EXPERIMENTAL RESULTS ON CONTENT ANALYSIS USING GOOGLE SET

Cătălin Constantin Cerbulescu, Ştefan Udriştoiu,
Claudia Monica Cerbulescu

University of Craiova, Faculty of Automatics,
Computer and Electronics, ccerbulescu@software.ucv.ro
University of Craiova, Faculty of Automatics,
Computer and Electronics, stefan@software.ucv.ro
Carol I College, Craiova

Abstract: An important problem in both actual research and software development is content analyzing. The results of those researches are reflected in eliminating unimportant messages (spam-filter) and selecting the most important messages from a set. Present paper presents a content analyzer algorithm, implemented in a web-based application, among with his experimental results. According to an interest domain (defined by a set of words), the target content is analyzed. The result, a vector of floating numbers, is processed and analyzed, according to statistical methods. This approach is based on http://labs.google.com/sets to group words by importance and relevance.

Keywords: pattern recognition, algorithms, data processing, machine learning, statistical Analysis.

1. INTRODUCTION

Important researches were made during past years in the field of classifying and retrieving data according to some specified rules. As a result of the extremely large amount of data that needs to be analyzed and classified, the presence of human operator needs to be replaced by software tools that react faster. As a response to those needs, several commercial applications were tested and developed.

Basically, the steps followed in such applications are:

1. Tagging or atomizing words from the analyzed content. This operation supposes to analyze and convert all sentence parts (nouns, pro-nouns, verbs) on every form, gender and number to nouns or verbs. In this process, some sentence parts, such as conjunctions and pro-nouns, can be eliminated.

This term (tagged) hides some algorithm of extracting the most relevant form for each word, by taking or not in consideration the neighbours. A tagged phrase will contain:

- nouns at singular;
- verbs at infinitive;
- no conjunctions

so that, for example, the text:
I was in vacation on sea with my boat. The door window was broken. My team uses a file manager. My central unit processor
is tagged as:
be vacation sea boat door window be broken team use file manager central unit processor

As shown in the example, the result of this process will be a set of words, related to the primary sentence in witch the general idea of the sentence is kept. Several software tools were developed to solve this problem.

2. Analyze the resulted word set in order to establish his importance.

If approach solutions for step 1 already exist, the step 2 is the most important part of the problem and it requires defining defines a word similarity. Two main directions emerge:

1. Defines a word similarity by manually means. This approach can be implemented by building a table of similarities for each used word;
2. Defining word similarities by analyzing large amount of data, large set of words used in sentences. This is the so known word clustering approach. One important step forward on this approach was made by google by making available the so called: google Set, at http://labs.google.com/sets. This free tool allows word clustering by grouping words by sense and patterns

Content analyzer applications try to extract the most relevant content from a data set. Also, researches were made in order to be able to ordering some data according to their importance.

Several proposed solutions for pattern recognition were based on analyzing the current amount of data and use some machine-learning algorithms so that the "current achieved experience" will be available for other data processing.

Dekang Li (1998) defines a word similarity measure based on the distributional pattern of the words. The similarity measure allow to build a thesaurus using a parsed corpus. Andrew Y. Ng (2001) discussed some algorithms that cluster points using eigenvectors of matrices derived from the data. It shows that there are a wide variety of algorithms that use the eigenvectors in slightly different ways and many of these algorithms have no proof that they will compute a reasonable clustering. Andrew Y. Ng and others presents, in "On spectral clustering: Analysis and an algorithm", a spectral clustering algorithm that can be implemented using few code lines in Matlab.

2. GOOGLE SET

Available at http://labs.google.com/sets, google set offers a new way of finding word patterns. Based on the extremely large number of sentences available on the Internet, google set finds the most common words related to words user enters.

The product is not yet officially documented but some discussions and tests show that this is probably the best tool now available for word patterns. Using web crawlers that navigate from one site to another, extracts the meta-tags and site content, google builds a huge database. This database is now exploited, besides the all known google query, by getting word patterns.

If you test google set, you enter some words (no more than 5) and you will get a word set (large or not), related to the words you entered. For example, if you entered red, green, blue you will get about all the colours name.

Google Set acts by using some algorithms to detect a pattern on the words user defines and then apply this pattern to the database he owns. The most significant results became available.

