

Desperately Searching for Travel Offers? Formulate Better Queries with Some Help from Linked Data

Chun Lu^{1,2}(✉), Milan Stankovic^{1,2}, and Philippe Laublet²

¹ Sépage, 27 rue du Chemin Vert, 75011 Paris, France
{chun,milstan}@sepage.fr

² STIH, Université Paris-Sorbonne, 28 rue Serpente, 75006 Paris, France
philippe.laublet@paris-sorbonne.fr

Abstract. Various studies have reported on inefficiencies of existing travel search engines, and user frustration generated through hours of searching and browsing, often with no satisfactory results. Not only do the users fail to find the right offer in the myriad of websites, but they end up browsing through many offers that do not correspond to their criteria. The Semantic Web framework is a reasonable candidate to improve this. In this paper, we present a semantic travel offer search system named “RE-ONE (Relevance Engine-One)”. We especially highlight its ability to help users formulate better search queries. An example of a permitted query is *in Croatia at the seaside where there is Vegetarian Restaurant*. We conducted two experiments to evaluate the Query Auto-completion mechanism. The results showed that our system outperforms the Google Custom Search baseline. Queries freely conducted in RE-ONE are shown to be 63.4 % longer in terms of number of words and 27 % richer in terms of number of search criteria. RE-ONE supports better users’ query formulation process by giving suggestions in greater accordance with users’ idea flow.

Keywords: Semantic search · Travel search · Linked data · Query Auto-completion

1 Introduction

The sector of e-tourism is today in lightning growth. According to Google 2013 Traveler [16], the Web is the source of inspiration for 61 % of people and the source of a travel planning for 80 %. Search is the primary entry point to the travel-related information online. 58 % of leisure travelers and 64 % of business travelers always start their travel booking and planning process with search. However, 68 % begin searching online without having a clear travel destination in mind. Consumers spend up to 45 days and conduct 38 visits to travel sites before booking [5]. In the sample session of the study, the user switched frequently among retail sites, travel sites, weather sites, generalist search engine sites, social media sites etc. Users have to interact with vast amounts of information by doing lots of browsing and searching before finding relevant travel offers, and being able to verify if those meet their criteria. Obviously, existing systems do not support users well in expressing their needs and finding what they want

if users end up spending so much time and doing so much manual verification (on external sites) for offers took into consideration. Our motivation is to develop an efficient search system to accelerate the travel offer finding with the following hypotheses:

- By guiding the user during the query formulation process in an intelligent way, we can make the user formulate longer and richer queries, that will yield more satisfactory results in comparison to commonly used Query Auto-completion systems such as our baseline Google Custom Search.
- By leveraging Semantic Web graphs and external sources (information not present in the travel offer's presentation), we can improve the user's ability to check if a travel offer satisfies his/her criteria, thus reducing the need to crosscheck information on multiple other sites. Users would thus use novel criteria directly in the search bar so that information access becomes more direct and quicker.

The main contributions from this paper are three fold:

- a travel destination-centered data graph gathering sources from RDF databases, social media websites and web services
- a pattern-based method to verbalize the semantic data graph in a controlled language
- a Query Auto-completion mechanism to guide users during the query formulation process perceived to be in great accordance with users' idea flow.

The remainder of the paper is organized as follows. In Sect. 2, we present some related work. In Sect. 3, we present our system named RE-ONE (Relevance Engine-One). In Sect. 4, we present the conducted experiments. In Sect. 5, we conclude the paper.

2 State of the Art

In this part, we relate our system to some academic and industrial contributions.

2.1 Why Semantic Web Framework?

The traditional full-text search approach is widely used by frameworks and services such as Google Search¹, Apache Solr², Lucene³, Elasticsearch⁴ etc. This approach is not efficient enough for the travel offer search. We highlight here two main weak points.

Firstly, the spectrum of search possibilities depends on words in the stored documents and indexes mainly generated from them. If an information is not described in

¹ <https://www.google.com/>.

² <http://lucene.apache.org/solr/>.

³ <http://lucene.apache.org/core/>.

⁴ <http://www.elasticsearch.org/>.

the documents, users can difficultly find documents with it. It is often the case in the travel search context. Because a travel can be associated to so many things that they cannot all be mentioned in the documents. For example, in a travel offer, the destinations are Hyères, le Lavandou etc., which are situated in the department of Var in France. But if this information is not mentioned in the presentation, full-text search systems would not find this offer given a very legitimate query like “Var” or “in the Var” or “in the department of Var”.

Secondly, they deliver documents containing the words in the search query but not satisfying the criteria carried by the words. For example, for the query “travel in Paris”, full-text search systems would find all offers having the word “Paris” in the presentation, even if Paris is the departure city.

