Implementation of the Recommendation System for the "Open Karelia" Information System

Kirill V. Krinkin, Tatyana A. Berlenko
Saint-Petersburg State Electrotechnical University (ETU)
Saint-Peterburg, Russia
{kirill.krinkin, tatyana.berlenko}@fruct.org

Mark M. Zaslavskiy
Saint-Petersburg National Research University of Information Technologies, Mechanics and Optics
Saint-Peterburg, Russia
mark.zaslavskiy@fruct.org

***Abstract*—This paper describes proximity points approach for building recommendations in information systems with anonymous access and semi-structured data. Formulas and algorithms of proximity points calculation are given. The approach allows creating a set of recommended objects without using any information about user behavior using only the content of information system. Due to flexibility the approach can be applied to various set of information systems. Implementation details of the "Open Karelia" system are given as an example.**

I. Introduction

Today recommendations systems became one of the most important parts of websites. The online movie catalog Imdb [1], Amazon [2] online shop, a Youtube [3] video hosting can serve as an example of this phenomenon. One of the reasons of wide use of similar systems is possibility of deduction of the user on the site by providing the additional information adequate to preferences of the user and contents of the current web page [4]. The most effective result is reached by taking into account user behavior on the website (visited pages statistics, time spent, ratings and comments). However, in several cases websites allows only anonymous access (without identification of the individual user) and therefore ratings based approach can't be used. In that case possible decisions are calculation of recommendations using the system content itself. Application of this approach to concrete information system depends on several tasks: objects similarity metric choice, unstructured data processing, recommendations correctness test. Because these tasks does not have any universal solution therefore creation of recommendatory system for the site with anonymous access is an actual task.

II. Information system "Open Karelia"

Information system "Open Karelia" was developed in 2014 within the Russian-Finnish framework of "Euregio Karelia - Museum hypertext" project [5]. Main goal of the system is an access to exhibits and exhibitions of Russian and Finnish Karelia museums. For achieving this goal the system solves following problems:

- data input;
- data storage;
- data access and analysis throw the public API;
- user access by web-frontends.

The data processed in the "Open Karelia" system has different structure and format due to big number of "Euregio Karelia - Museum hypertext" project participants[5]. That's why the system uses NoSql data storage MongoDB [6]. The basic entity in the discussed system is Object. The Object is a set of text fields, which describe real museum exhibit, building, multimedia object, article, graphic material or document. The only mandatory field for all objects is the name field, other ones can be omitted or set in different combinations. The system data also contains interactive exhibitions plans, thematic object sets (histories), tags, multimedia files. All possible fields of "Open Karelia" objects are shown at Fig. 1. For the simplicity we consider "Date" field as all data about creation or discovery date and time for the real exhibit or object related with system's "Object" entity instance. For geodata processing in the "Open Karelia" system the open LBS-platform Geo2Tag [8] is used.

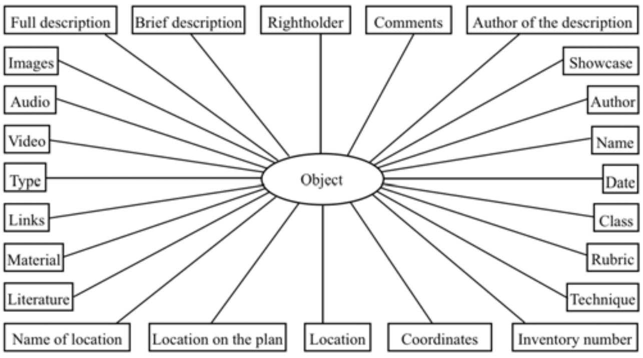

Fig. 1. All possible fields of "Open Karelia" objects

Information system "Open Karelia" provides API for object tags markup, processing, filtration and statistics output. Tags are inputted manually by system administrators; tags markup is performed automatically using word form dictionaries.

User data-access interface of "Open Karelia" is implemented as web-frontend using web-applications, which were written with Python language, Jinja2 and Flask frameworks. At the current moment, the system has three user frontends, which are shown at Fig. 2.

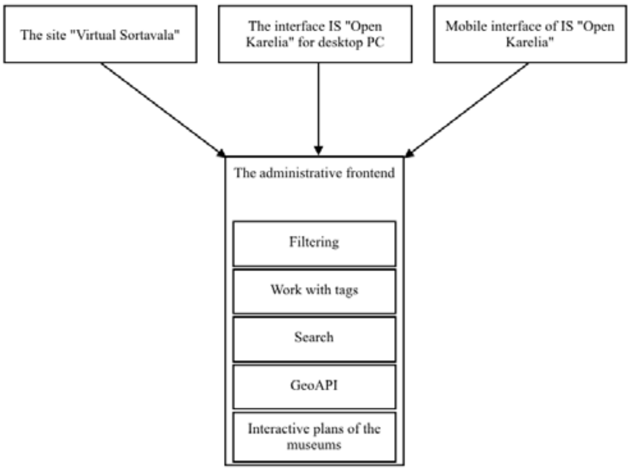
The site "Virtual Sortavala"
The interface IS "Open Karelia" for desktop PC
Mobile interface of IS "Open Karelia"
The administrative frontend
Filtering
Work with tags
Search
GeoAPI
Interactive plans of the museums