Google set successfully recognized the category the tested words belong and generates more words related to them. As specified before, no official information are available but it probably use classification algorithms so that given the test data set, the algorithm classifies it into prior clustered sets and then based on the similarity and the size specified finds the nearest similar words in the group.

One important observation about the returned results is that they don't depend of the input words order. So, if you entered the same input words but in different order, it is highly probable to get the same results.

An important restriction consists in the fact that the words are tagged so that, in order to obtain some results, tested data need to be tagged either manually either by using a software tool.

3. DEFINING THE PROBLEM

Seminal point for this research was a project that needs to automatically classify a large amount of messages by their importance, mark the most and less important messages related to my interests domain, defined by some witness words.

As an example, if I define my interest domain in terms: {thread software java database relational} which one of the following messages is the most important.

MB1={Wait for other good news from you}, tagged as follow:

M1={wait other good news from you}

MB2={I expect to take a few more days to go through the whole database and java code before final acceptance, so hope this timeframe is ok} tagged as follow:

M2={expect take few more day go through whole database java code before final acceptance hope this timeframe be ok}

MB3={Ensure that messages on threads synchronisation status screens are relevant and helpful and that the users is advised of any error conditions. Errors are logged in the database error table. Currently there are some conditions where low level system error messages (e.g. Java Beans) are shown, nothing is shown, or graphics are missing} tagged as follow:

M3={ensure message thread synchronisation status screen be relevant helpful user be advised error condition error be log database error table currently there be some condition where low level system error message java beans be show nothing be show graphic be missing}

For a human operator, it is obviously that the first text, M1 has no importance related to the imposed items. The other two messages are a little bit hard to qualify. A human operator can consider that M3 can be more significant.

Besides this, it will be more interesting if the human operator can extract the most significant part of text.

Although not visible at first view, the problem of defining the interest domain by no more than 5 words can be very difficult. This is the factor that will have the biggest influence for the results of the research.

4. PROPOSED ALGORITHM

The proposed algorithm will try to form a set of continuously words and extract a numeric value, then a number vector that will define the text according to the interest domain. Using this vector of numeric values, the algorithm will analyse the entire text, extract some characteristics that will be used to define the content characteristics.

We note with M, the word set of the message that will be studied. The set M will have n elements (words). From this set of words, we sequentially extract all the sets of s continuously words. If we note those sets with T_k, k=1,n-s+1

For the message M_1={wait other good news from you}, n=6. If s=5, we form the following T_k word groups:
T_1 = {wait other good news from}, T_2={other good news from you}.

A function g, based on google sets,

$$g: K \rightarrow S$$

is defined on the set of keys (K) with values on the set of google sets (S).

An important notice can be made about the way the keys K can be used. The larger the number of keys K by witch we describe out interest domain is, the more focused values for the resulted related word. So, for a "relaxing" search, I can define our interest domain by 1-2 keywords. Instead of a relaxed search, if we define our interest domain by 5 keywords (the larger number google set allows) we could obtain a "tide" search.

The function will use google sets to form a group of related words (S) based of some keys (K).
g(red, green, blue) = (Blue, Red, Green, Black, White, Yellow, Orange, Purple, Brown, Gray, magenta, cyan, Pink, Browser, …..)

An imposed word set, I, is used, so that the related words, obtained by querying google sets, are

$$g(I) \rightarrow S_I$$

and a tested word group T_K will have the related words

$$g(T_K) \rightarrow S_{Tk}$$

Building the word sets S_{Tk} implies the same rules as described above. If we use a small key group words (value for s is 1 or 2) we obtain a "relaxed" word groups, related to impose keys. For a close related search, a value s=5 it is recommended.

A function f,

$$f: (SxS) \rightarrow V_{(1x\ n\text{-}s+1)}$$

is defined so that it takes values from 2 word sets. A more specific definition of the function can be considered:

$$f: (S_I\ x\ S_{Tk}) \rightarrow V_{(1x\ n\text{-}s+1)}$$

The returned value is an array of floats values from range 0...1. Each value corresponds to the "importance" of the tested word group. A value equal to 0 means the word group has "no relevance". A value equal to "1" means the word group has the maximum relevance (the word group is the same as the imposed set).

The array V of floats can be considered to be characteristic for the tested message, according to some imposed words. From this point, little importance is given to the imposed words and to the analysed message.