The Semantic Web framework is a reasonable candidate to address these two weak points for two reasons. Firstly, a big amount of data about travel destinations is available on the Web. Many of them are already structured according to Semantic Web standards and can be found on RDF databases like DBpedia. Other data can be easily structured according to the same standards. These data can be used as additional indexes to enlarge the spectrum of search possibilities. Secondly, search systems based on Semantic Web framework deliver documents satisfying the criteria carried by the words in the search query. Given the same query “travel in Paris”, with an appropriate ontology, semantic search systems like [2, 3, 6–8, 11, 15] would understand that the user is searching for offers that have Paris as destination. This accelerates the access to relevant information and saves users’ time.

For these reasons, we decided to use Semantic Web framework to try to solve the travel search problem in a new way, building our travel offer search system that we could plug into existing travel websites and provide an advanced, more intelligent, search solution.

2.2 Search Over Linked Data

Keyword search is the most popular technique for querying data with loose structure on the Web. Its success comes from the flexibility it provides to the user to retrieve information from a data source without mastering a complex query language (e.g., XQuery, SPARQL) and without knowing the structure of the data source [7]. In [3], the authors proposed a method for effective and efficient entity search over RDF data. They described an adaptation of the BM25F ranking function for RDF data. In [8], the authors presented a semantic search system that provides the user with a capability to query Semantic Web information using natural language, by means of *PowerAqua* [11] and complements the specific answers retrieved during the Question Answering process with a ranked list of documents from the Web. *PowerAqua* is a natural language interface to ontologies. Natural language interfaces are systems which allow users to express their search queries in some natural language and retrieve answers from given ontological databases. *PowerAqua* can be coupled with multiple heterogeneous RDF databases but most of such systems can only be coupled with one RDF database. These systems often face the habitability

problem that refers to how easily, naturally and effectively users can use the language to express themselves within the constraints imposed by the system [6].

FREyA [6] uses two methods for improving the habitability: feedback and clarification dialogues. The system models feedback by showing the user how the query is interpreted, thus suggesting repair through query reformulation. Clarification dialogues are used to control the query interpretations generated by the systems. *GINO* [2] allows users to edit and query ontologies in a language akin to English. It uses a small static grammar, which it dynamically extends with elements from the loaded ontologies. *AGGREGO SEARCH* [15] is similar to *GINO*. It offers a keyword-based query solution. It suggests grammatical connectors from natural languages during the query formulation step in order to specify the meaning of each keyword, thus leading to a complete and explicit definition of the intent of the search.

In the travel context, search systems face also another challenge: users do not know what to search and their queries are very short. Making the user express his/her needs in a more expressive way is a declared need. While the approaches may be well adapted for the query disambiguation or interpretation task, this need is not really addressed.

In [12], the authors presented a number of methods and their implementation in an online tool for mining type-based query context information, i.e. query prefixes and postfixes that are common to a class of entities, while uncommon to other entities outside of their class. Postulating that these context words represent aspects of entities that search engine users are interested in, they proceeded to investigate on the case of Wikipedia the extent to which this schema of information needs matches the schema of available structured data. They found that at least for the most common context words the overlap is very low as the most common queries are not specific enough to be answered by factual data. They indicated that a promising direction of research is the investigation of how search engines might assist users in formulating more precise queries. This is exactly what we are trying to achieve in travel.

We developed a Query Auto-completion mechanism. Users are assisted during the whole query formulation process. Similar to *GINO* and *AGGREGO SEARCH*, our vocabularies come from the considered RDF database. There are two main differences. The first difference is that, in those systems, vocabularies are used almost directly with no extensive adaptation to the use-case, and vocabularies used in RDF databases are not always very natural and explicit. For example, on DBpedia, the `rdfs:label` value of the property `dbpedia-owl:country` is `country`. If we use this vocabulary directly, a segment of query would be: “country France”, while the natural way to say it is: “in France”. *AGGREGO SEARCH* enriched vocabularies with additional logical connectors like “and” and some tool words like “of”. However, this enrichment does not make its use enough natural. We used a pattern-based method to verbalize our semantic data graph in a controlled language. The second difference is that the suggestions in those systems are not ranked. All possible terms are suggested in the order of appearance in the considered RDF database. Our system contains an approach to rank criteria suggestions in a way leading the user to compose queries more likely to yield satisfactory results.

2.3 Linked Data Applied in Travel Context

In [14], the authors pointed out some issues that hampered the automatic access and reuse of data sets about travel statistical indicators: (i) by them being offered as data dumps in non-semantic encodings; (ii) by them assuming some implicit knowledge that is necessary to build applications (e.g., that a city is situated in a certain country) and (iii) by the use of incompatible ways to measure the same indicator without formally specifying the assumptions behind the measurement technique. They explored the use of linked data technologies to solve these issues by triplifying the content of a broadly used data source of European tourism statistics. They built a prototype system using the data to support tourism decision makers in their activities of combining and comparing statistical indicators.