Fig. 2. The architecture of "Open Karelia"

The central element of each frontend is "Object card" - web-page with detailed description of specific object. Object tags statistic, related multimedia files, histories and object geodata are also presented on this page. Users can access "Object card" by two ways - using QR-code generated by the system and by direct transition from other frontend pages. Any access to user frontends of "Open Karelia" does not require any registration and authorization.

III. Statement of the problem

The problem stated behind this work is the following - creation of the universal approach for building recommendations using only the data about object itself. The solution must satisfy next requirements:

1) Interface of recommendation mechanism should be flexible in terms of proximity criterions for objects.

2) Calculation of the recommended objects should be done with taking to account individual tag sets and big text fields such as "Name", "Full description", "Annotation".

IV. Data source for building recommendations

The important question for recommendation system is a data source for recommendation building process. According to literature [4, 7] there are four the most frequently used sources:

- information about user content ratings added;
- information about user;
- information about content;
- combination of information about user and content.

Certainly sources 1, 2 and 4 allow achieving big value of personalization. This fact appear because of explicit (by information about user) or implicit (by content ratings) usage of person interests. However such sources can not be used in information system "Open Karelia" due to user's anonymity. That's why content information was chosen as a basic source for recommendation building approach.

V. Solution

A. Used approach

We use approach based on points of proximity for building recommendations in the "Open Karelia" system. Let us define points of proximity as a non negative number, which characterize similarity of two objects using the specific proximity criterion. Points of proximity value of object A and object B using criterion C vary directly with objects similarity. Let us define proximity criterion as a set, consisting of selection criterion and weighted set of fields (fields list). The fields list is used for direct compare of two objects. Selection criterion is a restriction for object B fields. If selection criterion is not satisfied then points of proximity value for A and B objects using proximity criterion C is assumed to be equal zero.

Let us discuss procedure of proximity point's calculation. We define this value as D. Procedure steps are following:

3) Selection criterion check. If object B does not match the criterion then D is set to zero and procedure is finished.

4) Similarity calculation for each field in field list of the criterion C in objects A and B. The value of similarity metric multiplied with field weight in field list is added to value of D. The procedure of metric calculation for different field types is described below.

5) If both objects A and B are included at the same time in one or more histories then value of D is incremented by value of d multiplied by number of such histories.

Advantages of the described approach are the next:

- flexibility - by manipulating selection criterions, composition and weights of the field list proximity criterions for different object parameters can be built;

- universality - different fields are considered uniformly in recommending process.

B. Types of recommendations

So as to take into account various interests of different users several criteria were offered, recommendations about which are demonstrated in frontend at once.

Table I shows proximity criterion names used in “Open Karelia”. Lists of fields aren't provided in view of their not informative content and large volume.

TABLE I. NAMES OF THE PROXIMITY CRITERIA USED IN SYSTEM AND THEIR SELECTION CRITERION

Name	Selection criterion
Objects from other museums	The field "Museum" in A and B don't coincide.
Objects with similar tags	Sets of tags A and B have nonempty crossing.
Objects with similar date	Year is calculated from the field "Date" objects A and B is the same.
Objects with a similar location	The field "Museum" in A and B don't coincide.
Similar objects from immovable heritage	The field "Class" of object B is "built heritage".
Objects with other class	The field "Class" in A and B don't coincide.
Recommendations	The field "Class" in A and B is the same.
Name	Selection criterion
Objects from other museums	The field "Museum" in A and B don't coincide.
Objects with similar tags	Sets of tags A and B have nonempty crossing.

C. Calculation of proximity points for various fields

In this chapter similarity degree calculation for different field types are described. Let us consider that only the following fields are used in criteria of proximity: name, tags, distance, class, heading, location, author, show-window, description author, place, type, equipment, material, an arrangement on the plan. The given fields were selected because they provide comprehensive characteristics of object, using minimum of information.

As proximity criteria for the field "Name" we choose function which is inversely proportional to the value of Levenstein distance [9] between values of this field for the compared objects:

$$n_{dist}(n_1, n_2) = \frac{1}{(lev(n_1, n_2) + 1)}, \qquad (1)$$

where n_1 and n_2- value of the field "Name" for the first and second compared objects, $lev(n_1, n_2)$ - Levenstein distance between two lines.

