Content-based Recommendations Using Similarity Distance Measures with Application in the Tourism Domain

Ioannis Panagiotis Ziogas
Technical University of Crete
Chania, Greece
ziogas@isc.tuc.gr

Harris Papadakis
Hellenic Mediterranean University
Heraklion, Greece
adanar@hmu.gr

Errikos Streviniotis
Technical University of Crete
Chania, Greece
estreviniotis@intelligence.tuc.gr

Georgios Chalkiadakis
Technical University of Crete
Chania, Greece
gehalk@intelligence.tuc.gr

ABSTRACT

In this paper, we explore the use of similarity distance measures for Content-based recommendations for touristic attractions. First, we study ways of deploying hierarchies of points of interests (POIs) and operate upon them with well-known similarity distance measures originating in the text analysis domain. Then, we progressively build three novel, hierarchy-free, similarity measures, and discuss their strengths and weaknesses. We end up with a measure, the Weighted Extended Jaccard Similarity (WEJS) that combines information regarding the user interests (in the form of user preference-related weights applied on the items’ features) and specific items’ characteristics (in the form of particular values for the items’ features). As such, the use of WEJS allows the provision of recommendations that are effectively personalized. Interestingly, though it is a hierarchy-free measure, it is able to recommend items based on others that would naturally appear close in a features-based POIs hierarchy; while at the same time it is able to capture similarities among items that would be distant to each other in any hierarchy built solely based on the POIs' features. Our systematic experimental evaluation on a real-world dataset showcases the benefits and limitations of the various measures, and confirms the effectiveness of WEJS in offering “rich” and personalized recommendations.

KEYWORDS

Recommender Systems, Content-based, Hierarchies, Distance measures

1 INTRODUCTION

Recommender Systems (RSs) are techniques able to implement suggestions to a user for a type of item or items [15]. Various decision making processes are related to suggestions produced by a recommender, e.g., what items to buy, which movies to watch, which tourist destinations to visit, and so on. In general, RSs are designed to assist people in evaluating huge amounts of alternative items of interest. Many media companies or online retailers employ such systems as parts of their services, in order to maximize profits via increasing customer satisfaction. In general, such systems can be categorized as Content-based, Collaborative filtering or Hybrid, based on the techniques that they adopt [15].

At the same time, in our era the tourist industry is closely linked to the use of generic and specialized RSs. Such systems operate as a digital guide for the different activities that a tourist destination can provide to individuals according to their preferences. The tourism related available information on the Internet is endless, and several applications of travel RSs provide assistance in organizing a tour and enhance the experience of visiting a tourist destination. Chaudhari et. al [4] focus on travel or tourism RS and categorize them based on the type of recommendations they provide. The categories that surveyed them are hotel RSs, restaurant RSs, tourism RSs associated with group recommendations, tour planning or travel packages; and tourist attraction RSs (i.e., points of interests, museums, etc.). According to the authors, Content-based RSs are mostly found in restaurant RSs, while in the other categories are employed alongside Collaborative filtering RSs as hybrids.

Now, the semantic identification of the semantic distance is considered as a research subject of data processing, Intelligence and Linguistics [18]. Such measures are usually being deployed in Text Analysis [10] or Natural Language Processing [12]. Different types of applications can also take advantage of these measures. In the area of RSs, those measures are highly connected with the context that is represented by the user’s preferences. The semantic similarity measures usually present the ontological relations between words in cases of similarity identification of documents. As such, we consider that semantic similarity measures are categorized as: (i) measures for concepts that are arranged in hierarchical manner [18, 21]; and (ii) measures capable of calculating the similarity in the absence of a hierarchical structure [7].

In this work, we introduce a novel approach that combines Content-based RSs for the tourism domain with some well-known
semantic similarity measures; and test it on a real-world dataset built for the needs of a real-world tour planning recommender system currently under development for the tourist destination of Agios Nikolaos, Crete, Greece. We were keen to explore the possibility of deploying such metrics in our system, since to the best of our knowledge their use for non-collaborative filtering RS techniques is limited, and practically non-existent for tourist RSs [4]. In more detail, real world data on points of interests (POIs) and users’ preferences were collected by: (i) local knowledge, (ii) online sources, and (iii) questionnaires that were filled by tourists. Furthermore, the items, i.e., points of interests (POIs), were arranged in a hierarchical structure in order to deploy some well-known hierarchy distance measures. On the other hand, we systematically develop three novel distance similarity measures, which function in absence of a hierarchy structure. Among these, we champion the use of the Weighted Extended Jaccard Similarity (WEJS) metric, a sophisticated measure that is capable of capturing the similarity between POIs by importing feature values and weights that are influenced by users’ preferences. Thus, WEJS exploits users’ personal interests, and therefore provides effectively personalized recommendations. Finally, we conduct a systematic experimental evaluation of our measures on our real-world dataset for the Agios Nikolaos area. Our evaluation confirms that WEJS, in particular, is able to capture the similarity of both (i) POIs that would naturally appear close in a hierarchy; and (ii) POIs that would be distant to each other in any hierarchy built solely based on their features, but are still similar in terms of actual content that matches the user’s preferences. As such, WEJS can be used to offer recommendations that are quite “rich”—i.e., diverse enough but still relevant to user input and preferences.

2 BACKGROUND AND RELATED WORK

Here we provide some background on semantic similarity measures and recommender systems; and brief-review related work.

2.1 Semantic Similarity Measures

The semantic similarity, or semantic distance, between two concepts, given an ontology, measures the similarity of the concepts based on their common characteristics [19]. As such, it is mainly used in order to present the ontological relationships between words in cases of similarity identification between documents. The structure definition of the compared concepts, which are also related, is formally called IS-A hierarchy. In general, the most developed semantic similarity measures are those applied to the taxonomic or ontological representation of the context [6]. Various types of hierarchical data are represented by tree structures such as documents, genealogies, catalogs, language corpus etc.

In more detail, given an ontology, similarity between objects can be established either as the distance between individual concepts or as the distance between sets of concepts. The semantic similarity between two (individual) concepts can be determined by: (i) the edge-based approach and (ii) the node-based approach. Specifically, edge-based approach relies on applying graph distance measures to a hierarchy tree considered as a directed acyclic graph (DAG); while node-based approach relies on comparing properties of the concepts involved, such as the concepts themselves, their descendants or their ancestors. For instance, [3] introduced an algorithm that calculates the distance between two concepts in a hierarchy by the hierarchy level of their nearest common parent node. Information Content (IC) is used in node-based approaches and provides how informative a node is. An information content-based approach was introduced in [14] by Resnik and computes the relatedness between the concepts as a function of their information content, given by their probability of occurrence in a corpus [20]. On the other hand, the semantic similarity between two sets of concepts is mainly used in computing similarity for information retrieval purposes. Jaccard similarity [7] is an example of a similarity metric when the sets are represented as vectors in a hierarchical structure. Now we present some well-known semantic similarity measures in the literature.