In [9], the authors presented an application for exploiting, managing and organizing Linked Data in the domain of news and blogs about travelling. The system makes use of several heterogeneous datasets to help users to plan future trips, and relies on the Open Provenance Model for modeling the provenance information of the resources. This system can be considered mostly a visualization tool. The scenario on the top of which was developed the application was related to the general context of travelling, where travelers want to share, read and reuse experiences in blogs and online news items.

These two papers demonstrated the benefits of using Linked Data in the travel domain. But we have different concerns, [14] is interested in statistical indicators, [9] in travel experience visualization, and RE-ONE in travel offer search.

2.4 Industrial Travel Search Systems

*Kayak*⁵ represents the type of tools where users need to type the exact destination to find corresponding offers. It is not effective enough in the considered scenario. On *TripTuner*,⁶ users do not search with words but by adjusting six criteria sliders. The number of criteria is very limited and the criteria are the same for all users no matter what they search for.

*Find my carrots*⁷ and *Zap Travel*⁸ claim to be semantic-based travel search engines. Users can type queries in controlled natural languages. We observed some proposed query examples and found some notable points. Some queries are too long for the contained information. For example “*What cities can I visit in Europe that are good for nightlife?*”, “*I want to go to Los Angeles, CA from Bangalore with my wife, 10 year old son and 5 year old daughter next sunday and return 10 days later*”. Users’ willingness to type such queries is uncertain and the systems do not give a sufficient support in formulating them. *Zap Travel* ignores or changes the semantics of important elements

⁵ <http://www.kayak.com/>.

⁶ <http://triptuner.com/>.

⁷ <http://www.findmycarrots.com/>.

⁸ <http://www.zaptravel.com/>.

in users' queries. For example, in "*I want to go golfing in California*", "golfing" is ignored, in "*Somewhere warm*", "warm" is interpreted as "beach". Such systems thus, generate other frustrations (such as inaccuracy of results with regards to criteria etc.) while trying to solve the travel search problem.

2.5 Other Industrial Search Systems

*Facebook Graph Search*⁹ is a semantic search system introduced by Facebook in March 2013 which allows users to search in a English-based controlled natural language within the Facebook Social Graph. Some query examples are "*Photos of my friends in New York*", "*Photos I like*", "*People who like Cycling and live in Seattle, Washinton*", "*Cities my family visited*". However, *Facebook Graph Search* does not provide any travel search feature.

Yahoo Knowledge Graph is a knowledge base used by Yahoo to enhance its search engine's results with semantic-search information gathered from a wide variety of sources.

*IBM Watson*¹⁰ is an artificially intelligent computer system capable of answering questions posed in natural language, developed in IBM's DeepQA project. The sources of information for Watson include encyclopedias, dictionaries, thesauri, newswire articles, and literary works. Watson also used databases, taxonomies, and ontologies. Specifically, DBPedia, WordNet, and Yago were used.

*Wolfram Alpha*¹¹ is a computational knowledge engine. It uses built-in knowledge curated by human experts to compute on the fly a specific answer and analysis for every query. The long-term goal is to make all systematic knowledge computable and broadly accessible.

Google Search provides a platform called *Google Custom Search* that allows web developers to feature specialized information in web searches, refine and categorize queries and create customized search engines, based on *Google Search*. Many actors in the sector of the e-tourism install *Google Custom Search* on their websites. *Google Custom Search* provides no special feature for travel-related content. It indexes from and searched in the custom website's documents. The search quality depends strongly on the documents which are different on every website.

3 System RE-ONE

In this part, we present five aspects of our system: travel destination-centered semantic data graph, data graph verbalization, travel offer catalogue annotation, Query Auto-completion, search results ranking.

⁹ <https://www.facebook.com/about/graphsearch>.

¹⁰ http://en.wikipedia.org/wiki/Watson_%28computer%29.

¹¹ <http://www.wolframalpha.com/>.

3.1 Travel Destination-Centered Semantic Data Graph

Travel destinations are very important in travel offers. However, they are not always well described in travel offers' presentations. This can limit considerably the spectrum of search possibilities. This problem can be addressed by leveraging the Semantic graph of DBpedia where travel destinations are linked to other relevant information.

