

On Finding the Relevant User Reviews for Advancing Conversational Faceted Search

Eleftherios Dimitrakis^{1,2}, Konstantinos Sgontzos^{1,2}, Panagiotis Papadakos¹,
Yannis Marketakis¹, Alexandros Papangelis³,
Yannis Stylianou^{2,3}, and Yannis Tzitzikas^{1,2}

¹ Institute of Computer Science - FORTH-ICS, Greece

{dimitrakis, sgontzos, papadako, marketak, tzitzik}@ics.forth.gr

² Computer Science Department - University of Crete, Greece

³ Speech Technology Group - Toshiba Research Europe

{alex.papangelis, yannis.stylianou}@crl.toshiba.co.uk

Abstract. Faceted Search (FS) is a widely used exploratory search paradigm which is commonly applied over multidimensional or graph data. However sometimes the structured data are not sufficient for answering a user’s query. User comments (or reviews) is a valuable source of information that could be exploited in such cases for aiding the user to explore the information space and to decide what options suits him/her better (either through question answering or query-oriented sentiment analysis). To this end in this paper we introduce and comparatively evaluate methods for locating the more relevant user comments that are related with the user’s focus in the context of a conversational faceted search system. Specifically we introduce a dictionary-based method, a word embedding-based method, and one combination of them. The analysis and the experimental results showed that the combined method outperforms the other methods, without significantly affecting the overall response time.

1 Introduction

Faceted Search (FS) is a widely used exploratory search paradigm. It is used whenever the user wants to find the desired item from a list of items (either products, hotels, restaurants, publications, etc). Typically FS offers exploratory search over multidimensional or graph data. However sometimes the structured data are not enough for answering a user’s query. User comments (or reviews) is a valuable source of information that could be exploited in such cases for aiding the user to explore the information space and to decide what options suits him/her better. Indeed, user comments/reviews are available in various applications of faceted search, e.g. for hotel booking and in product catalogs.

Enabling the interaction of FS though spoken dialogue, is appropriate for situations where the user cannot (or is not convenient to) use his hands or eyes. In such cases, the user interacts using his voice and provides commands or poses questions. If a question cannot be translated to a query over the structured

resources of the dataset, then the system cannot deliver any answer. In such cases it is reasonable to resort to the available unstructured data, i.e. to users’ comments and reviews. Figure 1 illustrates the context. The objective is not to provide the user with a direct answer but first to identify which of the user comments are relevant to the user’s question. Direct query answering is reasonable only in cases where, there is a single and credible source of unstructured data (e.g. wikipedia). This is not the case with user comments since they can be numerous, and their content can be conflicting. If we manage to find the relevant comments, then the system could either read these comments to the user, or attempt to apply question answering if the user requests so, or any other kind of analysis, e.g. sentiment analysis as in [2, 14]. In any case spoken dialogue interaction, poses increased requirements on quality, since the system should not “read” irrelevant comments as reading costs user time.

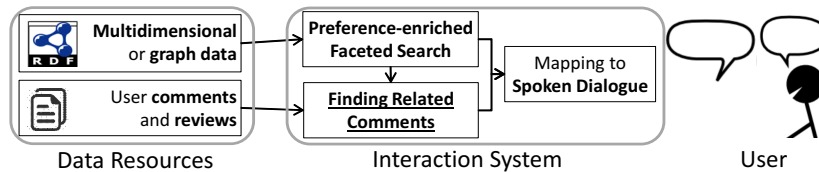


Fig. 1: Finding Related Comments and Conversational Faceted Search

Note that instead of analyzing the user comments for estimating whether a hotel is good or bad as a whole, the interaction that we propose enables the user to get information about the particular aspects or topics that are important for him, e.g. about noise, cleanliness, the quality of the wifi, parking, courtesy and helpfulness of staff, etc. The set of such topics is practically endless and we cannot make the assumption that structured data will exist for all such topics. Therefore, it is beneficial to have systems that are able to exploit associated unstructured data, e.g. user comments and reviews. The problem is challenging because user comments are usually short, meaning that it is hard to achieve an acceptable level of recall. In this paper we focus on this problem, and we introduce methods relying on hand crafted and statistical dictionaries for identifying the relevant comments. In addition we describe an evaluation collection that we have created for comparatively evaluating the introduced methods, as well as an ongoing application and evaluation over a bigger and real dataset. In a nutshell, the key contributions of this paper are: (a) we show how the FS interaction can be extended for exploiting unstructured data in the form of user comments and reviews, and (b) we introduce and comparatively evaluate four methods for identifying the more relevant user comments in datasets related to the task of hotel booking. The rest of this paper is organized as follows: Section 2 presents the required background and related work. Section 3 describes the proposed methods. Section 4 reports experimental results. Finally, Section 5 concludes the paper and discusses directions for future research and work.

