Open topics for students: Recommendations

Interactive Recommendations

With the growing complexity of the Web, users often find themselves overwhelmed by the mass of choices available. Shopping for DVDs, books or clothes online becomes more and more difficult, as the variety of offers increases rapidly and gets unmanageable. To facilitate users in their selection process, recommender systems provide suggestions on items, which might be interesting for the respective user. In particular, recommender systems aim at giving recommendations to users or groups of users by estimating their item preferences and recommending those items featuring the maximal predicted preference. The prerequisite for determining such recommendations is historical information on the users’ interests, e.g., the users’ purchase history.

In this project, our target is at giving users the access to control the way that the system computes recommendations for them. Specifically, we focus on several properties, such as popularity, freshness and diversity, and based on them, we would like to perform actions like “show well-known items”, “show new items”, or “show a set of items that exhibit some diversity”.

Progressive Recommendations

Given the huge amount of data, possibly produced as streams, that is available nowadays, a typical recommender system may need quite enough time for identifying qualitative suggestions. To overcome this delay, one direction is to follow a progressive approach that tries to maximize the number of computed suggestions, given a limited computing budget (e.g., in time or number of comparisons). Essentially, a progressive approach extends the typical workflow of a recommender system with a scheduling phase, which is responsible, for example, for selecting the best users in terms of similarity to a given user, based on which the recommended items will be computed. In this simple scenario, the goal is to favor more promising users, i.e., those that are more likely to have common interests with the user in query. This way, based on those users, we locate interesting suggestions early on in the process, before less promising suggestions computed taking into account users that are less similar to the given user. To generalize, in this project, we focus on different techniques that achieve the best suggestions first. Compared to a non-progressive approach, at any time, the progressive one yields more recommendations, while both yield the same final result.

Kostas Stefanidis
Contact: kostas.stefanidis@uta.fi
Open topics for students: Entity Resolution in the Web of Data

Over the past decade, numerous knowledge bases (KBs) have been built to power large-scale knowledge sharing, but also an entity-centric Web search, mixing both structured data and text querying. These KBs offer comprehensive, machine-readable descriptions of a large variety of real-world entities (e.g., persons, places, products, events) published on the Web as Linked Data (LD). Although KBs (e.g., DBpedia, Freebase) may be derived from the same data source (e.g., a Wikipedia entry), they may provide multiple, non-identical descriptions of the same real-world entities. Entity resolution (ER) aims to identify descriptions that refer to the same real-world entity appearing either within or across KBs.

The two core ER problems, namely how can we (a) effectively compute similarity of entity descriptions and (b) efficiently resolve sets of entities within or across sources, are challenged by the large scale (both in terms of the number of sources and entity descriptions), the high diversity (both in terms of number of entity types and properties) and the importance of relationships among entity descriptions (not committing to a particular schema defined in advance).

Evaluating Similarity Functions for Entity Resolution

The objective of entity resolution is to find the set M of pairs of descriptions that correspond to the same real-world entities. This knowledge about matching real-world entities is estimated by the set S of pairs of descriptions that, according to a similarity function, correspond to the same entities. The choice of the similarity function determines the quality of estimating M by S. Realistic similarity functions typically lead to discover only a fraction of the matches, as well as some non-matches. Their effectiveness is measured in terms of the fraction of the matches identified and the fraction of the suggested matches that are correct.

In this project, we will evaluate the main similarity functions that have been employed to resolve entity descriptions in the Web of data. Similarity functions compare descriptions either exclusively on their content (i.e., attribute values), or additionally on their structural relations with other descriptions.

Progressive Entity Resolution

Given that ER may entail a big overhead, we are interested in maximizing its benefit, given a computational cost budget. So, we need to estimate which part of the data is the most promising to explore first, in a progressive way. Specifically, we focus on extending the typical ER workflow with a scheduling phase, which is responsible for selecting which pairs of descriptions will be compared in an entity matching phase and in what order. The goal of this scheduling phase is to favor more promising comparisons, i.e., those that are more likely to increase the targeted benefit. This way, those comparisons are executed before less promising ones and thus, higher benefit is provided early on in the process. An update phase propagates the results of matching, such that a new scheduling phase will promote the comparison of pairs that were influenced by the previous matches. This iterative process continues until the cost budget is consumed.

On Recommending Entities

Nowadays, we face an entity-centric organization of data, in which real-word entities are described with semi-structured data in several KBs, and can be fused to improve various aspects of data quality. In more complex cases, entity descriptions can be annotated simultaneously with different semantic types, indicating various facets of those entities. Interlinking such descriptions, by exploiting the several types of available links, helps toward identifying complementary descriptions for the same entities.

In this project, we aim at enabling powerful new user experiences in an entity-centric Web infrastructure, from search results that directly show key facts about people, places, and things, to improved refinement
interfaces that allow searchers to discover information that covers their needs. That is, in addition to semantically enriching the answers of keyword queries with references to entities that are mentioned in the queries, our focus is on providing recommendations of related entities based on relationships explicitly encoded in a KB.

Kostas Stefanidis

Contact: kostas.stefanidis@uta.fi