reviewed paper

Data Analysis Methods for Urban Planning – Problem-Oriented Stakeholders Maps Building

Victor L. Kuriashkin, Natalia A. Zhukova

(Victor L. Kuriashkin, Scientific and Engineering Center of the Electrotechnical University , St. Petersburg, Russia, nazhukova@mail.ru)
(Natalia A. Zhukova, Scientific and Engineering Center of the Electro-technical University, St. Petersburg, Russia,, vk4arm@gmail.com)

1 ABSTRACT

A stakeholder is that which can affect or be affected by the actions of the business or interests area as a whole. The stakeholder concept was first used in a 1963 internal memorandum at the Stanford Research Institute. It defined stakeholders as "those groups without whose support the organization would cease to exist." [1]. The theory was later developed and championed by Edward Freeman in the 1980s. Stakeholders mapping is a creation a map of stakeholders in simple for usage format. This paper treats automatically creation problem-oriented stakeholders maps for urban planning purposes. Proposed method includes decision data mining, clustering, creation a stakeholders map in the mind map format.

2 INTRODUCTION

Today the urban economy is a complicated heterogeneous system, which includes a big number of different elements. Each concrete problem can affect or affected by the various organizations, social groups, officials and simple citizens. In a modern world resolving most complicated urban problems is impossible without data analysis, finding, who may be interested in the current state, who need or can to help to change it and which groups should be affected.

Stakeholders theory was created for business problems resolution, initially for competitor analysis, but nowadays it is a part of the decisions making in a different areas.

We propose to analyze on-line media, including social media (websites and social networks), official electronic directories and official web sites for automatically analysis and stakeholders maps creation. In additional we offer processing an official statistics and documents, where it is accessible.

As a result of analysis we builds a kind of knowledge base, and another task is a presenting this map in usable format for decision makers.

Classical systems for operations with knowledge base are not simple for use by businessman or officials. For example, well known Protege system [2], created by Stanford University, could be used only by knowledge engineers or advanced users. Modern management should have an instrument, which can be simple to integrate into the business processes, inserting into reports and presentations, and which could be extremely simple to use in any technical environment – desktops, web and mobile.

We propose to use a Mind Maps format. There are some commercial applications, like a MindJet by the JetBrains (http://www.jetbrains.com/), but we propose to use open format mind maps by the open source project FreeMind [3]. User can export FreeMind xml files into the html and flash for integration to the websites or presentations.

In this article we propose to use a 4-levels maps.

- Problem name (Topic). Examples: medicine, education, road planning, migration politics. In some cases it should be more precision problems like: “traffic jam in the center of the city”, “criminal situation in the specific area”
- Objects, affected by actions in this problem or to be affected.
- It could be names of officials, organizations/businesses, businessman/competitors names, some social media authors, bloggers, etc. For further implementations we can include into this level affected small social groups.

This level includes 2 types of items:

- Affiliate objects.

Example: organization, where author works, relatives, and so on.

- Affected problems. In general, it is a set of links to another problems in the 1st level.

3 INFORMATION RETRIEVAL

System should be designed for working in semi-automatic or full automatic mode. It should gather information from the on-line media for real-time mind maps rebuilding. This option should make a possibility always to work with actual information.

It means that critically important part of the system should be a web crawler and pages parser. We designed a crawler, which can solve a different types of technical problems.

For example:

- article text extraction from the different web pages formats with knowledge extraction using Kuriashkin and Kazekin pages processor [4]
- render a page and execute a javascript on the server side for retrieving information from the ajax requests

Another problem is that we can not trust to anything from the Internet, and it means, that we should build a fraud-safe system.

In some cases, on-line authors could add some personalities into the discussions about criminal situation (black PR).

Fraud safety in our case is a dividing information sources into the trust-level groups and extraction author names for evaluation an authors confidence level.

Data sources are divided into the following groups:

- Official organizations (government, business) and directories
- Serious media
- “Yellow” press
- Blogs and social networks

Current system should skip article comments in the on-line media.

As a social networking data source we use a livejournal.com, which is very popular in Russian Internet segment. We could not propose to use another sources as a social right now, except it, because of a big number of spam messages and copy-pastes between all social on-line media sites.