2 Background and Related Work

2.1 Background: Faceted Search and PFS

Faceted search is the de-facto standard in e-commerce and tourism services. It is an interaction framework based on a multi-dimensional classification of data objects, allowing users to browse and explore the information space in a guided, yet unconstrained way through a simple visual interface [15]. Features of this framework include: (a) display of current results in multiple categorization schemes (called facets, or dimensions, or just attributes), (b) display of facets and values leading to non-empty results only, (c) display of the count information for each value (i.e. the number of results the user will get by selecting that value), and (d) ability to refine the focus gradually, i.e. it is a session-based interaction paradigm in contrast to the stateless query-and-response dialogue of most search systems. Faceted search is currently the de facto standard in e-commerce (e.g. eBay, booking.com), and its popularity and adoption is increasing. It has been proposed and applied for web searching, for semantically enriching web search results, for patent-search, as well as for exploring RDF and Linked Data (e.g. see [4,16], as well as [19] for a recent survey). The enrichment of faceted search with *preferences*, hereafter *Preference-enriched Faceted Search*, for short PFS, was proposed in [12,20]. PFS offers actions that allow the user to order facets, values, and objects using *best*, *worst*, *prefer to* actions (i.e. relative preferences), *around to* actions (over a specific value), or actions that order them lexicographically, or based on their values or count values. Furthermore, the user is able to *compose* object related preference actions, using *Priority*, *Pareto*, *Pareto Optimal* (i.e. skyline) and other. The distinctive features of PFS is that it allows expressing preferences over attributes, whose values can be hierarchically organized (and/or multi-valued), it support preference inheritance, and it offers scope-based rules for resolving automatically the conflicts that may arise. As a result the user is able to restrict his current focus by using the faceted interaction scheme (hard restrictions) that lead to non-empty results, and rank the objects of his focus according to the expressed preferences. Recently, PFS has been used in various domains, e.g. for offering a flexible process for the identification of fish species [17], as a Voting Advice Application [18], as well as, for data that contain also geographical information [6].

2.2 Related Works

Conversational Faceted Search Only a few works exist that involve speech interfaces on top of the faceted search paradigm: [3] exploits a speech user interface over facets that index audio metadata associated with audio content (that system is used for the Spoken Web, an alternative to WWW based on audio content, and the associated Mediaeval Spoken Web Search Task), while a faceted browser over Linked Data is described in [7], where commands in natural language are translated to SPARQL queries. To the best of our knowledge though, the only work that combines spoken dialogue systems with faceted search is the

one presented in [13], where the described LD-SDS system is limited to spoken dialogues over structured datasets (expressed in RDF). In this work we extend conversational faceted search for exploiting also available unstructured data (e.g. user reviews). Note that, tackling the same problem using only a single large-scale source of unstructured data, e.g. Wikipedia (as described in [1]), is much easier since in that case we do not have the source selection problem (selection of user comments in our case), and the source contains many and long texts, therefore it is not difficult to achieve a good recall level.

Similar Tasks Two similar tasks, as regards the text size, from the area of Question Answering are: (1) *Machine Comprehension (MC)* which aims at identifying the answer boundaries from a given text passage and an input question (e.g. [1] performs MC over Wikipedia), and (2) *Answer Sentence Selection* which aims at identifying the right sentence from a list of candidate sentences, given an input question (e.g. as in [21]).

3 The Proposed Approach

In §3.1 we describe an extension of the interaction of PFS for exploiting also associated unstructured data, and in §3.2 we focus on the problem of finding the relevant comments.

