Experimenting with facilitating collaborative travel recommendations

Arkadiusz Załuski, Maria Ganzha
Warsaw University of Technology
Warsaw, Poland
Email: A.Zaluski,Maria.Ganzha@mini.pw.edu.pl

Marcin Paprzycki
Systems Research Institute Polish Academy of Sciences,
Warsaw, Poland
Email: Marcin.Paprzycki@isbpan.waw.pl

Costin Bădică, Amelia Bădică
University of Craiova,
Craiova, Romania
Email: cbadica@software.ucv.ro, ameliabd@yahoo.com

Mirjana Ivanović
University of Novi Sad,
Novi Sad, Serbia
Email: mira@dmi.uns.ac.rs

Stefka Fidanova, Ivan Lirkov
Bulgarian Academy of Sciences,
Sofia, Bulgaria
Email: stefka,ivan@parallel.bas.bg

***Abstract*—The aim of undertaken work was to experiment with, and compare, various approaches to facilitate a collaborative recommender system, for travelers who would like to visit "tourist places". For this purpose, different algorithms, including Kohonen-type neural networks (plain and elastic), as well as semantic technologies, have been applied to a dataset collected form the Web. Experiments have been performed for groups of users (measuring quality of recommendations), as well as for selected individuals. Completed comparison points out to main strength and weakness of each approach.**

***Index Terms*—Recommender system, collaborative filtering, Kohonen self-organizing maps, semantic technologies**

I. Introduction

Recent years have been characterized by a stream of research aiming at providing "users" with recommendations. Sample existing recommender systems (found in very-well-known online services) deal with, among others, films (Netflix), music (Spotify, YouTube), or shopping (Amazon). There have also been multiple attempts at delivering "travel recommendations", with various level of success[1]. This being the case, the aim of this note is to "go back to the basics" and study performance characteristics of various methods than can be applied to instantiate a collaborative recommender system. Here, let us note that imposed space limitations considerably restrict our ability to facilitate introductory and background material. Therefore, we assume that the reader is familiar with: ontology, semantic technologies, RDS, RDFS, OWL, Semantic Web, recommender systems, collaborative filtering, user similarity, cosine distance, Pearson correlation, modified Pearson correlation (with weighted average of object scores), nearest neighbors algorithm, defining user profiles, clustering, K-means clustering, Kohonen neural networks (self-organizing maps; SOM), elastic Kohonen neural networks (ESOM), as well as fundamentals of measuring similarity between entities. Readers requiring additional knowledge in these areas should consult pertinent resources.

[1]See, for instance http://www.ibspan.waw.pl/~paprzyck/mp/cvr/research/agents_TSS.html and references collected in papers listed there

Taking into account this limitation, we proceed as follows. We start, in Section II with description of the data sources, their preparation and initial processing. We follow with core experimental results and their analysis. Summary of contributions and potential research directions summarize the paper.

II. Data sources and their preprocessing

Our goal is to investigate various approaches to delivery of recommendations concerning "travel entities", such as, for instance, restaurants, bars, pubs, hotels, etc. We focus our attention on "collaborative filtering", i.e. recommendations based on similarity between "interests" of a given user and other users that are similar to her/him and to each other.

As the data input, to be used in the developed "system", we have used three data sources: OpenStreetMap, LinkGeoData and Yelp. Let us now briefly describe each one of them.

A. OpenStreetMaps

OpenStreetMaps[2] is a publicly created website, developed by an online community from around the world. Number of registered users is over 2 million. It contains multiple "points of public utility", added directly by the users. These are, among others:

- gastronomy-related points: restaurants, bars, pubs;
- accommodations: hotels, motels, guest houses;
- tourist attractions: museums, churches, monuments;
- entertainment spots: cinemas, theaters, clubs.