Path length similarity (PLS) is defined as the length between two concepts which connects them through their Least Common Ancestor (or Subsumer) (LCA) [13]. We used the term path loosely because of the hierarchy tree being in fact a DAG. Specifically:

$$PLS(X, Y) = N_1 + N_2 \text{ with } X \neq Y$$ (1)

where $N_1$ is the number of arcs between concept $X$ and the LCA with concept $Y$. $N_2$ is the number of arcs between concept $Y$ and the LCA respectively. Note that the path length similarity between identical concepts will be equal to zero.

Z. Wu and M. Palmer [21] proposed a similarity metric, known as Wu-Palmer similarity (WP), which describes the semantic similarity among concepts $X$ and $Y$ ($X \neq Y$) as:

$$WP(X, Y) = \frac{2N}{R_1 + R_2}$$ (2)

where $R_1$, $R_2$ is the number of arcs between concepts $X$, $Y$ and the root node respectively. $N$ is the distance of the LCA for the concepts $X$, $Y$ from the root node. A similarity distance for two compared objects or concepts of an ontology with a value which is close to 1, indicates a high similarity of these objects. On the contrary, not similar objects (i.e., distant in the hierarchy) have Wu-Palmer similarity near to zero.

![Figure 1: Example of a concept hierarchy for PLS](image)

Manjula et al. [18] developed a new similarity measure for concepts in the same hierarchy by taking into consideration the WP measure. Their inspiration is based on the simplicity and good performance WP similarity can offer. Furthermore, they combined the depth of the whole taxonomy and the shortest path between two concepts in order to include how far two concepts are semantically. MSA similarity, as we term it, is defined as:

$$MSA(X, Y) = \frac{2N \cdot e^{-\frac{D}{2R}}}{R_1 + R_2}$$ (3)
where $L$ is the shortest path between concepts $X$, $Y$ as follows, i.e. for every edge passed in the vertical direction a weight of 1 is given and when the direction is changed, a weight of 1 more is given. $D$ is the depth of the whole ontology tree. $R_1$, $R_2$ is the number of arcs between concepts $X$, $Y$ and the root node respectively and $N$ is the distance of the LCA for the concepts $X$, $Y$ from the root node. Finally, $\lambda$ is 1 for distant neighborhood concepts (i.e., concepts from a different hierarchy) and 0 for concepts from the same hierarchy (i.e., concepts coming from the same close ancestor).

The Jaccard index, also known as the Jaccard similarity ($JS$), is a statistic used for evaluating the similarity or diversity of finite sets and it is categorized as a semantic similarity measure for sets of concepts. Formally, the $JS$ of sets $A$ and $B$ can be computed as the size of intersection divided by the size of the union of two sets:

$$JS(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

- $|A \cap B|$: gives the number of members shared between both sets.
- $|A \cup B|$: gives the total number of members in both sets (shared and unshared).

The Jaccard similarity will be 0 if the two sets do not share any values and 1 if the two sets are identical.

### 2.2 Recommender Systems

In general, RSs can be categorized as non-personalized and personalized [8, 15]. Personalized RSs infer users’ preferences using their history in order to provide efficient recommendations [1]. By contrast, non-personalized RSs recommend what is popular and relevant to all the users, e.g., a list of top-N items for every new user.

Moreover, most established RSs can be classified as Content-based, Collaborative filtering and Hybrid (i.e., combinations of the aforementioned methods) recommenders. Content-based RSs provide effective recommendations by exploiting information that is available from previous user-system interactions. Specifically, such systems recommend users items that are similar to other items that the user has liked in the past. On the other hand, Collaborative filtering approaches calculate the similarity between users based on their rating history. Intuitively, Collaborative filtering techniques are based on the assumption that if two users have rated some items similarly, then they share similar interests and as such they will probably rate other items similarly as well [15].

### 2.3 Related Work

In terms of related work, there is a plethora of tourism- or travel-oriented RSs, potentially classified in different categories, as listed in [4]. Most of those systems recommend POIs that correspond to tourist attractions (e.g., restaurants, hotels, historical sites or museums), that are ideally highly connected with each individual tourist’s preferences. Here we brief-review a few representative such systems, noting that they do not in general make use of POI hierarchies or semantic similarity distance measures, as we do in our work in this paper.

To begin, a recommender for hotels is proposed by O’Mahony and Smyth [11] with the purpose of suggesting the review helpfulness of specific hotel services. Their method analyzes the hotel reviews in four features, and employs three different classifiers for reviews containing at least five opinions by users that reviewed at least one time hotels of Las Vegas or Chicago in the United States. Zeng et al. [22] proposed a restaurant RS in mobile environment that calculates the similarity between user and restaurant by employing a user preference model. The system was evaluated using Baidu map cloud services and GPS and the top-N restaurants were suggested based on the calculated similarity scores. A personalized tour planner, called eCOMPASS, was proposed by Gavalas et. al. [5]. In that work, the system recommended tours, that contained POIs, by visualizing them in a list view or a map; and was able to assist travelers through the public transportation by taking into consideration the weather forecast. The system was evaluated in the areas of Berlin, Germany and Athens, Greece.

The work that is most relevant to ours, in the sense that it employs hierarchies of items-to-recommend and semantic similarity measures—though it is not a tourist or travel-oriented RS—is the one of [2]. Specifically, it introduces a museum RS for mobile devices, which combines semantic similarity measures (Content-based method) and a semantically enhanced Collaborative filtering method to propose interesting exhibits to museum visitors. A personalized museum tour is generated through a contextual post filtering based on some variables such as the location of the visitor and the physical environment. In contrast to our approach, however, this work utilizes Jaccard and Wu-Palmer similarity measures only; specializes in museum collections only, since it is heavily based on the use of standard cultural heritage and artworks-specific ontologies; and does not provide any kind of evaluation for its approach.