3.1 The Interaction

The user interacts with the system using actions corresponding to PFS actions, i.e. actions that correspond either to hard constraints (i.e. filters), or soft constraints (i.e. preferences). We shall use the term *ofocus* to refer to the restricted set of objects (those after applying all filters), and *pfocus* to refer to the first bucket of the focus, that contains the more preferred objects. If the cardinality of either of the above sets is below a configurable threshold θ (say 10), then if the user’s questions cannot be answered by the structured dataset, the system resorts to the user comments for this. Note that if at some point in the interaction, the user’s focus is big (i.e. $\min(|ofocus|, |pfocus|) > \theta$) and the user asks a question that cannot be answered by the structured dataset, then the system suggests the user to “first refine the focus” in the sense that it is not useful to ask questions of the form “quiet hotel in Rome”, or “hotels with fast wifi in London”. In other words, we could say that the system enters this mode in the so-called “End Game” phase of faceted search [15]. This choice has several benefits:

- (a) *Applicability*: It can be applied without requiring the comments to be indexed a priori, and this enables the application of this model over RSS feeds and blog comment hosting services (e.g. Disqus).
- (b) *Efficiency*: Since the analysis will be done only for the comments of the hotels in the focus, it is feasible to make this analysis at real time.
- (c) *Less Noise, Better Quality*: For the same reason, as in (b), the quality of the retrieved comments is expected to be higher in comparison to the quality of retrieval over the entire set of comments (of all hotels).

3.2 Finding the Relevant Comments

We shall use a scoring function for estimating the relevance between an input question q and each user review r_i , where $1 \leq i \leq \theta$. Below we introduce four scoring methods: (I) a Baseline, (II) a WordNet-based, (III) a Word2vec-based, and (IV) a combination of (II) and (III).

WordNet [11] is lexical database for the English language comprising 166,000 (f, s) pairs, where f is a word-form and s the set of words that have the same sense, that also includes relations between words and senses (like Synonymy, Antonymy, Hypernymy etc.). *Word2Vec* [10] is a method for transforming individual words into vectors of low dimensionality (it is low in comparison a $|words|$ -dimensionality), e.g. 300, so that their distances reveal their semantic association (these representations are derived by training a two-layer neural network). The motivation for the selection of the above methods is their ability to capture semantically relevant reviews beyond the trivial task of exact string matching, and their rich and domain-agnostic vocabulary.

The process for identifying the more relevant comments, in any of the four I-IV methods, consists of the following steps:

- 1) For each review r_i we split its text into individual sentences and get a set of sentences r_{ij} , where $1 \leq j \leq s$ and s is the number of sentences in each review. In this way, we can score the reviews based on the maximal scored sentence.
- 2) Apply tokenization, removal of stop-words and punctuations, as well as lemmatization (using Stanford CoreNLP [8]) both to the input question q and each associated sentence r_{ij} of r_i . Let denote the result by q_words and r_{ij}_words respectively.
- 3) Construct the method-related representation of q and each r_{ij} (it will be described below).
- 4) Score and rank each review based on the defined relevancy formula.

Below we describe the representation and the scoring formula for each method.

I) **Baseline:** Here we just compute the maximum Jaccard Similarity between the q_words and the corresponding r_{ij}_words sets:

$$S(q, r_i) = \max_{\forall r_{ij} \in r_i} JaccardSim(q_words, r_{ij}_words)$$

II) **WordNet:** In this method we construct WordNet-based representations for the q and r_{ij} sets. Specifically, for each word in q_words and r_{ij}_words we take the union of the synonyms, antonyms and hypernyms, denoted by $wordNet(q)$ and $wordNet(r_{ij})$ respectively, as extracted from the WordNet. The final score is defined again using the maximum Jaccard Similarity as:

$$S(q, r_i) = \max_{\forall r_{ij} \in r_i} WNS(q, r_{ij}),$$

where $WNS(q, r_{ij}) = JaccardSim(wordNet(q), wordNet(r_{ij}))$.

III) **Word2vec:** This method exploits the word2vec embeddings available in the GoogleNews 300-dimensional pre-trained model⁴. Specifically, we get the

⁴ <https://code.google.com/archive/p/word2vec/>

word2vec vector representations of all words in q_words and r_{ij}_words , denoted by $word2vec(q)$ and $word2vec(r_{ij})$ respectively. Then we apply the Word Movers Distance (WMD) [5] which calculates the minimum distance (in the vector space) between the embedded words of the two sets. The score is defined as:

$S(q, r_i) = \max_{\forall r_{ij} \in r_i} WMS(q, r_{ij})$,
 where $WMS(q, r_{ij}) = 1 - WMD_n(word2vec(q), word2vec(r_{ij}))$ and WMD_n denotes the normalized distance calculated by the division with the max WMD over all comments.

