TVPulse: detecting TV highlights in Social Networks

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 allows a numerous quantity of applications in the industry of entertainment information.**

This paper describes a system that is capable of detecting TV highlights in one of the most important social networks - Twitter. Combining Twitter's messages and information from an Electronic Programming Guide (EPG) 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. Far from finished, we intend to further develop our system to take advantage of external metadata in order to improve matching rates.

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 perfect 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. 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.

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

[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

In this paper we describe a system built to extract and store information from Twitter and from publicly available EPG's, process text, 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. Very typical 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[3]. All text filtering options should be optimized for each case study.

A second choice to be made in text mining, after the filtering process, are the features to be used[1], [4]. 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.[5]. 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 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[6]. 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[7] 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.[8] 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%.

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). Aditionally we created a repository of several tabels containing extra metadata such as *hashtags* used for each program, slang commonly used in a TV show and players in major sports teams. Ultimately our will is to enhance this repository as automatic as possible from various other sources, but for the time being it is edited by a human.

[4]https://dev.twitter.com/overview/api

[5]https://store.services.sapo.pt/en/cat/catalog/other/meo-epg

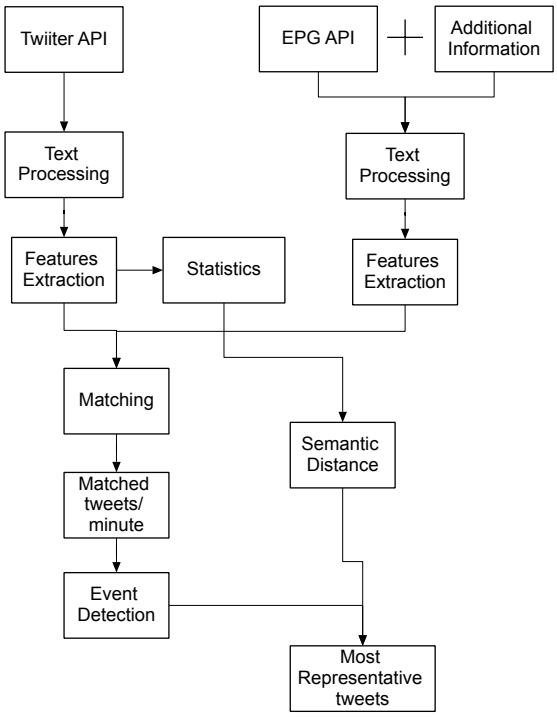

Fig. 1. *System diagram.*

Each tweet or program EPG 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 [3]. 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, description, short description and channel name and acronym. The same type of features are extracted from the metadata tables with terms and *hashtags* associated to each program. 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 a semantic features based on terms distributional profiles. To measure the distance of two words we use the cosine distance[7]:

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

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[9], also known as 80-20 rule, and the elbow method [10]. 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.

Our work is made available to others through a public API. The API implemented using CherryPy [6] responds to queries with matched tweet-program pairs for a given time interval. 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.

Semantic similarity may improve precision and recall of our model. A particular semantic can be built for each set of tweets associated to a specific program. Each TV program can have a specific graph, as described before. For a new tweet it is possible to compute its proximity to a program by computing a distance based on similarity between its words and words on tweets already associated to that program, using the own semantic. We tested this with some programs. Tweets are associated to a program using the previously described method, a semantic is built based on their terms, and then any new tweet can be compared to that program. The used measure of similarity between two tweets (t_1 and t_2) is a weighted average of cosine distances of words pairs with a penalization for very frequent words:

$$\begin{aligned} &Sim(t_1,t_2) = \sum_{i,j} \frac{s(w_{1i},w_{2j})}{|t_1|\times|t_2|},\ w_{1i}\in t_1,\ w_{2i}\in t_2, \\ &s(w_{1i},w_{2j}) = Cos(w_{1i},w_{2j}) log_2\left(\frac{N}{tf(w_{1i})}\right) log_2\left(\frac{N}{tf(w_{2j})}\right), \\ &|t_k| = \sqrt{\sum_i^{n_k} log_2^2\left(\frac{N}{tf(w_{ki})}\right)} \end{aligned} \quad (3)$$

where $Cos(w_{1i},w_{2j})$ is obtained from 1 and N and tf are the number of tweets and the term frequency on the set used to build the semantic.

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.