### 3 OUR APPROACH

In this section we discuss how semantic similarity measures can be employed to calculate the similarity between POIs instead of concepts. First, we construct a hierarchy tree that contains POIs in its leaves and apply several well-known hierarchy distance measures. Finally, we propose three different variations of Jaccard similarity that do not require the existence of any hierarchy structure and provide several examples that describe their functionality.
3.1 Hierarchy Distance Measures

Our method distributes the points of interests, i.e., the places or activities that a destination can provide, among the leaves of a specialized hierarchical structure. The design of the POIs’ hierarchy for a tourist destination plays a significant role in our approach. All the possible POIs must be examined extensively and should be added as leaves in the hierarchy tree based on the type of the amusement they offer. For instance, a monastery belongs to the cultural POIs or a wine bar comes under the leisure POIs and more specifically under the bar node. An example of POIs’ hierarchy tree is depicted in Figure 3 and will be used to review the several semantic similarity (or distance) measures. Let \( W_1 \) and \( W_2 \) denote two different wine bars, \( C_1 \) a classic bar and \( M_1 \) a monastery.

3.1.1 Path Length Similarity (PLS). We calculate the PLS between nodes of the POIs’ hierarchy tree depicted in Figure 3. The similarity measure will be examined between node \( W_1 \) and the other nodes:

- **Path Length Similarity:**
  - \( PLS(W_1, W_2) = 1 + 1 = 2 \)
  - \( PLS(W_1, C_1) = 2 + 2 = 4 \)
  - \( PLS(W_1, M_1) = 6 + 4 = 10 \)

Intuitively, the more similar two nodes are, the shortest will be the distance between them. In comparison to the other nodes in our example, \( W_2 \) is closer to \( W_1 \).

3.1.2 Wu-Palmer Similarity. An example applying Wu-Palmer similarity measure on the hierarchy tree of Figure 3 is presented, demonstrating that \( W_1 \) is more similar to \( W_2 \) than \( C_1 \) or \( M_1 \).

- **Wu-Palmer similarity:**
  - \( WP(W_1, W_2) = \frac{2.5}{2.5} = \frac{10}{12} = 0.8333 \)
  - \( WP(W_1, C_1) = \frac{2.4}{2.4} = \frac{8}{12} = 0.6666 \)
  - \( WP(W_1, M_1) = \frac{2.0}{2.0} = \frac{0}{10} = 0.0000 \)

3.1.3 MSA Similarity. This specific method calculates the similarity distance based on the Wu-Palmer similarity. MSA similarity (see Eq. 3), however, uses a new factor \( \lambda \) which checks if the two comparing nodes appertain to the same hierarchy or not. Like WP, similar concepts have MSA close to 1 and the less similar have MSA close to 0. Only the identical nodes have MSA equal to 1. In order to understand the functionality of the metric, we are going to employ it, considering concepts of the same hierarchy depending on their distance from their LCA, denoted as \( distance_{LCA} \). Again the example of Figure 3 will be used with two different approaches. Firstly, we assume that concepts belong in the same hierarchy if they have the exact same (immediate) parent, i.e. their \( distance_{LCA} \) is 1:

- \( \lambda = 0 \) if \( distance_{LCA} = 1 \), otherwise \( \lambda = 1 \)
  - \( MSA(W_1, W_2) = \frac{2.5 e^6}{6.6 + 6.6} = \frac{10}{12} = 0.8333 \)
  - \( MSA(W_1, C_1) = \frac{2.4 e^6}{6.6 + 6.6} = \frac{8.0 e^{4.345}}{12} = 0.2897 \)
  - \( MSA(W_1, M_1) = \frac{2.0 e^6}{6.6 + 6.6} = \frac{0}{10} = 0.0000 \)

Then, we consider concepts of the same hierarchy whose distance from their common parent less than or equal to 2.

- \( \lambda = 0 \) if \( distance_{LCA} \leq 2 \), otherwise \( \lambda = 1 \)
  - \( MSA(W_1, W_2) = \frac{2.5 e^6}{6.6 + 6.6} = \frac{10}{12} = 0.8333 \)
  - \( MSA(W_1, C_1) = \frac{2.4 e^6}{6.6 + 6.6} = \frac{8}{12} = 0.6666 \)
  - \( MSA(W_1, M_1) = \frac{2.0 e^6}{6.6 + 6.6} = \frac{0}{10} = 0.0000 \)

We recognize that the choice of \( \lambda \) offers flexibility in the way similarity is defined. For instance, comparing the similarity measures of the two approaches for nodes \( W_1 \) and \( C_1 \), we note that when we assume that the concepts to the same hierarchy (having \( \lambda = 0 \)), they are deemed as more similar by the measure.

3.2 Non-Hierarchy Distance Measures

As already stated, Jaccard similarity belongs to the category of semantic similarity measures for sets of concepts. In our case, the finite set refers to some POIs from a tourist destination. We provide an evaluation of the metric by calculating similarities between two POIs. Note that the existence of a hierarchy tree is not needed, while if such exists, we are unaware of their hierarchy tree distance. Finally, we present three novel metrics inspired by Jaccard similarity and examine them as similarity measures for sets of concepts.

3.2.1 Jaccard Similarity. We now offer a concrete example to demonstrate how the Jaccard similarity will help us measure the similarity of two nodes without having a hierarchy tree at hand. In our example we are provided with two different sets of features, \( A \) and \( B \), each one corresponding to a tourist POI in Agios Nikolaos, Crete, with the features (members of sets) presented in Table 1. The size of the intersection \( |A \cap B| \) is given by the number of the underlined features, and the size of union \( |A \cup B| \) of POIs is computed by the total number of members in both sets. Thus, \( JS \) is calculated as:

- \( A \cap B = \{ \text{Cost, Wine variety, Natural landscape, Food variety, Wine cellar, Wifi, Wheelchair accessibility, Type} \} \)
- \( A \cup B = \{ \text{Cost, Wine variety, Natural landscape, Food variety, Wine cellar, Other drinks, Wifi, Wine production, Cultural info, Wine store, Wheelchair Accessibility, Type, Secondary Type, Wine distribution, Type of tour, Serving Type, Live music} \} \)

\[ JS(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{8}{17} = 0.4705 \]

\(^1\)Some members of the sets in these examples represent synthetic data which have been added for the purpose of showcasing the behavior of the various metrics based on different input data.
The result of the above calculation indicates the low similarity of the two finite sets \((A, B)\). Despite the fact that sets have many shared members, the available information of the POIs’ features allows us to infer their dissimilarity.