4 BUILDING A VECTOR SPACE

For data analysis texts should be converted into the vector space. Classical approach in building vector space from the text includes: stemming, n-grams extraction, n-grams tf-idf calculation, selection for the vector space building only significant n-grams. By making decision about what n-grams are significant it is possible to increase a quality of the space for the concrete problem. Usually vector space builders skips n-grams with too high tf-idf value, because it could be syntax errors or very specific terms, and a “long tail” with low tf-idf values – non-significant terms.

For our task, building a vector space, besides the classical approach should include directory records analyzing.

In addition, for Russian language extremely important to process: prefixes and word endings, which could be extracted in a separate dictionary and could be used as a separate vectors elements.

Thus, we have 3 groups of the vector elements: directory components, syntactic components and n-grams analysis components. For each space types components we calculated a separate kind of tf-idf measure, which it is possible to use in the distances measure. Weights could be selected for tuning our algorithm, but we suggested to use:

w=2 for directory

w=1 for syntactic

w=0.5 for texts from the official sources and 0.3 for blogs and social networks.

All weights should be corrected for the concrete problem investigation.

5 TEXTS DISTANCE MEASURE

We have a 3 type of vectors components, so, we should use a weighted distance measure. In our case it is possible to use a weighted Euclidean distance or weighted Manhattan distance. In case of Manhattan measure [5], we have reduce an importance of the outlets, so, this approach is more usable for our case.c

We propose to use following distance measure:

$$D=\sum w_i*(A_i-B_i) ,$$

where w is a weight, selected for the components category.

6 PROBLEMS EXTRACTING AND MIND MAP TREE BUILDING

We propose two problems extraction methods:

- “Learning without teacher” approach
- Using a classification and learning dataset for the problem specification.

First method should be useful for the timely response to the emergence of the problem. We propose to use a Fuzzy C-means (Soft K-means) clustering algorithm [6].

The FCM algorithm attempts to partition a finite collection of n elements

$$X=[x_{1,}x_{2,}\dots x_n]$$

into a collection of c fuzzy clusters with respect to some given criterion. Given a finite set of data, the algorithm returns a list of c cluster centers

$$C=[c_{1,}c_{2\dots}c_c]$$

and a partition matrix

$$W=w_{ij}\in|0,1|$$

where each element w_{ij} tells the degree to which element x_i belongs to the cluster c_j

We propose to use a standart objective function for minimization:

$$w_k(x)=\frac{1}{\sum\left(\frac{d(center_k,x)}{d(center_j,x)}\right)^{\frac{2}{m-1}}}$$

The fuzzyfier m is a level of clustering fuzziness. In the common cases m=2, and we use this value of the fuzziness for our case.

For the real analyzing process we suggest to use a different number of initial clusters for receiving adequate results.

In our project we used an Apache Mahout library [7] for clustering process, which can use a calculations in the distributed or Cloud computation environment in case if we need to analyze a big amount of data or need to receive result in a critically short time or even in real time.

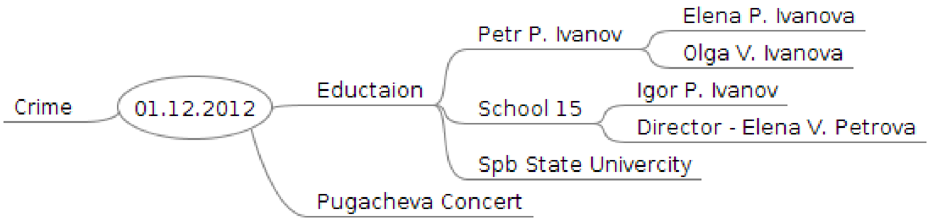
We can suggest to use such computation in the Amazon Cloud, because significant computing power we need only in process of data analyzing, which could not be a permanent process.

The result of the clustering is a set of clusters, which can be processed by operator for selection a names for each cluster, which can be used as a name of the problem.

System could select a set of directory objects, which could be used as an objects, could be affected or affect by the selected problem. It could be a person, government or business organization.

Next step is an association of this object with another objects through common problems/clusters.

Example of the part of the map could be the follows:



Where is possible to add a data sources.

Second approach, which is called “Supervised learning” implies a classification and test dataset using. We suggest to use a Wikipedia articles categories for the root of classification.