The OpenStreetMap database is available as an XML file. It is possible to limit searches to specific continents, countries or cities. In our work, the geofabrik[3] service, which preprocesses OpenStreetMap maps information, was used. The following listing presents information (collected by the OpenStreetMap community) for a "random restaurant from Montreal, CA".

```
<node id ="3426993381" version="1"
      timestamp="1970-01-01T00:00:00Z">
```

[2]https://www.openstreetmap.org
[3]https://www.geofabrik.de/

```
<tag k="name"
        v="Restaurant Le Muscadin"/>
<tag k="phone" v="514-842-0588"/>
<tag k="amenity" v="restaurant"/>
<tag k="cuisine" v="italian"/>
<tag k="smoking" v="outside"/>
<tag k="website"
        v="http://www.lemuscadin.ca"/>
<tag k="capacity" v="100"/>
<tag k="addr:city" v="Montreal"/>
<tag k="wheelchair" v="yes"/>
<tag k="addr:street"
        v="Rue Notre-DameOuest"/>
<tag k="addr:postcode" v="H3C1H8"/>
<tag k="addr:province" v="Quebec"/>
<tag k="opening_hours"
        v="MON-FRI: 11am-10pm
        SAT: 5pm-10pm"/>
<tag k="addr:housenumber" v="639"/>
</node>
```

Here, one can find the exact geographical location, date of listing creation, unique ID, and a set of "restaurant features". Each attribute has a key (k) and a value (v) to describe it. Within OpenStreetMap, there is also a collection of recommended and commonly used keys for description of specific places. For example, for restaurants, popular keys are cuisine, smoking space availability, or a phone number. In case of hotels, such keys are, for instance, number of rooms, available amenities, or the number of stars.

B. LinkedGeoData

The LinkedGeoData[4] aims at adding spatial dimension to the Semantic Web[5]. It has over 3 billion nodes, and 20 billion RDF triples. Additionally, data from LinkedGeoData is associated with the DBPedia[6] and the GeoNames[7] database. LinkedGeoData uses relational PostGIS[8] database for storing data from the OpenStreetMap site. Stored data is extended with tables and indexes, capturing relationships between records.

C. Yelp

The OpenStreetMap data does not include "user assessments". Therefore, we have decided to incorporate information available from Yelp[9]. Yelp is one of largest websites that enable users to write reviews about travel sites. It has over 77 million ratings, 2 million locations, and 142 million unique users per month. Yelp website provides APIs that allow access to data. However, limitations are imposed on the number of daily queries, and lack of ability to search all ratings provided by a specific user. In a recommender system, the latter functionality is needed to be able to develop user profiles and, "in the next step", to "clusterize" them. Therefore, we have used a "truncated test version" of Yelp, with 366,000 users, and 6,300,000 ratings of 160,000 different sites. These originate from 7 cities in US and Canada: Phoenix, Las Vegas, Charlotte, Edinburgh, Pittsburgh, Montreal and Madison.

D. Combining data sources

Obviously, we had to combine information available within Yelp with that from the OpenStreetMaps. These are two independent, and unrelated, repositories that use very different data models. Upon further investigation, we have not found a simple direct conversion that would allow to search for places within OpenStreetMap and assigning to them user ratings from Yelp. Simply said, there are too many inconsistencies between information stored in both repositories (e.g. non-matching addresses for the same point of interest, or the same address for entities with somewhat different names). To deal with this situation we have implemented solution that *jointly* uses the following approaches to match available information.

- Geospatial matching. In both databases, location is demarcated using latitude and longitude. Hence, we have assumed that two entities that are similar-enough in their names (see, next) are actually the same if they are no more than 50 meters apart.
- Semantic similarity was used to establish closeness of names. Here, the Jaro-Winkler distance [10] was used to measure the edit distance of two (or more) names of interest. Entities with Jaro-Winkler similarity measure at least 0.9 were deemed to represent the same entity.

After combining and cleaning data sources (mostly, removing incomplete entries), the total number of users became 3318. They issued 51,517 recommendations, on the scale 1 to 5, where 5 means "like a lot", while 1 means "do not like at all". In Figure 1 we depict number of reviews per user.