Let us now give a second example, using the Jaccard similarity for the above sets with fewer members of their union. The number of their shared members (intersection) is going to be identical with the number of the first example. The estimation of JS of the new sets \(C, D\) is presented below:

- \(C \cap D = \{\text{Cost}, \text{Wine variety}, \text{Natural landscape}, \text{Food variety}, \text{Wine cellar}, \text{Wifi}, \text{Wheelchair accessibility}, \text{Type}\}\)
- \(C \cup D = \{\text{Cost}, \text{Wine variety}, \text{Natural landscape}, \text{Food variety}, \text{Wine cellar}, \text{Other drinks}, \text{Wifi}, \text{Wine production}, \text{Cultural info}, \text{Wheelchair accessibility}, \text{Type}, \text{Wine distribution}, \text{Type of tour}\}\)
- \(\text{JS}(C, D) = \frac{|C \cap D|}{|C \cup D|} = 0.6153\)

The idea of this example is to demonstrate how the similarity of the sets \(C, D\) is affected by the number of features in the union. We see that the Jaccard similarity is greater when their union members are fewer than the first example \((A, B)\), and the shared members remained untouched. In this case, the Jaccard similarity equals to 0.6153, which compared to 0.4705, constitutes a significant increase in the estimate of similarity of the sets. However, the “Set C” POI is still a wine bar, while the “Set D” POI is still a monastery, thus the Jaccard similarity now provides us with a “signal” regarding semantic similarity that is rather distorted or flawed.

Another computation of JS will be presented by removing more members of the sets of the first example. Specifically, we remove more members of the union, some intersection members of the two sets and then we calculate their JS. The new sets \(E, F\) are:

- \(E \cap F = \{\text{Cost}, \text{Wine variety}, \text{Food variety}, \text{Wifi}, \text{Wheelchair Accessibility}, \text{Type}\}\)
- \(E \cup F = \{\text{Cost}, \text{Wine variety}, \text{Food variety}, \text{Other drinks}, \text{Wifi}, \text{Wine production}, \text{Wheelchair Accessibility}, \text{Type}\}\)
- \(\text{JS}(E, F) = \frac{|E \cap F|}{|E \cup F|} = 0.75\)

The JS of the third example is greater than the outcomes of the first two, indicating that the compared sets are more similar and JS is very sensitive to the number of features of the sets.

### 3.2.2 Extended Jaccard Similarity

The examples above indicated that Jaccard similarity is sensitive to the number of items in the comparing sets. Thus, it can be problematic when intending to employ JS. As such, we put forward an extended version of the JS, one that is potentially more appropriate for similarity comparisons. Extended Jaccard similarity (EJS) implements a comparison between the shared members of two sets and uses only the members with equal values for the calculation of the Jaccard index. Formally, we define the extended Jaccard similarity as:

\[
\text{EJS} = \frac{\sum_{i \in X \cap Y} x_i}{|X \cup Y|}
\]

where,

\[
x_i = \begin{cases} 
1, & \text{if } v^X_i = v^Y_i \\
0, & \text{otherwise}
\end{cases} \quad i \in X \cap Y
\]

\(v^X_i\) and \(v^Y_i\) denote the values of a given feature \(i\) for each set \(X\) and \(Y\) respectively, while feature \(i\) is a member of the intersection of the sets that are being compared.

Now we can test the functionality of our proposed metric, using the examples presented above. For each example, we are going to exploit the values of the shared members, seen in Table 2, in order to calculate the extended version of Jaccard similarity.

### Table 2: Feature values comparison of the example sets

<table>
<thead>
<tr>
<th>Sets A and B</th>
<th>Sets C and D</th>
<th>Sets E and F</th>
</tr>
</thead>
<tbody>
<tr>
<td>NaturalLand</td>
<td>WineCellar=</td>
<td>NaturalLand</td>
</tr>
<tr>
<td>FoodVariety</td>
<td>WineCellar=</td>
<td>FoodVariety</td>
</tr>
<tr>
<td>Cost</td>
<td>Cost</td>
<td>Cost</td>
</tr>
<tr>
<td>Time</td>
<td>Time</td>
<td>Type</td>
</tr>
<tr>
<td>NaturalLand</td>
<td>WineCellar=</td>
<td>NaturalLand</td>
</tr>
<tr>
<td>WineVariety</td>
<td>WineVariety=</td>
<td>WineVariety=</td>
</tr>
<tr>
<td>Cost</td>
<td>Cost</td>
<td>Cost</td>
</tr>
<tr>
<td>FoodVariety</td>
<td>FoodVariety=</td>
<td>FoodVariety=</td>
</tr>
<tr>
<td>Time</td>
<td>Time</td>
<td>Type</td>
</tr>
<tr>
<td>NaturalLand</td>
<td>WineCellar=</td>
<td>NaturalLand</td>
</tr>
<tr>
<td>FoodVariety</td>
<td>WineCellar=</td>
<td>FoodVariety=</td>
</tr>
<tr>
<td>Cost</td>
<td>Cost</td>
<td>Cost</td>
</tr>
<tr>
<td>Time</td>
<td>Time</td>
<td>Type</td>
</tr>
<tr>
<td>NaturalLand</td>
<td>WineCellar=</td>
<td>NaturalLand</td>
</tr>
<tr>
<td>FoodVariety</td>
<td>WineCellar=</td>
<td>FoodVariety=</td>
</tr>
<tr>
<td>Cost</td>
<td>Cost</td>
<td>Cost</td>
</tr>
<tr>
<td>Time</td>
<td>Time</td>
<td>Type</td>
</tr>
<tr>
<td>NaturalLand</td>
<td>WineCellar=</td>
<td>NaturalLand</td>
</tr>
<tr>
<td>FoodVariety</td>
<td>WineCellar=</td>
<td>FoodVariety=</td>
</tr>
<tr>
<td>Cost</td>
<td>Cost</td>
<td>Cost</td>
</tr>
<tr>
<td>Time</td>
<td>Time</td>
<td>Type</td>
</tr>
<tr>
<td>NaturalLand</td>
<td>WineCellar=</td>
<td>NaturalLand</td>
</tr>
<tr>
<td>FoodVariety</td>
<td>WineCellar=</td>
<td>FoodVariety=</td>
</tr>
<tr>
<td>Cost</td>
<td>Cost</td>
<td>Cost</td>
</tr>
<tr>
<td>Time</td>
<td>Time</td>
<td>Type</td>
</tr>
<tr>
<td>NaturalLand</td>
<td>WineCellar=</td>
<td>NaturalLand</td>
</tr>
<tr>
<td>FoodVariety</td>
<td>WineCellar=</td>
<td>FoodVariety=</td>
</tr>
<tr>
<td>Cost</td>
<td>Cost</td>
<td>Cost</td>
</tr>
<tr>
<td>Time</td>
<td>Time</td>
<td>Type</td>
</tr>
<tr>
<td>NaturalLand</td>
<td>WineCellar=</td>
<td>NaturalLand</td>
</tr>
<tr>
<td>FoodVariety</td>
<td>WineCellar=</td>
<td>FoodVariety=</td>
</tr>
<tr>
<td>Cost</td>
<td>Cost</td>
<td>Cost</td>
</tr>
<tr>
<td>Time</td>
<td>Time</td>
<td>Type</td>
</tr>
<tr>
<td>NaturalLand</td>
<td>WineCellar=</td>
<td>NaturalLand</td>
</tr>
<tr>
<td>FoodVariety</td>
<td>WineCellar=</td>
<td>FoodVariety=</td>
</tr>
<tr>
<td>Cost</td>
<td>Cost</td>
<td>Cost</td>
</tr>
<tr>
<td>Time</td>
<td>Time</td>
<td>Type</td>
</tr>
<tr>
<td>NaturalLand</td>
<td>WineCellar=</td>
<td>NaturalLand</td>
</tr>
<tr>
<td>FoodVariety</td>
<td>WineCellar=</td>
<td>FoodVariety=</td>
</tr>
<tr>
<td>Cost</td>
<td>Cost</td>
<td>Cost</td>
</tr>
<tr>
<td>Time</td>
<td>Time</td>
<td>Type</td>
</tr>
<tr>
<td>NaturalLand</td>
<td>WineCellar=</td>
<td>NaturalLand</td>
</tr>
<tr>
<td>FoodVariety</td>
<td>WineCellar=</td>
<td>FoodVariety=</td>
</tr>
<tr>
<td>Cost</td>
<td>Cost</td>
<td>Cost</td>
</tr>
<tr>
<td>Time</td>
<td>Time</td>
<td>Type</td>
</tr>
</tbody>
</table>