Work flow algorithm should contains following steps:

- Getting an articles tree from the Wiki. It could be not a full tree, but just a part, which is interesting for the customer.
- Crawling each category and subcategory for the wiki pages, extracting texts from the each page.
- Crawling on-line data sources and extracting texts, same as in the first approach.
- Use a Naive Bayes algorithm for finding the best category (problem, in our terms) for each predefined category.

$$p(C|F_{1,}F_{2,}\ldots F_n)=\frac{p(C)*p(F_{1,}F_{2,}\ldots F_n|C)}{p(F_{1,}F_{2,}\ldots F_n)}$$

In our case we should calculates a comparison of probabilities, denominator is a constant for the selected data set, so, we can decrease a computations to the numerator calculation and comparison.

We proposed to use a NaiveBayes [8] (MAHOUT-9) algorithm, implemented in the Apache Mahout library.

- Extracting directory objects from the selected texts and place it into the second level of the mind map tree.
- Selection an objects from the another problems, where each of the selected object has found and creation next level of the tree.

We proposed to use a Naive Bayes classifier, because it gives satisfactory results, but it is not correct in the strict sense, because it works in case if components in the vector space is independent from each other. In our case, n-gram analysis could contains terms, includes in the directory term set. This algorithm is not use a weights, selected in the vector space, and could takes not precision enough results, but it could works correctly for the case of small number of categories or in case of searching for the events in the news flow for the selected categories and problems.

Certainly, for the problems classification could be used not only Wikipedia, but any another data source. Better approach is an official data sets usage. In this case it is possible to use only objects finding in the texts from the Internet.

7 HELPFULNESS OF THE SYSTEM

System, which is based on the proposed approach, could be used for on-line monitoring a social media and building an easy to use mind maps. It could be used by government organizations for finding an affected stakeholders of the selected problems (in case of predefined problems classification) and making investigations of the media environment for some important problems for the urban planning. Business organizations can use approach for finding a competitors in the specific area of activity, customers and suppliers.

Proposed approach can be used for both long-term analysis and short-term problems monitoring.

8 PROBLEMS AND SOLUTIONS

Systems is depends on the sufficiently large directory of the personalities, organizations and government structures. Partially it could be found in the Internet using external services. We can suggest to use Open Calais API (Reuters inc) [9], which was used for some project for the Forbes Media. Problem is that it have not Russian texts, accessible for the on-line API. For some cases it is possible to translate texts to English language, extraction personalities and organization names (Reuters has this base for Russia) and translate it back. It could be a source of some mistakes, but useful for some non-critical investigations.

Proposed algorithms could be tuneup with the following coefficients:

- m – fuzziness – for the clustering. Default value is 2
- n – n-gram length.
- Google [42] uses n<=5, but in our case we proposed to use a 3,4,5-grams, filtered by tf-idf measure
- tf-idf borders for the vector space building.

We can not build a space for all texts elements, and propose to make a filtering and reducing a vector space in process of crawling and texts processing to reduce a number or calculations and memory efficiency. This approach could save memory, but it could be follows to the loose of the some data, which is means, that it is unusable for algorithm tuning for the selected use-case.

9 REFERENCES

(1) Freeman, R. Edward (1984). Strategic Management: A stakeholder approach. Boston: Pitman. ISBN 0-273-01913-9.

(2) http://protege.stanford.edu/

(3) Asmaa Hamdy, Mohamed H. ElHoseiny, Radwa Elsahn, Eslam Kamal, Mind Map Automation (MMA) System. SWWS, Las Vegas, Nevada, USA , 2009

(4) Kuriashkin V, Kazekin M, smart data extraction from the raw web pages https://github.com/vk4arm/dartanianparser

(5) Elena Deza & Michel Marie Deza (2009) Encyclopedia of Distances, page 94, Springer.

(6) Nock, R. and Nielsen, F. (2006) "On Weighting Clustering", IEEE Trans. on Pattern Analysis and Machine Intelligence, 28 (8), 1–13

(7) http://mahout.apache.org/

(8) https://cwiki.apache.org/confluence/display/MAHOUT/NaiveBayes

(9) http://www.opencalais.com/