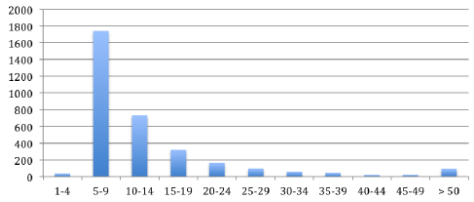

Fig. 1. Number of reviews per user.

As can be seen, making recommendations is not very popular (within our dataset). Majority, 52% of users, rated 5-9 items, while only 15% of them rated more than 20. The largest number of recommendations, issued by a single user, was 301. Users evaluated 4,353 unique locations. In Figure 2 we depict number of reviews for each property. As can be seen, approximately 25% of entities have less than 10 references.

Finally, when creating a recommender systems, the distribution of input data (which should be "relatively even") needs to be considered. In our dataset, scores 4 and 5 represent,

[4]http://linkedgeodata.org/About
[5]https://www.w3.org/standards/semanticweb/
[6]https://wiki.dbpedia.org
[7]https://www.geonames.org
[8]https://postgis.net
[9]https://www.yelp.com
[10]https://en.wikipedia.org/wiki/JaroWinkler_distance

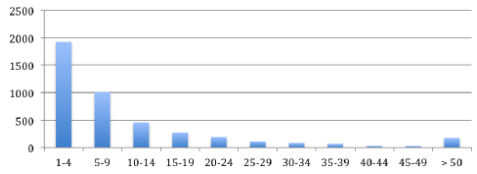

Fig. 2. Number of recommendations for individual entities.

respectively 41% and 23% of all reviews. Moreover, most negative scores – 1 and 2 – have been assigned in 4% and 9% of reviews, respectively. While this situation is not perfect (for using this data in a recommender system), it is quite common in real life (people are not keen to assign very bad grades, unless they are extremely unhappy). Therefore, we had accepted this data (keeping in mind all related limitations).

E. Preparing data for experiments

An important issue, is how to evaluate "quality" of a recommender system. An obvious way, i.e. "to ask users", requires large number of "appropriately diverse evaluators", to be methodologically correct. Hence, it is rather difficult to reliably complete this step. Instead, we have decided to use the existing dataset. Since each recommendation includes time stamp, it is possible to time-divide the dataset into two subsets. First, contains recommendations issued before a specific date (90%) and can be used for "training". Second, contains recommendations issued after the selected date (10%), can be used for testing. We are aware that this approach does not take into account preferences that "evolve over time". Moreover, since the number of reviews is relatively small, instead of the standard 80-20 split, we have decided to use the 90-10 split, which may also raise some questions. Regardless of these limitations, we believe that this approach is acceptable for comparatively assessing quality of recommendations.

Separately, since a recommender system should deliver correct recommendations for *individual users*, we have selected three the highest number of reviews (let us name them Anna – 216 reviews, Robert – 198 reviews and Julia – 301 reviews). We assume that the more recommendations the user makes the better her/his preferences are represented and understood.

III. Experimenting with recommending methods

Let us now present results of various experiments we have performed using various methods, for different recommendation scenarios.

A. Methods with memory

In our experiments we have used three methods utilizing similarity matrix, which apply: (i) cosine distances, (ii) standard Person correlation, and (iii) Pearson correlation with weighted average of object scores. We have applied them to users with "large number of reviews" (10 or more). Specifically, from the dataset we have removed all users who had less than 10 reviews, sorted all remaining reviews according to when they were posted, and divided into 90-10 subsets. Quality of recommendations has been evaluated on the basis of how close the score of the recommendation was to the actual review score. Results have been divided into three categories: (a) perfect (exact agreement between the predicted and the actual evaluation); neutral (the suggested score is at most by one off, e.g. 4 instead of 3); incorrect (suggested score is more than 1 point off, e.g. 2 instead of 4). In Figure 3 we summarize the results for the three methods, as well as random assignment.