As we noticed, five of eight shared members of the given sets are exactly equal, in the sense that they are the same and have exactly the same values. The extended Jaccard similarity is computed as:

\[
\text{EJS}(A, B) = \frac{\sum_{i \in A \cap B} x_i}{|A \cup B|} = \frac{5}{17} = 0.2941
\]

\(^2\)We suppose the cost values are equal if the difference in cost is not very large. In addition, the wine varieties of the two POIs are equal, because the wine variety of set B is a subset of wine variety of set A.
The outcome of the extended version of Jaccard similarity represents the difference when each one of the shared members for both sets \( A, B \) is being checked. The values of the shared members provide more information about the features each POI has. For instance, a better perspective of the similarity of the sets is acquired when an intersection member such as "Type" is compared.

We attempt to deploy our new measure for sets \( C, D \). Firstly, the members of their intersection are being checked in Table 2. Then, the extended Jaccard similarity is calculated:

\[
EJS(C, D) = \frac{\sum_{i \in \text{\text{C}} \cap \text{\text{D}}} x_i}{|C \cap D|} = \frac{3}{11} = 0.3846
\]

The similarity of sets \( C, D \) is greater than the similarity of sets \( A, B \), because of the reduced number of union members of the second case. Less information about the POIs’ features provides a higher similarity result, however the \( EJS \) is capable of still capturing the dissimilarity between the two POIs: their similarity score is still low, as even though they share several features, these do not have the same values. The extended Jaccard similarity can produce better results than the original version, giving us the opportunity to recognize the similarity distance between sets of a hierarchy tree.

We are going to deploy our measure for sets \( E, F \), and compare the results with the original version of the Jaccard index. The extended Jaccard similarity is computed as:

\[
EJS(E, F) = \frac{\sum_{i \in \text{\text{E}} \cap \text{\text{F}}} x_i}{|E \cap F|} = \frac{4}{8} = 0.50
\]

It turns out that two very similar sets (such as \( E, F \)) based on the Jaccard index, become less similar by employing our proposed similarity measure. The comparison of the feature values provides more informative outcomes about the similarity distance between two sets, and is less sensitive than the original version of Jaccard similarity to the reduction of the features of the sets.

### 3.2.3 Weighted Jaccard Similarity

Earlier we presented the deployment of Jaccard index for evaluating the similarity (or diversity) of finite sets and specifically for POIs through several examples. The outcomes showed that Jaccard index can be quite efficient when we have been provided with enough information about the features of the compared sets. The extended Jaccard similarity, on the other hand, is capable of creating reliable results by including the important content given by the feature values of the compared POIs: the dissimilarity in feature values led to POIs being deemed less similar by \( EJS \) than when employing the original JS version. However, it was observed that this reduction in similarity by \( EJS \) can be rather drastic, while \( EJS \) does not take user preferences into account.

We thus now come up with a new technique for discovering the similarity of two finite sets, which unlike the previous methods, utilizes weights for all POIs features based on the user’s preferences. We assume these weights sum up to 1.

Formally, we define the weighted Jaccard similarity (WJS)\(^3\) for two finite sets \( (X, Y) \) as:

\[
WJS = \frac{\sum_{i \in X \cap Y} w_i}{\sum_{j \in X \cap Y} w_j}
\]

where, \( w_i \) is the weight of \( i^{th} \) member of the intersection of two compared sets and \( w_j \) is the weight of \( j^{th} \) member of the union.

\( ^3 \)Note that the Ruzicka similarity [17] is sometimes mentioned as weighted Jaccard similarity in the literature, while the Tanimoto coefficient [16] is also called extended Jaccard. Those terms should not be confused with the concepts defined in our work.

The weighted Jaccard similarity will be 0 if the two sets do not share any values and 1 if the two sets are identical. We now demonstrate the effectiveness of the weighted Jaccard similarity by employing it on the sets \( E \) and \( F \) described in the previous examples. Our example makes apparent the importance of having features of the intersection with high weight values.

We provide four different example applications of the weighted index by presenting four different assignments of weight values of the \( E, F \) sets described above and we are going to calculate the WJS for each one. For the purpose of distinguishing the features that are members of the intersection we underline them in our examples. Specifically, the scenarios are presented in Table 3.