In this paper, travel destinations that we consider are cities. On DBpedia, we did not find a class of which the instances are all the cities in the world. The class *dbpedia-owl:City* exists, but its instances are not complete, for example, *dbpedia:Paris* is not an instance of it. The most appropriate class is *dbpedia-owl:Settlement*, even though some instances are countries, districts or other types. So our travel destinations are instances of the class *dbpedia-owl:Settlement*. This class is linked with a big number of object properties and datatype properties. Many of them are not interesting in the travel context. For example, *dbpedia-owl:inseeCode* which links to a numerical indexing code used by the French National Institute for Statistics and Economic Studies (INSEE) to identify various entities, *foaf:homepage* which links to the homepage of something, *dbpedia-owl:subdivisions* which links to the number of subdivisions etc. Some of them do not have clear semantics. For example, *dbpprop:alt*, *dbpprop:align* etc. We did a manual selection of properties which might be interesting. So far, the selected properties are the following: *dbpedia-owl:country*, *rdfs:label*, *dbpedia-owl:location*, *dcterms:subject*. We isolated the selected classes and properties and we created a travel destination-centered semantic data graph.

The information that we retrieved from DBpedia can only cover a part of users' search needs. As [5] shows, users consult diverse types of websites to gather information when they do travel search. To simplify users' search task, we included in our data graph these types of information that are now manually crosschecked by users on multiple sites. They are: points of interest, coastal or not, and weather.

Foursquare has a very rich taxonomy of points of interest categories¹². We did a manual selection of categories that are relevant to travel search and representative to travel destinations. We did not consider categories like "Professional", "Office", "School", "Residence" etc. Via its API, we retrieved for each of our destination, if it exists in *Foursquare*, the points of interest that belong to the selected categories and are located in it. As to if a travel destination is coastal or not, we calculated this information with data retrieved via the API of *Bing Maps*¹³. We retrieved monthly average temperatures of our travel destinations via the API of *World Weather Online*¹⁴.

Thus, we created a travel destination-centered semantic data graph gathering sources from DBpedia, social media websites and web services. Figure 1 shows its structure.

¹² <https://developer.foursquare.com/categorytree>.

¹³ <http://msdn.microsoft.com/en-us/library/dd877180.aspx>.

¹⁴ <http://www.worldweatheronline.com/>.

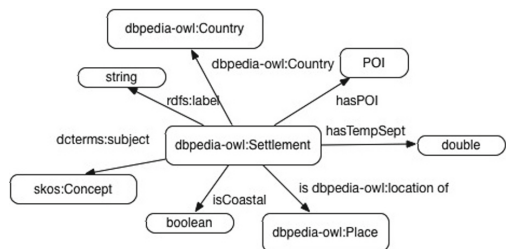


Fig. 1. Main structure of the travel destination-centered data graph

3.2 Data Graph Verbalization

We used a pattern-based method to verbalize the (property, object) couples of the triples in the data graph. Here are the patterns.

(Property, object) couples	Verbalization pattern
dbpedia-owl:country, dbpedia-owl:Country	in + country’s label
is dbpedia-owl:location of, dbpedia-owl:Place	where is located + place’s label
rdfs:label, datatype value	in + value
dcterms:subject, skos:Concept	Concept’s label
hasTempSept, datatype value	where it is hot/moderate/cold in September
hasPOI, POI	where there is + POI’s category
isCoastal, datatype value	at the seaside

After the verbalization process, each instance of the class *dbpedia-owl:Settlement* is associated with a certain number of semantic tags written in a controlled language. Figure 2 is a result excerpt for the city of Nice.

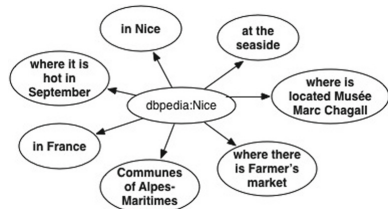


Fig. 2. Example illustrating the pattern-based data graph verbalization

3.3 Travel Offer Catalogue Annotation

Given a travel offer catalogue, an annotation task is conducted. An offer initially retrieved from a particular travel catalogue, is often a structured data entry containing travel destinations, dates, prices as structured data, and a URL containing textual and multimedia information. Offers are processed individually. Only two elements are considered and analysed: the travel destinations and the URL.

Each offer is first assigned all semantic tags that are associated with its travel destinations. Then we used a Named Entity extractor called “Dandelion” [13] to

analyze the textual content in the URL and to extract DBpedia concepts. The semantic tags and the labels of extracted concepts constitute together the offer's indexes. As we can see in Fig. 3 below, each offer of the catalogue is annotated with a certain number of indexes. Indexes can be unique to one offer or common to several offers (index 3).

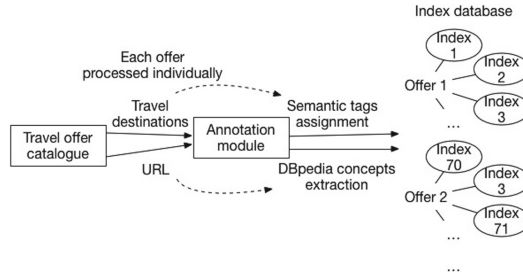


Fig. 3. Travel offer catalogue annotation workflow.

