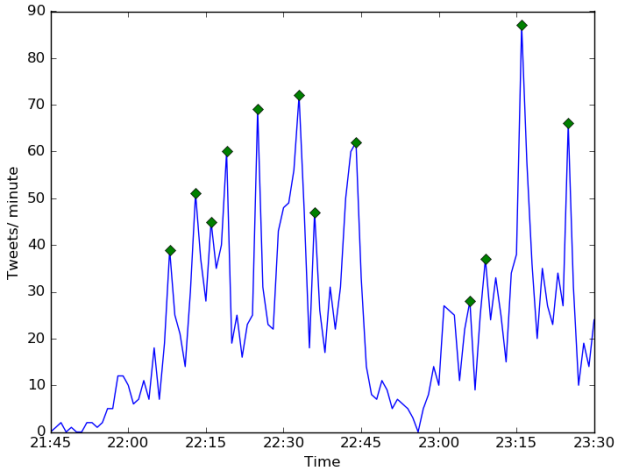

Fig. 2. *Matched tweets frequency for the TV program "Ídolos" on 12th May 2015. The green diamonds represent the detected events.*

TABLE I
Phase 1 matching analysis

TV Program	Avg. Freq. (tweets/min)	Precision
Got Talent Portugal	1.0	0.84
Dança com as Estrelas	3.5	0.95
Benfica x Setúbal	5.0	0.80
Ídolos	29.0	0.99

On the first phase of our work, when we were using manual inserted information as metadata, we computed the precision of our model by manual verification. We tested four programs: *Got Talent Portugal*, *Ídolos*, *Dança com as Estrelas* and *Benfica x Setúbal - Taça da Liga 1ª Meia Final*. The first three are entertainment contests and the last one is a football match for the Portuguese League Cup semi-final. We present the results in Table I. We observe that for the TV contests the precision grows with the frequency of matched tweets. We also see that the matching is less precise for the football match.

After the implementation of metadata extraction from Wikipedia, we compared its performance of the new method with the previous one. We computed not only the precision but also the recall. We considered 10 distinct and separated minutes during the simultaneous emission of two of the programs analyzed before, in a different day: *Ídolos* and *Dança com as Estrelas*. Tweets were manually tagged and after the matching we computed the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). Table II shows the results. We can see that the popularity of the TV programs changed. From the first case to the second, the average frequency of matched tweets for *Ídolos* decreased

TABLE II
PHASE 2 MATCHING ANALYSIS

TV Program	Ídolos		DCAE	
Metadata	Manual	Wikipedia	Manual	Wikipedia
TP	20	20	62	77
TN	1507	1503	1407	1408
FP	0	4	2	1
FN	24	24	80	65
Precision	1	0.83	0.97	0.99
Recall	0.45	0.45	0.44	0.54

TABLE III
PHASE 3 MATCHING ANALYSIS

TV Program	Porto x Benfica	
Matching with similarity	No	Yes
Precision	0.95	0.91
Recall	0.44	0.62
F-measure	0.60	0.73

from 29.0 to 2.0 tweets/min and for *Dança com as Estrelas* increased from 3.5 to 6.4. The use of Wikipedia as metadata source, does not significantly influences the precision of *Dança com as Estrelas*. For *Ídolos* the precision drops but remains above 80%. The reason for this change is on the quantity of spurious terms that exist on Wikipedia pages, even on tables and on the selected sections. The quantity of those misleading terms change from page to page. We can also realize that the recall is low for both cases, meaning that half of related tweets are being ignored. Without using machine learning algorithms but gathering TV programs information from different sources we achieve results with precision values close to what the models described on the State of the Art show.

The ambition to improve the recall lead us to implement a solution that could take advantage from the semantic profile of a TV program. We analyzed this third approach on a classical football match in Portugal: *FC Porto x Benfica - Primeira Liga*. After the construction of the tweets similarity graph for this program, we measured the similarity of the neglected tweets with the 20 most representative tweets. From the 3200 with higher similarity, we calculated the similarity threshold that maximizes the F_1 score: $F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$

We achieve a value of 0.49 for similarity, corresponding to an F_1 score of 0.84. Again, we extracted a set of tweets from 10 distinct separated minutes, and evaluated the performance of our model with and without the use of semantic similarity on matching. From Table III we observe that although the precision lightly decreases, the recall substantially rises with the use of semantic similarity, increasing the F-measure. We realized that some detected tweets are only matched with a context understanding, e.g.: *"O melhor está a jogar"* or *"Jogamos mal pa c**... ganhamos 2-0. Jogamos bem.... perdemos"*. These messages are composed by words that are not found on the TV programs sources. Although it is not possible to take generic conclusions from this analysis, it gave us reasons to deeply explore the use of semantic similarity on the detection of tweets related to TV programs, namely by testing it with other programs and tuning some parameters, like the number of representative tweets and the similarity threshold.