Table 3: Weight assignment for each scenario

<table>
<thead>
<tr>
<th>Measures</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Scenario 3</th>
<th>Scenario 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>0.125</td>
<td>0.05</td>
<td>0.30</td>
<td>0.20</td>
</tr>
<tr>
<td>Wine Varieties</td>
<td>0.125</td>
<td>0.40</td>
<td>0.02</td>
<td>0.40</td>
</tr>
<tr>
<td>Food Varieties</td>
<td>0.125</td>
<td>0.20</td>
<td>0.02</td>
<td>0.025</td>
</tr>
<tr>
<td>WiFi</td>
<td>0.125</td>
<td>0.05</td>
<td>0.02</td>
<td>0.025</td>
</tr>
<tr>
<td>Wheelchair</td>
<td>0.125</td>
<td>0.05</td>
<td>0.02</td>
<td>0.025</td>
</tr>
<tr>
<td>Type</td>
<td>0.125</td>
<td>0.05</td>
<td>0.02</td>
<td>0.025</td>
</tr>
<tr>
<td>Other Drinks</td>
<td>0.125</td>
<td>0.10</td>
<td>0.30</td>
<td>0.15</td>
</tr>
<tr>
<td>Wine Proof</td>
<td>0.125</td>
<td>0.10</td>
<td>0.30</td>
<td>0.15</td>
</tr>
</tbody>
</table>

The weight utilization of scenario 1 describes that the user does not have a special bias for the features presented in Table 3. In scenarios where the weight values of “wine variety” are high such as scenarios 2, 4 present the significance of including a “wine variety” as feature by the POIs based the user’s preferences.

The weighted Jaccard similarity for each scenario is calculated based on Equation 7 for sets \( E, F \) in the first line of Table 4. Observing the examples, while we change the weight values the outcomes of the weighted Jaccard similarity vary. In cases where the weight values of the intersection members are high the compared sets are very similar like scenarios 2, 4. In contrast, when features, that are only union members, have higher weight values the similarity of the sets decreased unambiguously (i.e., scenario 3).

Table 4: WJS and WEJS values per scenario (for the \( E, F \) sets)

<table>
<thead>
<tr>
<th>Measures</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Scenario 3</th>
<th>Scenario 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>WJS</td>
<td>0.75</td>
<td>0.80</td>
<td>0.40</td>
<td>0.70</td>
</tr>
<tr>
<td>WEJS</td>
<td>0.50</td>
<td>0.55</td>
<td>0.36</td>
<td>0.65</td>
</tr>
</tbody>
</table>

### 3.2.4 Weighted Extended Jaccard Similarity

Up to this point, we defined two new metrics based on the Jaccard similarity and each one is capable of measuring the similarity of two sets. \( EJS \) metric extracts the valuable information about the similarity of two POIs through the comparison of the feature values. On the other hand, \( WJS \) employs a weight factor that reflects the importance of each value based on the user’s interests and exploits it to calculate the similarity of the sets. It would be very interesting to develop a new metric as the combination of the aforementioned measures for the calculation of the similarity of the sets.
Weighted extended Jaccard similarity (WEJS), as we could name it, takes advantage of the weight values (inspired by weighted Jaccard similarity) of the different features when only their feature values are equal (inspired by extended Jaccard similarity) in order to measure the similarity. Content of the POIs provided by the feature values and the user’s preferences are taken into consideration in order to create a new metric capable of including all the important information and measuring the similarity between sets. Specifically, we assume all the members (or features) that belong to possible \( X \) and \( Y \) respectively, while feature values are high (values of the intersection members, specially when their weight values and the user’s preferences are taken into consideration in their actual values) when \( i \in X \cap Y \).

The weighted extended Jaccard similarity is defined as:

\[
WEJS = \frac{\sum_{i \in X \cap Y} \alpha \cdot w_i}{\sum_{i \in X \cup Y} w_j}
\]

where, \( v^X_i \) and \( v^Y_i \) denote the values of a given feature \( i \) for each set \( X \) and \( Y \) respectively, while feature \( i \) is a member of the intersection of the sets. \( w_i \) is the weight of \( i^{th} \) member of the intersection of two compared sets and \( w_j \) is the weight of \( j^{th} \) member of the union. \( \alpha \) is the variable that checks when the compared values of the features are equal. The \( WEJS \) will be 0 if the two sets do not share any values and 1 if the two sets are identical. We demonstrate this index’s effectiveness by employing it on the sets \( E, F \) for the scenarios of Table 3 and comparing their feature values in Table 2. Like the \( JWS \) the union members are the only features that we assigned weights, so the sum of their weights is equal to 1. The results of each different scenario are depicted in Table 4.

In the first case, where the weight values of all features are equal, the result of our new metric is similar to the outcome of extended Jaccard similarity for sets \( E, F \). However, these compared sets are less similar than the corresponding example in weighted Jaccard similarity because of the unequal values of features “Type” and “Food Variety”. We should recall that, the comparison implemented between a wine bar and a monastery and the information provided for the pairs of items is very different. For example, in scenarios 2 and 4 of the above example the similarity of the compared sets changes dramatically, as soon as we assign a higher weight to the “Cost” feature which has the similar value for both sets.

### 4 EXPERIMENTAL EVALUATION

In this section, we evaluate the metrics introduced earlier (see Section 3) on a real world dataset including 430 POIs in the vicinity of Agios Nikolaos, Crete, Greece. As we already mentioned, the creation of the dataset is one of our contributions in this work, and uses 58 features to describe all the included POIs. The first 14 features represent the character of a POI (e.g., how cultural or how family friendly a POI is) and take values in the range of [1, 10], with such a value corresponding to the degree that this feature describes a POI. The characteristics, such as the “Type” of a POI and the amenities a POI can provide, are captured by the last 44 features. An example of these features is presented in Table 1.

#### 4.1 Hierarchy Distance Measures Experiments

We construct a hierarchy, which has similar form as the tree structure in Figure 3, by exploiting the 58 features that describe all the available POIs. We note that each POI may belong to more than one node of the hierarchy tree (i.e., having multiple ancestors) depending on the information provided by the 58 features. Two POIs are chosen randomly as inputs for the purpose of examining the distance measures of Section 3.1. In a real world application, input POIs represent items that a user “likes” in order to get recommendations similar to her choices, while the recommendation technique is based on hierarchy distance measures.

<table>
<thead>
<tr>
<th>Table 5: Hierarchy distance measures results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input POI</td>
</tr>
<tr>
<td>Kritsa’s Gorge</td>
</tr>
<tr>
<td>Wu-Palmer</td>
</tr>
<tr>
<td>MSA(1)</td>
</tr>
<tr>
<td>MSA(2)</td>
</tr>
</tbody>
</table>

The similarity distance results between the first input (Kritsa’s Gorge, ancestor = Gorge), the second input (Minos, ancestor = Men Casual) and the remaining POIs of the dataset, are illustrated in Table 5. The 20 most similar POIs are provided as a ranking list alongside their names, ancestor nodes and similarity outcomes based on path length similarity, Wu-Palmer similarity and two different versions of MSA similarity (MSA(1), MSA(2)). We remind the reader that lower score values denote higher similarity objects in the case of path length similarity; while larger values denote higher similarity objects in cases on Wu-Palmer similarity and MSA similarity. Furthermore, Wu-Palmer, MSA(1) and MSA(2) are depicted together, since they return the same ranking of recommendations (which is expected since they operate similarly). As described earlier (see Section 2.1) the Wu-Palmer, MSA(1) and MSA(2) metrics capture the similarities among items by exploiting the whole hierarchy tree structure, as they consider (i) the distance from the hierarchy root