3.4 Query Auto-Completion

In [1], we can find a short explanation of the autocomplete algorithm of *Google Search*: autocomplete predictions are automatically generated by an algorithm without any human involvement, based on a number of objective factors, including how often past users have searched for a term. The algorithm automatically detects and excludes a small set of search terms. But it is designed to reflect the diversity of our users' searches and content on the web. So just like the web, the search terms shown may seem strange or surprising. Except for this, we did not have access to a more detailed explanation about their algorithm.

Our Query Auto-completion mechanism is different. In order to maximize the chances of satisfactory searches, our approach is not based on how often past users have searched for a term but on how a criterion is relevant to find offers corresponding to it. We define a criterion as an attribute of an offer allowing direct verification whether the offer has it or not. A criterion can contain one or several words. A criterion can be expressed in different forms: keywords or in a natural language, for example, "seafood restaurant", "where there is a seafood restaurant", "where I can find a seafood restaurant", "with a seafood restaurant" are all different forms of the same criterion.

Indexes generated in Sect. 3.3 can be considered as criteria. Knowing all criteria that allow to find offers in a catalogue, we can calculate, for each criterion, the number of offers that can be found with it. Criteria are then ranked by this number in descending order. At the beginning of the search session, users are suggested 8 criteria which are the best ranked criteria of each type of information (data graph & extracted DBpedia concept) (Fig. 4).

It is important to note that the systems reinforces the diversity of suggestion types so that the users can, from the very start, get to know broadly what are the types of search criteria that can be searched. The users are subtly informed of the constraints imposed by the system.

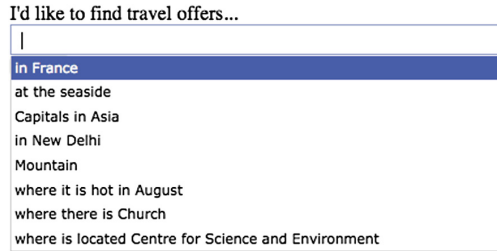


Fig. 4. Screenshot of the system at the beginning of the search session.

The system is implemented using jQuery Tokeninput¹⁵. The interaction between the system and users are controlled. Users can only type criteria that exist in the index database. Users have to validate a criterion before typing another one. The suggestions are generated in two different modes. The first mode is active while the user is typing a word. The suggestions are in this case calculated with regards to the highest string similarity with the text being typed by the user. We used the Levenshtein distance as our string metric. The other mode is active when one or more criteria are validated and the user is idle, then the system proposes additional criteria suggestions while taking into account the criteria already validated. In the two modes, the ranking of suggestions is performed in a way that maximizes the probability of performing a successful query on a given website catalogue, and that favors the diversity of proposed criteria types.

$$\text{rank}(c, Q) = p(''c + Q'') \tag{1}$$

The system searches in the offers that correspond to the validated criteria (remaining offers) all available criteria (that are not yet validated). Available criteria are ranked according to the formula (1). The rank of a criterion c (for instance “in Slovenia”), provided that c is one of the available criteria, not included in the validated ones that constitute the current query Q , is proportional to the probability of finding an offer in the catalogue that corresponds to the extended query: $Q + c$, in which criterion c is added to the already validated criteria. An essential part of the process is making sure that proposed criteria are diverse: a step made possible by the use of semantic data graphs in which offer properties are typed in a hierarchical way. We display one criterion (the best-ranked) from each type of information as much as possible (if the type is still present in the available criteria. For example, “at the seaside” is the only criterion of its type, if it is validated, it will not be suggested again.), while preserving the order of appearance in order of the best ranked ones.

3.5 Search Results Ranking

We did not implement a specific search results ranking method. Because in our case, all retrieved results correspond to the query and are of the same importance. For example,

¹⁵ <http://loopj.com/jquery-tokeninput/>.

for the query “in Croatia at the seaside”, all found offers are in Croatia and at the seaside. When a search query is submitted, the system searches for offers that correspond to all the criteria. Search results are now ranked by the profitability index defined by catalogue providers.

4 Evaluation

In this section, we describe two experiments conducted to test our research hypotheses.

4.1 Experiment Data

Our experiments are conducted using the travel offer catalogue provided by a French tour operator. The catalogue counts 956 travel offers. The travel destinations are numerous and cover more than 150 countries.

4.2 Baseline

The baseline that we compare with is the Google Custom Search installed on the website of the French tour-operator. This tool is a good candidate for baseline for two reasons. Firstly, it is the personalized version of the most used search engine in the United States [3] and it is widely used by travel websites. Secondly, we have full access to the catalogue, this allowed us to compare the two systems with the same data.