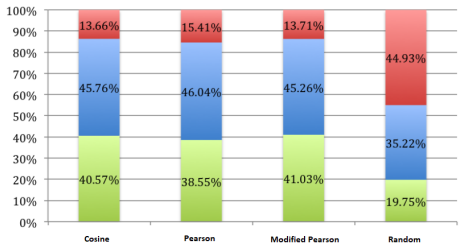

Fig. 3. Comparison between memory-based methods; perfect match – green, neutral – blue; incorrect – red.

As can be seen, all three methods give very similar scores (significantly better than the random assignment). When looking into exact numbers, the standard Pearson correlation method is slightly worse than both cosine and modified Pearson. For perfect matches, the modified Pearson is the winner. If perfect matches and neutral scores are combined, the approach based on the cosine distance gives best results.

Separately, we have applied the Pearson correlation based approach to the three users with largest number of reviews. Results have been summarized in Figure 4.

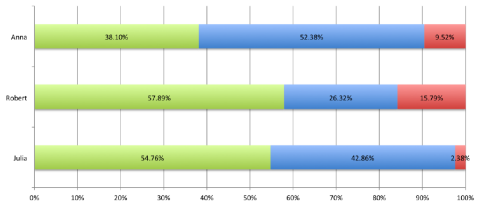

Fig. 4. Applying Pearson method to selected three users.

Here, results were better than those reported in Figure 3. For Julia, with the best results, matches occurred in $\sim 55\%$ of cases, neutral in $\sim 43\%$ (with $\sim 2\%$ of "errors"). Moreover, they are in line with those obtained for the complete dataset (all users, regardless of the number of reviews). There, for all three methods, we observed drop of $\sim 3\%$ for the match category. This drop was partially offset by $\sim 1\%$ increase of neutral scores. Overall, the message is clear: the more data about the users, the better the recommendation.

B. Methods based on user profile

While the memory-based methods work well for users who have a lot of past scores, they have problems when the number of scores is small (sometimes, no past reviews at all; so-called

cold start problem). Here, one can apply methods based on user profiles (see, [1] and references found there).

Following this path, we have introduced a 20-element user profile. Each preference can have value from the interval $(0, 1)$. Names of categories, and specific values for users, are based on information available in Yelp. For example, attitude towards smoking is defined through the number of places for smokers and their ratings (hotels, restaurants). Let us present an example of user profile:

movies: 0.87
fast food: 0.08
galleries: 0.13
party goer: 0.01
outdoor eating: 0.09
American cuisine: 0.16
Asian cuisine: 0.94
Italian cuisine: 0.38
European cuisine: 0.23
vegetarian: 0.1
Mediterranean cuisine: 0.04
International cuisine: 0.17
home eating: 0.78
expensive hotels: 0.0
cheap hotels, hostels: 0.0
monuments: 0.15
smoker: 0.87
Internet access: 0.63
traveler: 0.05
home food delivery: 0.9

Here, we see person who loves to go to the cinema and order (typically, Asian) food delivery. This person does not like traveling (posts only reviews of "places near-by"). Moreover, it is a smoker, who values Internet access. Recall, that the profile is generated directly on the basis of her/his ratings.

User profiles have been clustered using Self-Organizing Maps (SOM) and their elastic version (ESOM). Standard versions of SOM and ESOM, with typical parameters, have been used (initial and final values of the neighborhood function – 2 and 0; initial ratio of the learning function – 0.05, final ratio – 0.005) . Results were examined for users who rated less than 7 places. Again, recommendations based on SOM and ESOM were calculated and compared with the actual recommendation. Results for the same dataset were calculated also for the approach with memory based on cosine similarity measure (see, Section III-A). As previously, three measures of success have been recognized (success – green, neutral – blue and failure – red). In Figure 5 we depict obtained results.