5Note that those features also correspond to preference-related data collected from actual tourists via questionnaires.
of the compared POIs; and (ii) the distance of the compared POIs’ LCA from the hierarchy root. \( \text{MSA}(1) \) considers nodes as originating from the same hierarchy and thus having a \( \lambda = 0 \) when their \( \text{distance}_{\text{LCA}} = 1 \), and sets \( \lambda = 1 \) for \( \text{distance}_{\text{LCA}} > 1 \); while \( \text{MSA}(2) \) sets \( \lambda = 0 \) to those items with \( \text{distance}_{\text{LCA}} > 2 \) (otherwise \( \lambda = 0 \)). Still, \( \text{MSA}(1) \) does rank the POIs in a meaningful way (albeit with a lower score than \( \text{Wu-Palmer} \) or \( \text{MSA}(2) \)).

In Table 5 the ranking lists for both inputs are very similar for all aforementioned metrics. However the \( \text{Wu-Palmer} \), \( \text{MSA}(1) \) and \( \text{MSA}(2) \) metrics provide more informative scores regarding to the similarity between two compared POIs. For instance, \( \text{Alexisport} \) (i.e., the output POI for input POI \( \text{Minos} \)) has \( \lambda = 4 \); while \( \text{Oxia} \) (i.e., the output POI for input POI \( \text{Kritsa’s Gorge} \)) has also \( \lambda = 4 \). As such, in both scenarios the \( \lambda \) scores are identical (equal to 4), signifying simply that the output POIs lie in the same distance from their input POIs, as calculated with respect to their LCA for each input POI respectively. However, more informative similarity scores appear when \( \text{Wu-Palmer} \), \( \text{MSA}(1) \) and \( \text{MSA}(2) \) are employed as our hierarchy tree is an asymmetrical structure (i.e., trees have varying numbers of children per node, and different records might lie at different depths). For the examples above, \( \text{Alexisport} \) has \( \text{Wu-Palmer} = 0.8 \) while \( \text{Oxia} \) has \( \text{Wu-Palmer} = 0.7142 \), as the depth of the compared POIs is taken into account by the \( \text{Wu-Palmer} \) metric.

Notice that, naturally, for any hierarchy-based measure to function properly, it is crucial to have a well-defined hierarchy tree in place. However, given the potentially enormous number of POIs to place in a hierarchy appropriately, the use of hierarchy distance measures requires post-filtering techniques for excluding a portion of the results. This can be done based on different factors, such as the POIs’ ratings or the user’s preferences.

### 4.2 Non-Hierarchy Distance Measures

Experiments

The evaluation of \( \text{Jaccard similarity} \) (JS) and its proposed variants is conducted via a new series of experiments with the assistance of \( \text{Kritsa’s Gorge} \) and three more POIs of the dataset as inputs. We remind the reader, larger similarity scores for JS and its variations denote higher similarity objects. In the manner of a RS application, input POIs represent items that a user “likes” and the recommendation technique is based on non-hierarchy distance measures. While no hierarchy structure is employed and the ancestors are not provided, all the experiments present POIs (inputs and outputs) along with the “Type” feature which is part of the last 44 features of the dataset. We stress that the calculation of the JS and its variations for the members of each compared set, i.e., POI, is based on a subset of the last 44 features. Specifically, we consider only the features that have a non-empty field, i.e., are assigned to a value.

Table 6 depicts the experiments conducted by employing JS and extended Jaccard similarity (EJS) for each input POI respectively. The top ten recommendations are presented in accordance with the similarity score (JS or EJS), output POI’s name and “Type” feature we mentioned earlier. In the case of \( \text{Moni Toplou} \) (“Type” = monastery), JS outputs low similarity scores and none of the top ten recommendations includes another item of the same “Type” feature, because of the incapability to integrate information about the feature values. On the other hand, EJS does incorporate such data, and thus the ranking changes to better reflect POIs similarities or differences. For instance, the similarity scores for the input POI (\( \text{Moni Toplou} \))-related items are lower for EJS, because the compared feature values of the input are not equal to the corresponding feature values of the other POIs. However, a POI with equal “Type” feature (monastery) now appeared in the first place, while JS had ranked a “monument” at the top and had failed to output a “monastery” as a “top-ten” similar item. In the other three cases, JS outcomes have large similarity scores in the top ten places of the ranking list, because they share many members (i.e., intersection members), but it is problematic for inputs such as Skala fusion taverna (“Type” = restaurant) where a portion of the most similar items have “Type” = fast food. EJS similarity scores are also lower than the respective JS outcomes because of the comparison the POIs’ features values for the remaining cases. We should note that JS and EJS create only non-personalized recommendations as no information about user’s preferences is included. However, as is evident in the results, EJS results are more fine-grained, with outputs’ scores belonging in a wider range of values; while JS scores are characterized by much homogeneity. The similarity scores for both metrics are sensitive to the number of features given by the compared POIs (i.e., features assigned to some value) and are large specially when enough information is provided such as the cases of \( \text{Fysalida} \) and \( \text{Skala fusion taverna} \) that are taken as inputs.

<table>
<thead>
<tr>
<th>Input POI Short Name</th>
<th>Type</th>
<th>JS</th>
<th>EJS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skala fusion taverna</td>
<td>Type:Restaurant</td>
<td>0.4615 Must-Grill House fish meat Restaurant</td>
<td>0.8 Palermo Fast Food</td>
</tr>
<tr>
<td>Fysalida</td>
<td>Type:Bar</td>
<td>0.4 Tou Listi o Spilios Attraction</td>
<td>0.75 Ancient Naxos Attraction</td>
</tr>
<tr>
<td>I.M. Panagia Faneromenis Monastery</td>
<td>Type:Monastery</td>
<td>0.2857 I.M. Panagia Faneromenis Monastery</td>
<td>0.4166 Saint Athanasios Church</td>
</tr>
<tr>
<td>BEERaki Cretan Pub Bar</td>
<td>Type:Bar</td>
<td>0.1875 Filema Caffe Cafeteria</td>
<td>0.3846 Siganos - Winespirits Store</td>
</tr>
</tbody>
</table>