IV) **WordNet and Word2vec:** Here we combine the two previous methods through a weighted sum, reaching to the following definition of score:

$$S(q, r_i) = w_{wN} * \max_{\forall r_{ij} \in r_i} WNS(q, r_{ij}) + w_{w2v} * \max_{\forall r_{ij} \in r_i} WMS(q, r_{ij})$$

where $w_{wN}, w_{w2v} \in [0, 1]$ and $w_{wN} + w_{w2v} = 1$.

4 Evaluation

4.1 Evaluation over the Collection FRUCE

We constructed a small evaluation collection in order to compare the presented methods. The collection consists of 40 hand crafted user reviews/comments related to hotels (c_1, \dots, c_{40}) and 2 manually crafted queries (q_1 and q_2) related to the topic of *noise*. The complete list of comments is web accessible⁵ and the queries are the following: $q_1 =$ “Has anyone reported a problem about noise?”, $q_2 =$ “Is this hotel quiet?”.

For the needs of the evaluation we manually judged the relevance of the collection’s reviews to each query. Specifically, each review c_i is labeled with 1 if it is relevant, and with 0 otherwise. The relevant/irrelevant ratio in the collection is 1/3.

Quality. We measured the *mean R – Precision* and *mean AveP* over the two queries q_1 and q_2 for all methods. Specifically, for the IV method we computed various weights combinations and chose the model that achieved the highest *mean AveP*. Note that methods II and III correspond to the pairs ($w_{wN} = 1.0, w_{w2v} = 0.0$) and ($w_{wN} = 0.0, w_{w2v} = 1.0$) respectively. In our case the maximizing weights were found to be $w_{wN} = 0.7$ and $w_{w2v} = 0.3$ with *mean AveP* = 0.569 and *mean R – Precision* = 0.649. The corresponding scores for method II were *mean AveP* = 0.398 and *mean R – Precision* = 0.449, while method III achieved *mean AveP* = 0.366 and *mean R – Precision* = 0.4 (the precision of Word2vec-based methods in analogous challenges [9] is around 55%, i.e. similar to what we measured in our setting). Finally, IV outperforms both II, III, while II slightly outpoints III. As expected, all of the above models outperformed our baseline (*mean AveP* = 0.05 and *mean R – Precision* = 0.05) as shown in Table 1. We have to stress though that the results of methods II

⁵ at <http://www.ics.forth.gr/is1/sar/resources/dataset/fruce>

and IV could be further improved by combining other thesaurus with WordNet or an updated version of WordNet, since WordNet currently fails to provide the synonyms, hypernyms and antonyms of many words. Further, since we currently consider all possible senses of a word in the WordNet based approach, we might be introducing wrong terms in the $wordNet(q)$ and $wordNet(r_{ij})$ set. This problem can possibly be avoided with proper sense identification methods.

Method	Mean AveP	Mean R-Precision
I	0.05	0.05
II	0.398	0.449
III	0.366	0.4
IV	0.569	0.649

Table 1: Mean Average Precision of methods I-IV.

Method	Total time (ms)	Aver. time (ms)
I	141	3
II	797	19
III	47	1
IV	546	13

Table 2: Time for computing the score of 40 reviews for each method.

In addition, we plot a 2D diagram for each of the three models II, III, IV (baseline excluded), where the y -axis represents the computed score for (r_i, q_i) and the x -axis indicates its true binary relevance. The plots are shown in Figure 2. We can observe that the points are not separable by a threshold in any of the figures (parallel line to x -axis). However, it is obvious that the IV approach clearly improves the separation, preserving higher scores to the true relevant reviews, like III, and lower scores to the non-relevant ones, like II.

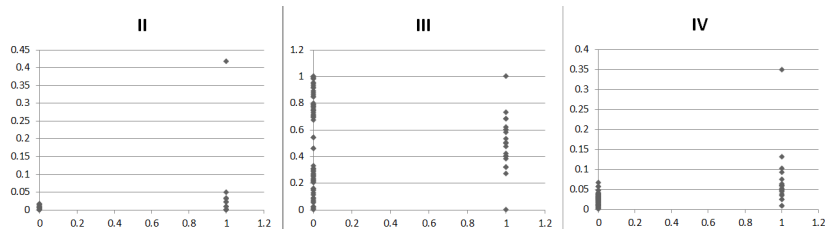


Fig. 2: Distribution of query-review pairs as a function of their calculated (floating point) and true binary relevance score for methods II, III, IV.