It is easy to see that, when number of rankings is small, user-profile-based methods lead to better results. Here, note that scores obtained using cosine distance, for users with less than 7 ratings, have far worse quality than for users with more than 10 reviews (see, Figure 3). Match occurs in only 29.22% cases (decrease of 11.35%). The SOM algorithm obtained (for users with less than 7 reviews) results similar to those of the Pearson measure approach (for users with more than 10 reviews). The

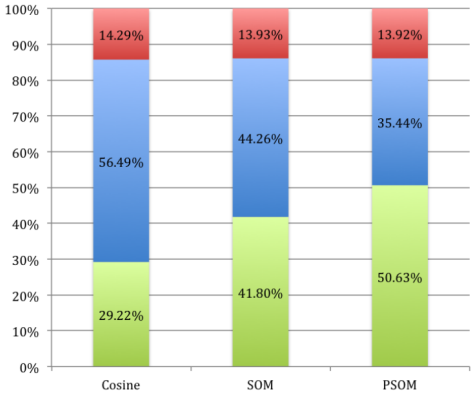

Fig. 5. Predicting review scores using SOM and ESOM, and comparing them with cosine measure in memory based approach.

ESOM algorithm was definitely the best, as it provided 50.63% of matching recommendations.

We have also applied ESOM to the three users with most recommendations. In Figure 6 we present the results.

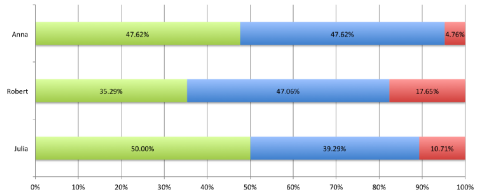

Fig. 6. Applying ESOM to three users with most reviews.

Interestingly, the results are quite different than in memory-based approaches. When comparing Figure 4 and Figure 6, when match (green) category is concerned, ESOM "wins" for Anna, but "looses" for Robert and Julia. For the neutral (blue) category, ESOM wins for Robert. Combining match and neutral, ESOM is the best for Anna.

C. Item-based approach – semantic similarity

Item-based methods, instead of similarity of users, are based on similarity of objects. As an example of such approach we have used the semantic similarity. To deal with semantic similarity the ontology of tourist places has been defined (its fragment can be found in Figure 7).

It is important to stress that every element (node) of this ontology has specific property, which describes some specific information about a given place, e.g. to describe a restaurant we need to know its name, opening hours, type of cuisine, and so on (see Figure 8):

Since it is not obvious, let us briefly discuss how one can calculate "semantic similarity" of tourist objects (see, also [3]). Taking into account that objects, in the ontology, are nodes of graphs, which can be represented as RDF triples, the recommender system may calculate semantic similarity using:

1) comparison of properties of objects, and
2) numerical representation of hierarchical dependences of different nodes in the ontology.

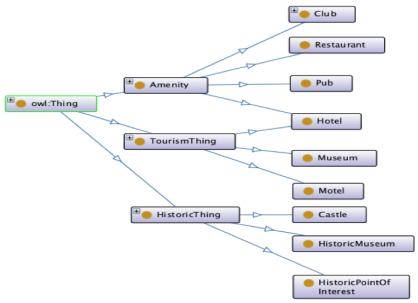

Fig. 7. Ontology of tourist places

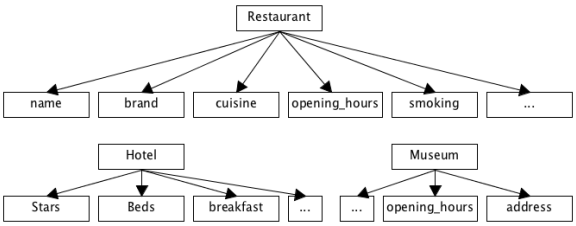

Fig. 8. Example of tourist object properties

When comparing attributes describing concepts, one should realize that different characteristics are not always equally important. For example, for two restaurants, information about their location and cuisine is much more important than their phone numbers. In recommendation systems, based on semantic similarity, different weights can (and should) be assigned to individual attributes. Obviously, the more important the attribute, the larger the assigned weight.