We choose the “Name” field criterion in described way because using such function similarity of objects names is proportional with proximity points value. The greatest value of function n_{dist} is reached when name strings are the same and the Levenstein distance equals to 0. In case of growing difference between two names the value of n_{dist} asymptotically aspires to zero that corresponds to completely various objects in the sense of proximity points.

For "Tags" field we defined proximity criteria which equal the intersection power of tag sets of the compared objects:

$$t_{dist}(t_1, t_2) = |t_1 \cap t_2| \qquad (2)$$

The formula (2) allows to understand tag sets similarity as a proximity points – big number of common tags will lead to big value of proximity points. An obvious drawback of this formalization is unification of various tags – because the system of tag mapping does not take tags semantic into account so some meaningless tags will be treated the same as important ones.

For the field "Date" calculation of quantitative degree of similarity is carried out with use of the quantitative difference of values of this field at both objects designated through *D*:

$$d_{dist}(d_1, d_2) = \frac{1}{D + 1} \qquad (3)$$

Size *D* is determined depending on the available information on dates for the compared objects by the following procedure;

- if both objects have information on year (century) in the field "Date", D is found as a difference of years (centuries);
- if at one object does not contain year information about year in the "Date" field, but there is information about century while another has both values, the difference of D is calculated as a difference of centuries for objects.

For other object fields as the quantity of similarity exact comparison of values is used – if values equals, the value of proximity points equals to 1, in other cases proximity point’s value equals to 0. This approach was chosen for two reasons. First, exact comparison of fields is significantly simpler in realization, than approaches described above on the basis of set-theoretic operations or Levenstein's distance. Secondly, sets of values of all other fields (except the fields "Name", "Tags", "Date") are finite and have low cardinality.

D. Realization of the recommendations creation mechanism

The recommendations mechanism was realized as a program module for information system "Open Karelia". Work with recommendations within the module is divided into two stages:

1) Creation a recommendations cashe. At each start of "Open Karelia" backend there is a consecutive calculation of proximity points for all couples of objects and by all proximity criterions. The calculated values are indexed by set of the ordered couple objects, couple identifiers and the proximity criterion identifier.

2) Granting the program interface of obtaining recommendations about the identifier of object and proximity criterion. Recommended objects set of size N is carried out by N objects choice which correspond N greatest values of proximity points at the set proximity criterion.

We choose preliminary caching due to performance issues because proximity points calculation in the course of the user appeal to system can lead to an additional delay of answer. However, such approach is associated with significant limitations:

1) The volume of memory required is directly proportional to the number of objects in the database.

2) Restart of a web application is necessary for data updating in the cache.

3) At start of a web application recommendations service is inaccessible for the cache creation period.

VI. Conclusion

The developed recommendations creation system for information system "Open Karelia" represents the decision based on proximity points concept. This decision is flexible - only the selection criterion, the set of fields and scales on which calculation of proximity points will be conducted is necessary for creation of new recommendations type. Thus the system considers fields of various structure - having the fixed values amount, full text and tags that allows to find interrelations between objects, various by the nature.

Further work on recommendations system will include:

- research of applicability and expediency of various proximity criteria;
- development of system expansions for comparison of full text fields of large volume ("brief description", "full description");
- realization of the offline-caching mechanism - creation of a background procedure of recommendations cache, with periodic data updating.

References

[1] Personalized Recommendations Frequently Asked Questions, Web:http://www.imdb.com/help/show_leaf?personalrecommendations.

[2] Amazon.com Help: About Recommendations, Web: http://www.amazon.com/gp/help/customer/display.html/ref=hp_left_sib?ie=UTF8&nodeId=16465251.

[3] YouTube, Recommended videos for you, Web: https://www.youtube.com/feed/recommended.

[4] Euregio Karelia: Museum Hypertext, Web: http://openkarelia.org/about.

[5] Rokach, Lior, Bracha Shapira, and Paul B. Kantor, *Recommender systems handbook*, New York:Springer, vol.1., 2011.

[6] mongoDB , Web: http://www.mongodb.org/.

[7] Melville, Prem, Raymond J. Mooney, and Ramadass Nagarajan. "Content-boosted collaborative filtering for improved recommendations." *In AAAI/IAAI*, 2002, pp. 187-192.

[8] V. Romanikhin and M. Zaslavsky, “Spacial Filters For Geo2tag LBS Platform”, *Proceedings of 11th Conference of Open Innovations Association FRUCT*, St-Petersburg, Russia, Publisher: SUAI, 23-27 April 2012, pp 151-157.

[9] Levenshtein, V., “Binary codes with correction of loses, inserts and replacement of symbols” [Dvoichnyye kody s ispravleniyem vypadeniy, vstavok i zameshcheniy simvolov], *Reports of Sciences Academies of the USSR*, 1965, pp 845-848.