5. ANALYZIG THE VECTOR V

Having the vector V of float values, range 0..1, brings us to another problem that consists in analysing the vector by statistical methods so that the obtained values can be most relevant for the analysed message. In the test results we consider some statistical function:

- maximum value, a measure of the most important word group in the message. Several "most important word groups" can exist. The number of those "most important word groups" reported to the number of word groups can also be used as a measure of text importance;

- medium value of the vector V was used;
- standard deviation for the vector V,

$$\sum_{i=0}^{n-s+1} \frac{\left(v_i - \bar{v}\right)^2}{n-s}$$

as a measure of the deviation from the medium value. In our first tests, this characteristic of the message was basically used to classify the messages according to their importance, as show in the example below.

The testes presented below show that the standard deviation could be used to automatically describe the importance of the analysed message. Yet, results on large real messages show that the standard deviation is not enough to order messages by importance.

6. EXPERIMENTAL RESULTS

For tests, an application, located at http://193.231.39.113:8080/content_analyzer was used.

For this experiment, the field of interest was defined in terms {thread software java database relational}. The following messages were analysed:

M1={wait other good news from you}
M2={expect take few more day go through whole database java code before final acceptance hope this timeframe be ok}
M3={ensure message thread synchronisation status screen be relevant helpful user be advised error condition error be log database error table currently there be some condition where low level system error message java beans be show nothing be show graphic be missing}

The characteristics for the message M1 are:
Most relevant word group: {wait other good news from} {other good news from you}
An average value: 0.0
A standard deviation: 0.10327955

In the next table some detailed results for the message M2 are presented.

Nr	Word Group	Relevance
1	(expect) (take) (few) (more) (day)	0.0
2	(take) (few) (more) (day) (go)	0.0
3	(few) (more) (day) (go) (through)	0.0
4	(more) (day) (go) (through) (whole)	0.0
5	(day) (go) (through) (whole) (database)	0.0
6	(go) (through) (whole) (database) (java)	0.3333
7	(through) (whole) (database) (java) (code)	0.3125
8	(whole) (database) (java) (code) (before)	0.3125
9	(database) (java) (code) (before) (final)	0.3125
10	(java) (code) (before) (final) (acceptance)	0.0208
11	(code) (before) (final) (acceptance) (hope)	0.0
12	(before) (final) (acceptance) (hope) (this)	0.0
13	(final) (acceptance) (hope) (this) (timeframe)	0.0
14	(acceptance) (hope) (this) (timeframe) (be)	0.0
15	(hope) (this) (timeframe) (be) (ok)	0.0

The characteristics for the message M2 are:
Most relevant word group: go through whole database java
An average value: 4.133333
A standard deviation: 1.2986722

In the next table, parts of the analysis results for M3 are shown:

Nr	Word Group	Relevance
1	(condition) (error) (be) (log) (database)	0.0208
2	(error) (be) (log) (database) (error)	0.0208
3	(be) (log) (database) (error) (table)	0.0416
4	(log) (database) (error) (table) (currently)	0.0416
5	(database) (error) (table) (currently) (there)	0.0416
6	(error) (table) (currently) (there) (be)	0
7	(table) (currently) (there) (be) (some)	0
	....	
8	(low) (level) (system) (error) (message)	0
9	(level) (system) (error) (message) (java)	0.0208
10	(system) (error) (message) (java) (beans)	0.0208
11	(error) (message) (java) (beans) (be)	0.0208
12	(message) (java) (beans) (be) (show)	0.0208

13	(java) (beans) (be) (show) (nothing)	0.0208

The characteristics for the message M3 are:
Most relevant word group: {level system error message java} {system error message java beans}
An average value: 0.5135135
A standard deviation: 0.19368063

Several experiments performed show that the standard deviation, yet important, was not enough to order test messages by importance.

For short messages, the results on analyzing the standard deviation can be resume as follow:

- The difference between an important and less important message is relatively high (from 3…. To 0.1…)
- By imposing a border value for standard deviations (ex. 2.00), we can select from a list of messages those with great importance, uncertain and less importance.

Using only standard deviation, previous messages can be descended sorted by importance as follow: M2, M3, M1.

Yet, a human operator, witch will be consider as a standard and referred next, may consider that the message M3 is more important than M2. Experiments conducted on real messages suggest that standard deviation is not enough due to the fact that, although a message can have a "tide-close" sense, his relevance, using this algorithm can be 0.