In case of "hierarchical comparison", similarity of two objects c_1 and c_2 can be calculated using equations 1 and 2:

$$d_c(c_1, c_2) = d_c(c_1, ccp) + d_c(c_2, ccp) \quad (1)$$

$$d_c(c_1, ccp) = \big(1 - \frac{depth(cpp)}{depth(N)}\big) - \big(1 - \frac{depth(c_1)}{depth(N)}\big) \quad (2)$$

where: cpp is the closest common ancestor of nodes c_1 and c_2, while $depth(c)$ is the distance between node and tree root.

Here, for the property-based approach to comparison we distinguish two possible situations:

1) Compared attributes are numeric. Then, similarity of numeric attributes α_1 and α_2 is calculated as:

$$p(\alpha_1, \alpha_2) = 1 - \frac{|val(\alpha_1) - val(\alpha_2)|}{\max(\alpha)} \quad (3)$$

where $val(\alpha)$ is value of attribute α and $\max(\alpha)$ is maximal value of all attributes.

2) Compared attributes are a sign sequence. Here, similarity of attributes α_1 and α_2 is calculated based on WordNet[11]. This allows to take into account synonyms or expressions with close meaning (e.g. *good* and *right*).

[11]https://wordnet.princeton.edu

Recall that different aspects of a given tourist location have different importance for establishing semantic similarity. For example, outdoor seating, or information about smoking area, are definitely more important (for the similarity) than the seating capacity. Therefore, the algorithm compares not only individual properties, but also used weight assigned to them. Specifically, we have introduced the following weights:

```
type of place (restaurant, pub, ...): 2.0
cuisine (French, Italian, ...): 4.0
city: 2.0
facilities for disabled customers: 4.0
Internet access: 4.0
places for smokers: 4.0
opening hours: 1.0
operator (Subway, Burger King): 4.0
possibility to order take away: 4.0
beer garden: 3.0
drive through: 3.0
home delivery: 3.0
availability of alcohol: 4.0
draft beers: 3.0
parking spaces: 1.0
dishes (hamburger, pizza, sushi, ...): 3.0
vegetarian dishes: 4.0
vegan dishes: 4.0
rating: 3.0
```

These features are based on information available in Yelp. However, the specific weights are our choice and should be treated only as "parameters".

Our initial experiments with the "tourist places" ontology of travel objects (described above) indicated that its performance is not satisfactory. Therefore, we have decided to extend our initial ontology with that of cuisine and dishes (see, Figure 9). This, in turn, forced us to restrict the domain of experiments to "food places". In our data, we have located 3290 such points (and this number of entities was used in the experiments reported in the remaining parts of this contribution). In Figure 10 we compare the performance of the approach based only on semantics of travel places with the one utilizing an extended "food places ontology".

First, let us note that results presented in Figure 10 cannot be compared with the earlier ones. This is because we are dealing with a different number of objects (and recommendations) in the dataset. Second, the improvement of using extended ontology is $\sim 1\%$ for the match and $\sim 2\%$ for the neutral category (for a total of $\sim 3.2\%$). This improvement indicates that the more robust the domain ontology the better the chance of properly using it in the recommender system.

We have also applied the semantic approach to the three selected individuals. The best results were obtained for Julia ($\sim 61\%$ match, $\sim 33\%$ neutral and $\sim 5\%$ fail), Robert had second largest percent of matches ($\sim 47\%$) but $\sim 16\%$ fail, whereas Anna had only $\sim 43\%$ match, but more than 52% neutral, which left her with as many failures as Julia ($\sim 5\%$).