4.3 Metrics

The following metrics have been used:

- *average number of words per query*
- *average number of criteria per query*
- *provenance of criteria.*

We used the *average number of words per query* because the travel website owners told us about the limits they encounter with short queries in terms of providing personalized and optimal user experience. But it is an objective measure as there is no human judgment involved, the fact that different systems propose criteria in different forms, and according to different algorithms, the difference observed in the number of words is not sufficient to conclude that one or the other system provides better support in query formulation process. For this reason, we were interested in the *average number of criteria per query*, which should not be influenced by the form in which the two systems present their suggestions, but only by their capacity to suggest a criterion that the user would actually consider adding to his/her query. In addition to these metrics, we were interested in the qualification of criteria used, in terms of their provenance (user thought-of, or suggested by the system), expressed on the following scale from option 1 (criterion formulated by the user and not suggested by the system) to option 5 (criterion suggested by the system that user hasn't thought of himself).

1. *It is a criterion that I already had in mind and it was not in the suggestions.*
2. *It is a criterion that I already had in mind and it was also in the suggestions.*
3. *It is a criterion that I did not have in mind. It was in the suggestions. It is relevant. But I could have thought about it.*
4. *It is a criterion that I did not have in mind. It was in the suggestions. It is relevant. It is surprising and I could not have thought about it.*
5. *It is a criterion that I did not have in mind. Some suggestions inspired me and helped me find this criterion.*

4.4 Two Conducted Experiments

The website of the tour-operator is only in French. The Google Custom Search is also performed in French. We developed a French version of RE-ONE. Vocabularies and the pattern-based verbalization method are translated and adapted to French. We asked 34 people to participate in the evaluation. They are all French citizens or French speakers. They have between 23 and 35 years old. They are used to doing travel search on the Web.

The duration of the offers in the test catalogue is very varied, from 1 day to 71 days. We are thus exposed to the risk of people manifesting different behavior and using different criteria for long and short trip searches. In order to avoid bias, we thus divided our users into two groups: A to whom we gave the task of searching for short (1–4 days) trips that correspond to (extended) weekend trips and B to whom we gave the task of searching for long (5 and more days) trips that correspond to longer holidays. Both groups used both our system and the baseline system.

Firstly, participants put themselves into the scenario of searching for the next travel. They identify clearly a concrete possible next occasion for travel in their agendas, and imagine the concrete usual context of looking for places to go on that occasion. At this very early stage they do not yet have a precise destination. They turn to the search systems to find offers. Secondly, Participants performed their search twice, once in RE-ONE and once in the baseline system. To avoid possible bias due to the order of use of the systems, we made half of the users search in RE-ONE first and the other half search in the baseline system first. Thirdly, participants splitted the queries into criteria and chose for each criterion the most suitable option from the scale introduced in Sect. 4.3. They are well informed of our definition of criterion introduced in Sect. 3.4 and are given some splitting examples. In spite of this process, this task remains a potential source of subjectivity or error. To avoid bias, we ran a verification phase after the experiment, in which 3 independent reviewers were asked to verify, for each query-to-criteria split performed by the users in the study, if the split was performed according to our definition. Only splits that 2 or 3 reviewers approved were taken into account in further measurement. Finally this review phase required the elimination of only 2 queries.

4.5 First Experiment: Searching for a Short Trip

In the first group which performed a search for a short trip, a significant difference in average word number was observed between the systems: 2.06 ± 1.14 in the baseline

system and 5.76 ± 3.46 in RE-ONE. The difference was significant according to T-test ($p\text{-value} < 0.001$). A smaller, but still significant ($p\text{-value} < 0.05$) difference was observed in the number of criteria: 1.59 ± 0.78 in the baseline system and 2.12 ± 0.8 in RE-ONE. Both metrics are consistent and allow us to conclude about the positive impact of our system in terms of assistance in query formulation; queries conducted in RE-ONE were in average 64.2 % longer and 25 % richer than that in the baseline system. The following table is a part of users' queries for the short trip search that are translated literally by us from French to English. The same user carried out the two queries of each row (Fig. 5).

RE-ONE	The baseline system
Church New art	Church
Northern Europe North Cape	Europe
Cross-country skiing in Coroico	Cross-country skiing France
Volcanic island Porto-Novo Ribera Grande	Island of Malta

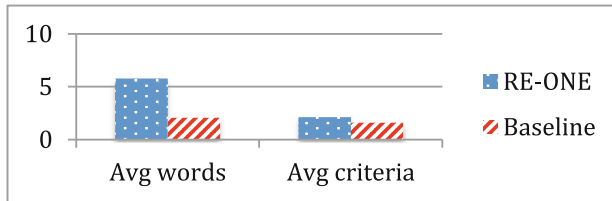


Fig. 5. Average number of words and criteria per query statistics in the first experiment.