Efficiency. All experiments were performed using a 16GB RAM machine. Regarding speed efficiency, it is worth measuring: a) the required time for loading the appropriate resources (Dataset, WordNet, Word2Vec) (i.e. *Init Time*), and b) the required time for computing the similarity score of one query-review pair (i.e. *Execution Time*). Note that the *Init Time* cost has to be paid only once, while *Execution Time* affects the user interaction.

Regarding *Init Time*, the most time consuming resource is Word2Vec due to its enormous size (491,061 ms), followed by the loading of the FRUCE dataset (39,149 ms). The WordNet dictionary loads almost instantly (63ms). The *Execution Time* on the other hand is very fast for all methods (only 13 ms on average). We only need about 1.5 seconds for analyzing and scoring 100 reviews

with 15 words on average. Table 2 shows the *Execution Time* for computing the scores of the 40 reviews for all methods (the minimum values are in bold).

4.2 Experiments over a Real Dataset

We also evaluated the proposed methods over a real dataset, that we scrapped from a travel website. This specific dataset contains information about 382 different hotels located in 4 different cities (Kyoto, Tokyo, Osaka, Kobe) of Japan. The extracted data are logically structured in facets so that they can be directly plugged into the system, containing the following types of information: (a) boolean values, used for describing the facilities of a hotel (e.g. free of charge wifi, free parking, etc.), (b) numerical values (integers or floats) for describing quantitative values (e.g. price, review rating, distance from various points of interest, etc.), (c) geographic values for describing the location and (d) textual values. In the last category there are also comments that review hotels, which are categorized into comments with a positive and negative aspect. We would like to remark that almost all (more than 23 thousand) review comments that we have extracted contain both a positive and a negative part. Table 3 shows the total number of hotels and the average number of comments per hotel for the 4 different cities of Japan.

City	hotels	avg num. of comments per hotel
Kyoto	100	71
Osaka	100	65
Tokyo	100	71
Kobe	82	33
Total	382	61

Table 3: The Japan hotels dataset containing more than 23,000 comments

Efficiency. The time required to load the user reviews is 186,769 ms. For evaluating the execution time, we measured the required time for analysing and scoring (according to the q_1 and q_2) 2,000 randomly selected reviews, returning the 10 most highly ranked ones. The minimum, maximum and average times were 21 ms, 6,427 ms and 56 ms respectively (on average each review has 48 words), and the total time was 113,870 ms. It follows that the proposed method is acceptable in terms of efficiency. Specifically, if we assume that we have 3 hotels in the current user focus and the average number of reviews per hotel is 61 (as shown in Table 3), we can score all reviews in around 10 secs.

Quality. Since the reviews are not annotated with binary relevance scores for the two used queries, it is difficult to evaluate the quality of the scoring methods on this collection. Annotating the whole collection is a laborious and time consuming task. However we have started to manually annotate a part of the full reviews for the two queries that we have used in the FRUCE Collection. For the time being, we have marked 71 distinct comments, and identified 66 relevant and 76 irrelevant (c_i, q_i) pairs. The average top-2 precision of the IV method for the 2 queries by considering *only* the subcollection of 71 human judged comments is 0.5, while the average R -precision ($R = 33$) is again 0.5. We have noticed that

we would get higher results if the comments were clean, in the sense that the collection has several spam comments that affect negatively the results. Currently, we are in the process of cleaning the collection.