**Table 6: JS, EJS Experiments**

**Table 7: The respective 14 features of the 4 input POIs presented**

<table>
<thead>
<tr>
<th>Input POI Short Name</th>
<th>Type</th>
<th>Type:Monastery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skala fusion taverna</td>
<td>Type:Restaurant</td>
<td>0.4615 Must-Grill House fish meat Restaurant</td>
</tr>
<tr>
<td>Fysalida</td>
<td>Type:Bar</td>
<td>0.4 Tou Listi o Spilios Attraction</td>
</tr>
<tr>
<td>I.M. Panagia Faneromenis Monastery</td>
<td>Type:Monastery</td>
<td>0.2857 I.M. Panagia Faneromenis Monastery</td>
</tr>
<tr>
<td>BEERaki Cretan Pub Bar</td>
<td>Type:Bar</td>
<td>0.1875 Filema Caffe Cafeteria</td>
</tr>
</tbody>
</table>

**Table 7: The respective 14 features of the 4 input POIs presented**

<table>
<thead>
<tr>
<th>Input POI Short Name</th>
<th>Type</th>
<th>Type:Monastery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skala fusion taverna</td>
<td>Type:Restaurant</td>
<td>0.4615 Must-Grill House fish meat Restaurant</td>
</tr>
<tr>
<td>Fysalida</td>
<td>Type:Bar</td>
<td>0.4 Tou Listi o Spilios Attraction</td>
</tr>
<tr>
<td>I.M. Panagia Faneromenis Monastery</td>
<td>Type:Monastery</td>
<td>0.2857 I.M. Panagia Faneromenis Monastery</td>
</tr>
<tr>
<td>BEERaki Cretan Pub Bar</td>
<td>Type:Bar</td>
<td>0.1875 Filema Caffe Cafeteria</td>
</tr>
</tbody>
</table>

**Table 7: The respective 14 features of the 4 input POIs presented**

Weighted Jaccard similarity (WJS) and weighted extended Jaccard similarity (WES) are examined via a series of experiments where user’s preferences are taken into consideration. Two synthetic users, whose preferences are captured by 14 features, identical with the first 14 features of our dataset, are depicted in the first two rows of Table 7. The respective 14 features of the 4 input POIs presented above, are also given in Table 7. Notably, the two users have similar feature vectors with two of the four given POIs. As such, \( \text{User}0 \) will
be referred as a "monastery-user", since she is a user with a feature vector (i.e., the 14 features provided in Table 7) similar to that of Moni Toplou POI, which has a "Type" feature equal to "monastery". Similarly, User1 is a "bar-user", in the sense that she has a feature vector similar to that of Fysalida (whose "Type" is a "bar"). Both metrics integrate the user’s preferences by weights assigning to the last 44 features of the POIs. In order to accomplish this, cosine similarity [9] is employed between the user’s feature vector and all POIs’ respective feature vector (i.e., the first 44 features of the dataset). The most similar POIs are chosen, based on a threshold, exported by cosine similarity, and the number of appearances of a feature (last 44 features) is counted, indicating its level of importance based on the user’s preferences. We note that in this counting process, we consider only the features that have a non-empty field, i.e., they are assigned to some value. Thus, the counters are transformed into weights as we normalize them within [0, 1] for all 44 features, and all sum up to 1.

Table 8a and Table 8b present the WJS and WEJS similarity scores for bar-user and monastery-user respectively for the 4 input POIs. Large WJS values appear for both users, as no information about the feature values is imported in this similarity measure.

In cases like Fysalida, Krītsa’s Gorge and Skala fusion taverna the large WJS values indicate many intersection members between the input POIs and the rest of the dataset for both users. When Moni Toplou is given as input, the WJS scores are lower than the other outputs for both users, because of the reduced number of the shared members. Notably, the ranking lists for each input between the two users differ, as their preferences are taken into account through the integration of the weights of the 44 POIs features. However, the recommendations provided by the WJS metric are characterized by homogeneity and poor performance (see the cases where Moni Toplou and Skala fusion taverna are the input POI for both users, in which the “Type” of the outputs very seldomly matches that of the input).

On the other hand, WEJS combines the information about the content of the POIs and the user’s preferences. The similarity scores are lower than the respective outcomes of WJS because of the feature values’ comparison (i.e., 44 POIs’ features). The ranking lists now differ based on each user’s preferences for each input POI. In the case of Moni Toplou, low similarity scores in the top ten ranks indicate limited information about the feature values of the compared POIs for both users, however the items are listed in accordance with the user’s interests: For instance, a store (Grainini A.E.) which provides wines and spirits appears in the first place.
above the items with “Type = monastery” for the bar-user, while the monasteries are recommended first for the monastery-user because her feature vector is similar to the Moni Toplou monastery. We note that, items with “Type = store” or “Type = bar” are still presented in the top ten ranks, as many of them provide a wine list or wine tasting, like Moni Toplou, and this information is given by the features (i.e., POIs’ content) of our dataset. As such, WEJS provides different similarity outcomes based on the different users, and is able to create diverse recommendations. In the remaining cases, where enough information is granted items with the same “Type” as the input POI are listed in the top ten ranks. Here the “richness” of the recommended items is also observed. For example, in the case of Kritsa’s Gorge many items with “Type = attraction” are not gorges, such as Oxia and Okismois Duo Prinio (i.e., in Table 5 their ancestor node is “mountain-top”) which are recommended by WEJS to both users.

5 CONCLUSIONS AND FUTURE WORK

We experimented with several semantic similarity measures for computing the similarity between POIs in order to create content-based recommendations in the tourist domain. Well-known similarity distance measures are examined by operating them upon hierarchies of POIs. Progressively, we build three novel versions of Jaccard similarity and we provide their advantages and disadvantages by employing them to measure the similarity between POIs. We end up with WEJS, a hierarchy-free measure, capable of recommending POIs by combining information based on the user’s preferences and the feature values of POIs. WEJS is able to create “rich” recommendations by capturing similarities among POIs that are relevant to the user’s preferences. There are several items on the agenda of ongoing and future work. First, we are in the process of performing more extensive tests of proposed similarity metrics on a dataset enriched with more information provided by the feature values of POIs. Moreover, we intend to deploy and test our approach with actual users, via incorporating the various measures in different versions of a real-world software application for tours planning currently under development. Finally, we aim to exploit the real-world data that will be collected via this testing process, in order to explore the possibility of combining hierarchy-based and non-hierarchy distance measures for a highly satisfactory user experience.

ACKNOWLEDGMENTS

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH –CREATE-INNOVATE B cycle (project code: T2EDK-03135). E. Streviniotis was also supported by the Onassis Foundation - Scholarship ID: G ZR 012-1/2021-2022.

REFERENCES


