Generating Personalized Destination Suggestions for Automotive Navigation Systems under Uncertainty

Michael Feld[1], Martin Theobald[2], Christoph Stahl[1], Timm Meiser[2], Christian Müller[1]

[1] German Research Center for Artificial Intelligence (DFKI)
[2] Max Planck Institute for Informatics (MPII)
Saarbrücken, Germany

Abstract. Programming a car's navigation system manually takes time and is error-prone. When the address is not handy, a cumbersome search may start. Changing the destination while driving is even more problematic. Given a modern car's role as an information hub, we argue that an intelligent system could in many cases infer the right destination or have it among the top N suggestions. In this work, we propose a personalized navigation system that is built from three main ingredients: strong user models, knowledge source fusion, and reasoning under uncertainty. We focus on emails as one particular knowledge source, exploring the uncertainties involved when extracting empirical data of email appointments.

Keywords: Knowledge Fusion, Uncertainty Reasoning, Information Extraction

1 Personalizing Destination Suggestions

If you take the car today and need directions, the destination must be entered manually into the navigation system. Even if one is familiar with the route, the use of such a system is beneficial as it provides current traffic information. But no matter what kind of input device one uses, the process takes some time and is error-prone. If one feels confident to remember the route, things can even become dangerous when the navigation system is programmed later while driving. Existing knowledge sources can be exploited to suggest an accurate set of personalized, situated navigational destinations to the driver in order to reduce the necessary interaction and avoid distractions from the traffic. Possible knowledge sources that contain clues about the destination are calendars and address books, usually stored on mobile devices or services in the cloud. Additional sources include email communication and GPS sensor logs, even though they are less structured and inherently unreliable. Our main claims are: 1) Extracting appointment information (in order to derive possible destinations) from emails is subject to uncertainties. 2) Taking into account probabilistic models allows for an accurate ranking of destination suggestions with uncertain and potentially conflicting destination information obtained from these extractions.

The proposed system is therefore designed to maximize the accuracy of personalized destination suggestions by dealing with uncertainties, using a combination of rule-based reasoning and probabilistic ranking.

Aggregating user information in a common knowledge base (see [2]) provides an ideal set-up for further reasoning tasks, which in turn enables personalization. However, much of the collected information is not known for certain, therefore care must be taken when drawing conclusions from it. The reasons are manifold, and include accuracy of information extraction (IE) systems, quality of pattern recognition models, precision of hardware sensors, human errors, etc. Using emails (as a mostly unstructured source for extracting appointment data) requires IE methods that deal with the automatic discovery of information in text. We have performed a human analysis of appointment specifications on an email corpus. 350 mails were considered for this study, where 29% of these mails contained a total of 143 appointments. The messages were manually annotated according to a fixed scheme. Incomplete time, place, and attendee information was given in 12%, 64%, and 21% of the events, respectively. Data was however in a straightforward, standard parsable format in only 36%, 9%, and 27% of the cases. Overall, the study confirms two things: 1) In many cases, emails contain the relevant time, location, and attendee information for meetings. 2) In a few cases, the information can be easily extracted, but in the majority of cases, a more sophisticated approach (e.g., using NLP techniques) is needed, which introduces uncertainties.

2 URDF Reasoning Framework

The URDF reasoning framework [5] we use as our reasoning backend provides a SPARQL-compliant query model for knowledge bases captured in RDF/S. In addition to constraints expressible in RDF/S, URDF supports Datalog-style *soft rules* which are grounded against the base facts provided by the RDF knowledge base. Via these soft inference rules, URDF can also derive new facts which were not initially present in the knowledge base itself. Moreover, as soft rules may be noisy as well, *hard rules* can be employed to enforce consistency constraints over both base and derived facts. The initial grounding phase of URDF is followed by a subsequent *consistency reasoning* phase, where probabilistic inference techniques are applied to calculate the confidence of derived facts. Confidence computations are based on the lineage (i.e., the derivation structure) of facts inferred from rules, which captures the logical dependencies of the derived facts back to the base facts that were used for grounding. Moreover, lineage also provides a convenient means for *explaining* how these answers were derived [4].

Inference Rules. While queries in URDF are conjunctions over subject-predicate-object (SPO) patterns just like in SPARQL, the presence of rules drastically impacts how these queries are evaluated, and how potential conflicts are resolved. Soft rules have the form of implications (Horn clauses), with exactly one positive head literal, while the body of the rule is a conjunction of positive literals. As an example, suppose we have the following knowledge base, consisting of a number of base facts extracted from various email correspondences, as well as an inference rule about the possible destinations `?eloc` of a user `?x`, given the current time `?ctime` and location `?cloc`.

```
hasEvent(Mike, E1) [0.7].
eventTime(E1, 24.01/10:00) [0.6].
eventLoc(E1, DFKI-KL) [0.8].

hasDestination(?x, ?eloc, ?ctime, ?cloc) :-
       hasEvent(?x, ?e) ∧ eventTime(?e, ?etime) ∧
       difference(?etime - ?ctime) ≤ 60 ∧ eventLoc(?e, ?eloc) ∧
       distance(?eloc - ?cloc) ≤ 80 [0.9].
```

When issuing the following query

```
hasDestination(Mike, ?dest, 24.01/09:00, DFKI-SB)
```

thus looking for the place where Mike might want to go at 9:00am starting from the location DFKI-SB, the engine infers that Mike is associated with an event E1 at the location DFKI-KL, which is about to take place in one hour. Since the distance to this location is less than 80 kilometers, which can typically be reached within less than 1 hour, DFKI-KL is a likely destination of Mike on this morning. Hence, we obtain the derived fact

```
hasDestination(Mike, DFKI-KL, 24.01/09:00, DFKI-SB) [0.3].
```

as the only possible answer to our query with a confidence of 0.3024 (which can be obtained by multiplying all input confidences in this simple example).

Lineage & Confidence Computation. URDF employs SLD resolution, which is also the default grounding technique used in Datalog. In analogy to uncertain and probabilistic databases [1], we represent lineage of a derived fact as a Boolean formula (see also [3] for details on the semantics of these operations). The lineage formula is expanded recursively when grounding a query against the rules. That is, whenever we expand a rule, the head literal is replaced by the literals in the body of the rule, such that only variables related to base facts (and first-order soft rules) are contained in the final Boolean lineage formula.

References

1. Benjelloun, O., Sarma, A.D., Halevy, A.Y., Widom, J.: ULDBs: Databases with uncertainty and lineage. In: VLDB. pp. 953–964 (2006)
2. Feld, M., Endres, C.: Sharing User and Context Models in Automotive HMI. In: Adj. Proc. of 2nd Int. Conf. on Automotive UI and Interactive Vehicular Apps (2010)
3. Sarma, A.D., Theobald, M., Widom, J.: Exploiting lineage for confidence computation in uncertain and probabilistic databases. In: ICDE. pp. 1023–1032 (2008)
4. Sonntag, D., Theobald, M.: Explanations in dialogue systems through uncertain RDF knowledge bases. In: Workshop on Explanation-aware Computing (2010)
5. Theobald, M., Sozio, M., Suchanek, F., Nakashole, N.: URDF: Efficient reasoning in uncertain RDF knowledge bases with soft and hard rules. Tech. Rep. MPI-I-2010-5-002 (2010)