Message M3 reveals another important fact in human conversation. That is, following an idea, important sentences in a message can came among with other sentences that are indispensable, such as: salutation or common sentences.

If we "relax" the search by using s=2 for the number of the imposed words (thread, software), we get:

The characteristics for the message M1 are:
An average value: 2.5
A standard deviation: 0.36696956

The characteristics for the message M2 are:
An average value: 0.0
A standard deviation: 0.08377077

The characteristics for the message M3 are:
An average value: 0.08108108
A standard deviation: 0.083520055

Those results are more appropriate with the ones human operator expects to get. The high average value of the vector V, for the message M1 is due to the message shortness. It has only 6 words. Yet, the standard deviation is relatively high, fact that suggests that only a few groups of words are relevant and all the rest all completely irrelevant.

The results in analysed messages M2 and M3 suggest that the average value of the vector V, for M2 is lower than a trigger value, so it is 0. Although the standard deviation of messages M2 and M3 are relatively equals, the fact that standard deviation for M3 is less that the one for M2 and the average value of the vector V for M3 is higher that the one for M2, suggests that message M3 is more important than M2.

7. RESTRICTIONS AND FUTURE DEVELOPMENTS

Tests on the application show some directions for future developments:

1. as mentioned in the chapter 3, choosing a right set of words that will describe the interest domain will primarily affect the vector V of numbers and the final results. At this point, some machine learning algorithms can be used in order to re-define the imposed word set if the results are not very good. Only human operator can make the difference between a good and a bad result. This is possible by showing to the operator messages with the highest results;

2. modifying the factor s will produce modifications in the results vector V. The available application uses only the value s=5 for the number of the imposed words. Reducing the value of s will produce a more relaxing build of the related words family so that the average value and standard deviation for the vector v will be both greater. It is possible that the words included in the returned set will be far (or very far) related to the imposed set the human operator have in mind. That not necessary means that the message became more important or the vector V is more significant. It is possible that the most relevant part of the message will be different by using s=2 and s=5. Experiments show that, by using s=2, the results are more like the ones human operator expects;

3. an important restriction lies in the way the google set builds the related word family. As mentioned, there is no official documentation about this service but experiments show that google set use, as a research base, web content stored in their databases. An imposed key set returns a "medium" close related word group, no matter if the source for google analysis was a technical, medical or literary text;

4. other indicators need to be found in order to get a more accurate result on the content analysing. Those need to consider the length of the message, the number of relevant words related to the words number in the message;

5. terms like “more important”, “less important”, “higher relevance”, “lower relevance” suggests that a fuzzy approach can be used. The trigger levels used in a fuzzy algorithm can be permanently adjusted so that the result of the message sorting by relevance operation will be the one human operator expects to get.

8. CONCLUSIONS

Present paper presents a content analyzer algorithm, implemented in a web-based application, running at http://193.231.39.113:8080/content_analyzer/), among with his experimental results. According to an interest domain (defined by a set of words), the target message content is analyzed. The result, a vector of floating numbers, is processed and analyzed, according to statistical methods. This approach is based on http://labs.google.com/sets to group words by importance and relevance.

The field of interest is defined using a number of 1 to 5 keywords. Although the algorithm can extract most important sentences from the message, according to imposed keywords, with enough accuracy, his results on sorting messages by relevance are not always the way human operator expect to be.

As expected, experiments show that a more relaxed search, defined by 1 or 2 imposed words, produce a higher average value of the vector V and a decrease of the standard deviation. If the search is defined by 5 keywords, a lower average value of the vector V and an increase of the standard deviation will be produced.

If the field of interest is well defined using 1 or 2 words, this approach will produce the better results. As shown, problem now became a linguistically one, that is better defining my interest field using 1 or 2 words.

9. REFERENCES

Andrew Y. Ng, A. Ng, M. Jordan, and Y. Weiss. (2001) *On spectral clustering: Analysis and an algorithm. In Advances in Neural Information Processing Systems 14: Proceedings of the 2001 Neural Information Processing Systems (NIPS) Conference, pages 849-856, Cambridge, MA, 2002.*

Cerbulescu Catalin (2004). Content analysis using google set. In *Analele Universitatii Craiova, 2004,* Craiova, Roumania, oct 2004.

Dekang Li (1998). Automatic Retrieval and Clustering of Similar Words. In *Proceedings of COLINGACL '98*, pages 768--774, Montreal, Canada, August 1998.