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

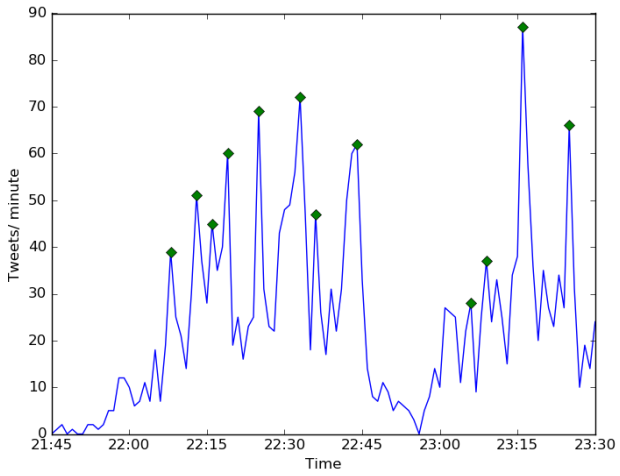

Fig. 2. *Matched tweets frequency for the TV program "Ídolos" on 12th May 2015. The green diamonds represent the detected events.*

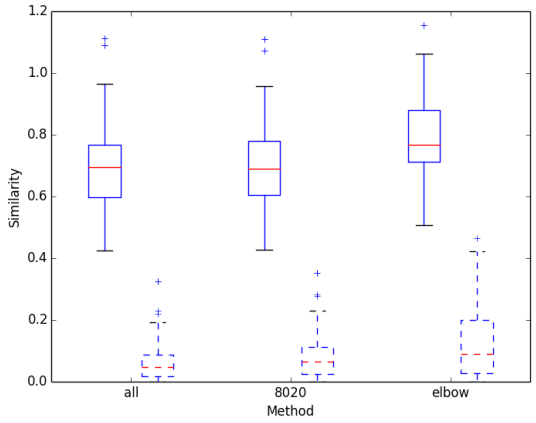

Fig. 3. *Distribution of similarity values between tweets related to a same program (solid line) and to two distinct programs (dashed line), based on a semantic built with tweets related to the program "Dança com as Estrelas". For boxes with solid lines the test was done with the tweets also related to that program, while for boxes with dashed line the test was done with tweets related to the program "Got Talent Portugal".*

TABLE I
PRECISION OF THE MATCHING MODEL

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

For 4 programs we computed the precision of our model by manual verification. Those programs are *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.

Finally, we consider the computation of semantic similarity between pairs of tweets. We realize that when a semantic from a particular program is used, tweets related to that program have a stronger similarity between them and with tweets from the training set. Fig. 3 shows a distribution of similarities of pairs of tweets, being related to a same program and being related to two distinct programs. We show results using all the terms on the co-occurrences matrix and doing data reduction: 80-20 and elbow methods. This example shows that a program may have its own vocabulary and the construction of a proper semantic allows the association or rejection of new tweets to that program. We also see that the 80-20 method does not have a strong influence in the results, but elbow method improves the similarity measure between the tweets. This fact was also observed on tests with other programs.

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. Future work will focus in automating and gathering extra metadata to increase our matching rates, we also intend to achieve near-realtime detection of the highlights.

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] C. J. van Rijsbergen, S. E. Robertson and M. F. Porter, *New models in probabilistic information retrieval* London: British Library, 1980.
[3] V. M. Orengo and C. Huyck, *A Stemming Algorithm for the Portuguese Language* SPIRE Conference, Laguna de San Raphael, Chile, November 13-15, 2001.
[4] S. M. Weiss, N. Indurkhya and T. Zhang, *Fundamentals of Predictive Text Mining* Springer-Verlag London Limited, 2010.
[5] 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.
[6] V. I. Levenshtein *Binary codes capable of correcting deletions, insertions, and reversals* Soviet Physics Doklady, Feb 1966.
[7] 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.
[8] P. Cremonesi, R. Pagano, S. Pasquali and R. Turrin *TV Program Detection in Tweets* EuroITVâĂŹ13, Como, Italy, June 24âĂŞ26, 2013.
[9] A. Clauset, C. R. Shalizi and M. E. J. Newman *Power-Law Distributions in Empirical Data* SIAM Review 51, 661-703, 2009.
[10] D. J. Keychen and C. L. Shook *The Application of Cluster Analysis in Strategic Management Research: An Analysis and Critique* SIAM Review 51, 661-703, 2009.