5 Conclusion

In the context of Faceted Search quite often the structured data are not enough for answering a users query. In such cases the system could resort to related textual comments (posed in natural language) for identifying those that could be exploited for helping the user. This requires finding the most relevant comments that (a) are associated with the most preferred objects, and (b) are related to a user’s question. Moreover, spoken dialogue interaction poses increased requirements on quality, in order to avoid wasting user’s time by reading irrelevant comments. To this end, we introduced a dictionary-based method that uses WordNet, a word embedding-based method, specifically Word2vec, and one that combines both. The analysis and the experimental results showed that the key result is that without dictionaries (either human-made or statistical ones), the effectiveness of retrieving the relevant comments is very low even in a small dataset. Specifically, the baseline method achieved $mean\ AveP = 0.05$ and $mean\ R - precision = 0.05$. However the method that uses both WordNet and Word2vec outperforms every other method with $mean\ AveP = 0.569$ and $mean\ R - precision = 0.649$, taking on average only 13 ms to score a review. We believe that the proposed method can be applied in several domains and for various tasks, from booking services to product selection. As part of our future work we plan to: (a) continue the quality evaluation over the real dataset, (b) extend the system described in [13] with this functionality, (c) investigate the applicability of comparative opinion mining and query-oriented sentiment analysis, and (d) investigate how we could exploit external sources in cases where even the user comments/reviews are not sufficient for answering a user’s question.

References

1. D. Chen, A. Fisch, J. Weston, and A. Bordes. Reading wikipedia to answer open-domain questions. *CoRR*, abs/1704.00051, 2017.
2. K. Cortis, A. Freitas, T. Daudert, M. Hürlimann, M. Zarrouk, S. Handschuh, and B. Davis. Semeval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news. In *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017*, pages 519–535, 2017.
3. M. Diao, S. Mukherjea, N. Rajput, and K. Srivastava. Faceted search and browsing of audio content on spoken web. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1029–1038. ACM, 2010.
4. S. Ferré. Sparklis: an expressive query builder for sparql endpoints with guidance in natural language. *Semantic Web*, 8(3):405–418, 2017.
5. M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966, 2015.

6. P. Lionakis and Y. Tzitzikas. Pfsgeo: Preference-enriched faceted search for geographical data. In *OTM Federated International Conferences "On the Move to Meaningful Internet Systems"*, pages 125–143. Springer, 2017.
7. B. L. López-Ochoa, J. L. Sánchez-Cervantes, G. Alor-Hernández, M. A. Abud-Figueroa, B. A. Olivares-Zepahua, and L. Rodríguez-Mazahua. An architecture based in voice command recognition for faceted search in linked open datasets. In *International Conference on Software Process Improvement*, pages 174–185. Springer, 2017.
8. C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.
9. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
10. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.
11. G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995.
12. P. Papadakos and Y. Tzitzikas. Comparing the effectiveness of intentional preferences versus preferences over specific choices: a user study. *International Journal of Information and Decision Sciences*, 8(4):378–403, 2016.
13. A. Papangelis, P. Papadakos, M. Kotti, Y. Stylianou, Y. Tzitzikas, and D. Plexousakis. Ld-sds: Towards an expressive spoken dialogue system based on linked-data. In *Search Oriented Conversational AI, SCAI 17 Workshop (co-located with ICTIR 17)*, 2017.
14. G. Petrucci and M. Dragoni. An information retrieval-based system for multi-domain sentiment analysis. In F. Gandon, E. Cabrio, M. Stankovic, and A. Zimmermann, editors, *Semantic Web Evaluation Challenges*, pages 234–243. Cham, 2015. Springer International Publishing.
15. G. M. Sacco and Y. Tzitzikas. *Dynamic taxonomies and faceted search: theory, practice, and experience*, volume 25. Springer Science & Business Media, 2009.
16. E. Sherkhonov, B. C. Grau, E. Kharlamov, and E. V. Kostylev. Semantic faceted search with aggregation and recursion. In *International Semantic Web Conference*, pages 594–610. Springer, 2017.
17. Y. Tzitzikas, N. Bailly, P. Papadakos, N. Minadakis, and G. Nikitakis. Using preference-enriched faceted search for species identification. *International Journal of Metadata, Semantics and Ontologies*, 11(3):165–179, 2016.
18. Y. Tzitzikas and E. Dimitrakis. Preference-enriched faceted search for voting aid applications. *IEEE Transactions on Emerging Topics in Computing*, PP(99):1–1, 2016.
19. Y. Tzitzikas, N. Manolis, and P. Papadakos. Faceted exploration of rdf/s datasets: a survey. *Journal of Intelligent Information Systems*, pages 1–36, 2016.
20. Y. Tzitzikas and P. Papadakos. Interactive exploration of multidimensional and hierarchical information spaces with real-time preference elicitation. *Fundamenta Informaticae*, 20:1–42, 2012.
21. L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman. Deep learning for answer sentence selection. *CoRR*, abs/1412.1632, 2014.