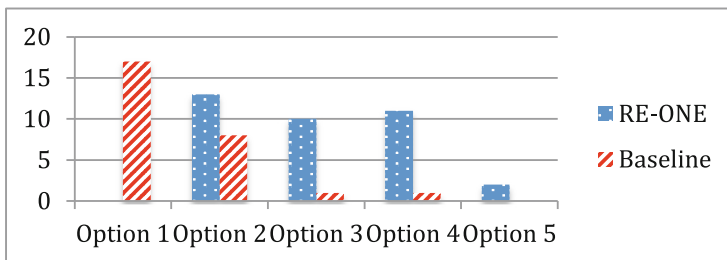


Fig. 6. Provenance of criteria statistics in the first experiment.

Figure 6 allows us to observe the difference in criteria provenance, on our scale option 1–option 5, options closer to one being closer to user's own ideas, and options closer to 5 being closer criteria attributable to the systems' assistance. We see that the distribution of criteria provenance for RE-ONE is stronger in options attributing the presence of criteria in final queries to the system's assistance, while queries formulated in the baseline system are richer in criteria thought-of by the user.

4.6 Second Experiment: Search for a Long Trip

Similarly to the first experiment, in the case of long trip search, a significant (t-test p -value < 0.05) was observed in the average number of words per query 2.65 ± 1.37 for the baseline system and 7.06 ± 5.4 for RE-ONE. For the number of criteria, again a slightly smaller but significant (p -value < 0.05) difference was created: 1.59 ± 0.62 for the baseline system and 2.24 ± 0.9 for RE-ONE. In other words, queries conducted in RE-ONE were in average 62.5 % longer and 29 % richer than those in the baseline system. The following table is a part of users' queries for the long trip search (Fig. 7).

RE-ONE	The baseline system
Seafood Culture Salvador Dali	Seafood
Summer where there is a historical site	Summer historical
South-East Asia Hô-Chi-Minh City	South-East Asia travel
Trek in New Zealand	Linguistic stay

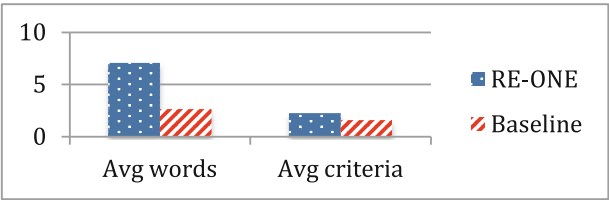


Fig. 7. Average number of words and criteria per query statistics in the second experiment.

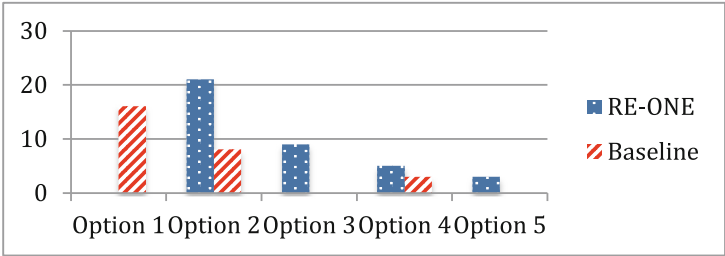


Fig. 8. Provenance of criteria in the second experiment

Looking at the provenance of criteria, again we can conclude the superiority of RE-ONE over the baseline in terms of the presence of criteria attributable to the system's assistance (Fig. 8).

4.7 General Discussion

In both experiments, RE-ONE outperforms the the baseline system baseline, and that according to all metrics that we used. In average, queries conducted in RE-ONE are

63.4 % longer and 27 % richer than that in the baseline system. As regards to supporting users' query formulation process, RE-ONE also shows a clear advantage.

While the interaction seems smoother in the baseline system as users can type whatever they like, if the user starts typing a term that is not suggested, then the baseline system is not able to do suggestions afterwards. This explains the big number of criteria that correspond to option 1. Since RE-ONE is a controlled system, users can only type search criteria that exist in our known indexes. This is why there is 0 criterion corresponding to option 1. We were interested in the effect of this constraint. We examined the logs of the experiments, we calculated the number of times that users could not type what he/she had in mind and then typed another criterion. There were in total 8 times for the short trip search and 11 times for the long trip search. The number is much smaller than the baseline in both experiments. The results for the other four options show that the suggestions of RE-ONE are always in greater accordance with users' idea flow, and able to suggest criteria of added value with regards to users own ideas, which end up being accepted and constituting the final queries. It is especially when looking at options qualifying the criteria as external to what the user had in mind (options 3–5) that RE-ONE shows great superiority with regards to the baseline.