Finally, by observing lists of suggested Youtube videos, from the programs mentioned above, we concluded that the automatic search is successful, retrieving related contents. The videos are always related to moments of the program (same or other episode) or with an highlighted person. For some of the detected moments no video may be retrieved or the retrieved video may not correspond to the event. However that is not happening on events with a big impact (large peaks).

V. CONCLUSION

We have built a solution based on state-of-art social media text mining techniques, applied to the Portuguese language and TV commenting habits. Albeit all the limitations of our system, we are able to successfully detect highlights in a TV program solely based on what viewers comment on Twitter. We improved our model, automating the gathering of metadata and starting a new approach by using semantic properties of the messages. Another extra was included, a successful automatic search of Youtube videos related to the TV programs and their main topics. Future work will focus on improving system performance in terms of matching results and processing time.

ACKNOWLEDGMENT

This work was made possible thanks to a grant by PT Inovação - Project TVPulse. The authors would also like to thank Univ. Aveiro Social iTV research group with whom have collaborated.

REFERENCES

[1] J. Perkins, *Python Text Processing with NLTK 2.0 Cookbook* Packt Publishing, 2010.

[2] V. M. Orengo and C. Huyck, *A Stemming Algorithm for the Portuguese Language* SPIRE Conference, Laguna de San Raphael, Chile, November 13-15, 2001.

[3] S. M. Weiss, N. Indurkhya and T. Zhang, *Fundamentals of Predictive Text Mining* Springer-Verlag London Limited, 2010.

[4] O. Owoputi, C. Dyer, K. Gimpel, N. Schneider and N. A. Smith *Improved part-of-speech tagging for online conversational text with word clusters* In Proceedings of NAACL, 2013.

[5] V. I. Levenshtein *Binary codes capable of correcting deletions, insertions, and reversals* Soviet Physics Doklady, Feb 1966.

[6] S. Mohammad and G. Hirst *Distributional measures of concept-distance: a task-oriented evaluatio* EMNLP '06 Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 2006.

[7] P. Cremonesi, R. Pagano, S. Pasquali and R. Turrin *TV Program Detection in Tweets* EuroITV'13, Como, Italy, June 24âĂŞ26, 2013.

[8] M. Nakazawa, M. Erdmann, H. Hoashi and C. Ono *Social Indexing of TV Programs:Detection and Labeling of Significant TV Scenes by Twitter Analysis* ï£ijWAINA '12 Proceedings of the 2012 26th International Conference on Advanced Information Networking and Applications Workshops, Fukuoka, Japan, Mar 2012.

[9] Y. Genc, Y. Sakamoto and J. V. Nickerson *Discovering Context: Classifying tweets through a semantic transform based on Wikipedia* Proceedings of the 6th International Conference, FAC 2011, Held as Part of HCI International 2011, Orlando, FL, USA, July 9-14, 2011.

[10] F. Abel, Q. Gao, G.-J. Houben and K. Tao *Semantic Enrichment of Twitter Posts for User Profile Construction on the Social Web* 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29 âĂŞ June 2, 2011, Proceedings, Part II.

[11] O. Ozdikis, P. Senkul and H. Oguztuzun *Semantic expansion of hashtags for enhanced event detection in Twitter* Proceedings of VLDB 2012 Workshop on Online Social Systems, Istanbul, Turkey, Aug 31, 2012

[12] A. Vilaça, M. Antunes, D. Gomes, *TV-Pulse: detecting TV highlights in Social Networks* Proc. 10th ConfTele 2015 - Conference on Telecommunications, Aveiro, Portugal, Sep 2015.
[13] A. Clauset, C. R. Shalizi and M. E. J. Newman *Power-Law Distributions in Empirical Data* SIAM Review 51, 661-703, 2009.
[14] D. J. Ketchen and C. L. Shook *The Application of Cluster Analysis in Strategic Management Research: An Analysis and Critique* SIAM Review 51, 661-703, 2009.