IV. Concluding Remarks

The aim of our work was to compare performance of collaborative recommender systems based on different approaches. For the series of experiments, reported here, we

Fig. 9. Cuisine and dishes ontology

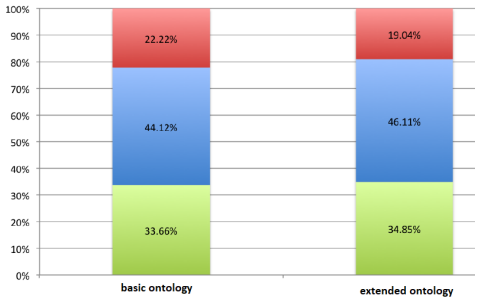

Fig. 10. Recommendation improvement due to ontology extension

have used subset of data available in Yelp. We have applied three different classes of approaches: (1) memory based, (2) user profile-based, and (3) item-based. In the latter case we have applied semantic similarity, based on travel domain mini-ontologies. Let us note that, results reported here, are a "core subset" of our experiments. Moreover, validity of our findings is mitigated by: (A) overall, relatively small size of the dataset, (B) non-standard division of data between training and testing sets, and (C) "verification" based on "temporal split of data". Nevertheless, we are convinced that the following general observations hold, representing our core contributions.

1) All approaches that we have experimented with, when "averaged" over a set of users, produce similar results.
2) The more "we know" about the user(s), the better the recommendations we can deliver.
3) While the averaged results are similar, results obtained for individuals vary considerably. Moreover, no decipherable pattern of behaviors has been observed. Therefore, we cannot offer definite explanations of reasons why given method (X) is better or worse than method Y, in the case of specific user U_i.
4) Improvement related to making domain ontology more comprehensive may indicate that it would be worthy to extend travel ontology and expect improved quality of recommendations. However, power of this approach is somewhat limited. While it is possible to create very detailed domain ontologies, it is not clear that there exists data that would allow to fully describe its individuals. Recall that results are based on concepts available within Yelp.
5) Existence of large number of results that have been dubbed as "neutral" (they represent recommendations that are close, but not exact), brings about possibility of introducing the "serendipity effect" to the recommender system. In other words, system can recommend objects that are "close" but "not exact" – thus introducing "novel ideas" to users.
6) Finally, taking into account that different methods produce different recommendations (in particular in case of individuals), there is a place for "hybrid methods". Such methods could apply different recommender algorithms (e.g. one from each category) and combine results using, for instance, approaches reported in [2].

Based on these points, we believe that the most promising future research directions are: (i) revisit the above approaches for much larger dataset (apparently Yelp has, recently, opened a much larger test dataset), and (ii) investigate viability of hybrid approaches.

ACKNOWLEDGMENT

Work presented in this paper was supported in part by: PAS-BAS bilateral project "Practical aspects of scientific computing", PAS-RAS bilateral project "Semantic foundation of the Internet of Things", as well as a collaboration agreement between University of Novi Sad, University of Craiova, SRI PAS and Warsaw University of Technology.

REFERENCES

[1] Maciej Gawinecki and Mateusz Kruszyk and Marcin Paprzycki (2005) *Ontology-based Stereotyping in a Travel Support System.* In: Proceedings of the XXI Fall Meeting of Polish Information Processing Society, PTI Press, 73-85; http://www.ibspan.waw.pl/~paprzyck/mp/cvr/research/agent_papers/PIPS_2005_MMP.pdf

[2] Maria Ganzha and Marcin Paprzycki and Jakub Stadnik, *Combining information from multiple search engines – preliminary comparison.* In: Information Sciences, 180(10), 2010, 1908-1923.

[3] Pawel Szmeja, Maria Ganzha, Marcin Paprzycki, Wieslaw Pawlowski, *Dimensions of semantic similarity.* In: Gaweda A. et.al. (eds.) Advances in Data Analysis with Computational Intelligence Methods, Studies in Computational Intelligence, vol. 738, Springer, Berlin, 87-125.