We have thus demonstrated the ability of our system, combining travel offers with external information from semantic graphs to provide effective assistance in travel search query formulation.

5 Conclusions

In this paper, we presented RE-ONE, a semantic travel offer search system. We highlighted its ability to help users formulate better queries. This is made possible by leveraging the semantic graph containing sources from RDF databases, social media websites and web services. As for the evaluation, 34 people participated in two experiments, half of them simulated a short trip search, the other half a long trip search. The data is the travel offer catalogue provided by a French tour-operator. Our system outperforms the Google Custom Search baseline in both experiments leading users to formulate longer and richer queries. We provide better support by giving relevant (accepted by the users) suggestions in great accordance with users' idea flow.

In spite of good evaluation results, we still find some weak points to be improved in a future work. Firstly, we can enrich the data graph with other properties on DBpedia or other external sources. Secondly, the quality of DBpedia concepts extraction is not good enough. We can find some strange concepts like "A40 autoroute". Thirdly, some concepts are well extracted but not relevant to travel like "Laborer". We need to find a method to remove automatically concepts that are poorly extracted or not relevant. Fourthly, we can reduce the lexical constraints and provide more flexibility to users. Fifthly, we can develop a personalization module to analyze users' search behavior and to do some adaptations.

In today's technology-rich world, users are confronted with vast amounts of data and spend much effort interacting with them. As we showed in this paper, we are convinced that Semantic Web technologies can provide a solid support in users' query formulation process.

References

1. Autocomplete. <https://support.google.com/websearch/answer/106230?hl=en>. Accessed 4 Nov 2014
2. Bernstein, A., Kaufmann, E.: GINO - a guided input natural language ontology editor. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 144–157. Springer, Heidelberg (2006)
3. Blanco, R., Mika, P., Vigna, S.: Effective and efficient entity search in RDF data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 83–97. Springer, Heidelberg (2011)
4. comScore Releases October 2014 U.S. Desktop Search Engine Rankings. <http://www.comscore.com/press/Insights/Market-Rankings/comScore-Releases-October-2014-US-Desktop-Search-Engine-Rankings>. Accessed 29 Dec 2014
5. Custom Research: Exploring the Traveler's Path to Purchase. <http://info.advertising.expedia.com/path-to-purchase>. Accessed 28 Aug 2014
6. Damjanović, D., Agatonović, M., Cunningham, H., Bontcheva, K.: Improving habitability of natural language interfaces for querying ontologies with feedback and clarification dialogues. *Web Semant. Sci. Serv. Agents World Wide Web* **19**, 1–21 (2013)
7. Dass, A., Aksoy, C., Dimitriou, A., Theodoratos, D.: Exploiting semantic result clustering to support keyword search on linked data. In: Benatallah, B., Bestavros, A., Manolopoulos, Y., Vakali, A., Zhang, Y. (eds.) WISE 2014, Part I. LNCS, vol. 8786, pp. 448–463. Springer, Heidelberg (2014)
8. Fernandez, M., Lopez, V., Sabou, M., Uren, V., Vallet, D., Motta, E., Castells, P.: Semantic search meets the web. In: *Proceedings of the IEEE International Conference on Semantic Computing*, pp. 253–260 (2008)
9. Garijo, D., Villazón-Terrazas, B., Corcho, Ó.: A provenance-aware linked data application for trip management and organization. In: *I-SEMANTICS 2011*, pp. 224–226
10. Introducing Graph Search. <https://www.facebook.com/about/graphsearch>. Accessed 4 Apr 2014
11. Lopez, V., Fernández, M., Motta, E., Stielor, N.: PowerAqua: supporting users in querying and exploring the semantic web. *Semant. Web* **3**(3), 249–265 (2012)
12. Meij, E., Mika, P., Zaragoza, H.: Investigating the demand side of semantic search through query log analysis. In: *Proceedings of the Workshop on Semantic Search at the 18th International World Wide Web Conference* (2009)
13. Parmesan, S., Scaiella, U., Barbera, M., Tarasova, T.: Dandelion: from raw data to dataGEMs for developers. In: *Proceedings of the ISWC Developers Workshop 2014*, Riva del Garda, Italy, pp. 1–6, October 2014
14. Sabou, M., Brasoveanu, A., Arsal, I.: Supporting tourism decision making with linked data. In: *I-SEMANTICS 2012*, pp. 201–204
15. Smits, G., Pivert, O., Jaudoin, H., Paulus, F.: AGGREGO SEARCH: interactive keyword query construction. In: *EDBT*, pp. 636–639 (2014)
16. The 2013 Traveler. <http://www.thinkwithgoogle.com/research-studies/2013-traveler.html>. Accessed 28 Aug 2014