

DECiSION: Data-drivEn Customer Service InnovatiON

Dario Esposito[1(✉)], Marco Polignano[2], Pierpaolo Basile[2], Marco de Gemmis[2], Davide Primiceri[2], Stefano Lisi[1], Mauro Casaburi[3], Giorgio Basile[3], Matteo Mennitti[4], Valentina Carella[4], and Vito Manzari[4]

[1] Polytechnic University of Bari, Bari, Italy
dario.esposito@poliba.it
[2] University of Bari Aldo Moro, Bari, Italy
[3] Planetek Italia, Bari, Italy
[4] Sud Sistemi, Bari, Italy

Abstract. The paper presents DECiSION, an innovative framework in the field of Information Seeking Support Systems, able to retrieve all the data involved in a decision-making process, and to process, categorize and make them available in a useful form for the ultimate purpose of the user request. The platform is equipped with natural language understanding capabilities, allowing the interpretation of user requests and the identification of information sources from which to independently retrieve the information needed for the sensemaking task. The project foresees the implementation of a chatbot, which acts as a virtual assistant, and a conversational recommender system, able to dialogue with the user to discover their preferences and orient their answers in a personalized way. The goal is therefore to create an intelligent system to answer autonomously and comprehensively questions posed in natural language about a specific reference domain, to support the decision-making process. The paper describes the general architecture of the framework and then focuses on the key component that automatically translate the natural language user query into a machine-readable query for the service repository.

Keywords: Information Seeking Support Systems · Natural language processing · Information Filtering

1 Introduction

Decision-making involves identifying the various possibilities of alternative actions and choosing one or more of these through an evaluation process which should be sensible, rational and based on evidence. Thus, decision-making is a complex process that requires the integration of different information generated by different sources which must be interpreted to build useful knowledge. Moreover, in complex decision-making situations, problems are not well structured or clearly defined and these conditions can produce uncertainty. In particular, one key issue is to retrieve the information needed to solve the decision problem.

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12252, pp. 94–103, 2020.
https://doi.org/10.1007/978-3-030-58811-3_7

However, there are information seeking scenarios that cannot be solved merely through a straightforward matching of queries and documents When information is sought for learning, decision making and other complex mental activities, retrieval is necessary but not sufficient. According to Pirolli, many information search tasks are part of a broader class of tasks called sensemaking [1]. Such tasks involve finding information from large document collections, categorizing and understanding that information, and producing some product, such as an actionable decision. Examples of such tasks include understanding a health problem in order to make a medical decision or deciding which laptop to buy. Information seeking support systems (ISSSs) provide search solutions that empower users to go beyond single-session lookup tasks and aim at serving the more complex requirement: "Tell me what I don't know that I need to know". They provide services that aid people in managing and analyzing sets of retrieved information [2].

The DECiSION project therefore aims to create a framework to build Decision Support Systems that can be used through an artificial dialogue system able to query heterogeneous data sources and to provide relevant and complete answers to the user's needs. These responses are customized according to user interaction habits. The DECiSION project was proposed to overcome the lack of tools for accessing structured and unstructured data services and archives that can provide an effective support to decision-making processes.

The project developed knowledge, technologies, algorithms and interfaces. These were organized in a software technology to query, interact and consult heterogeneous data sources, obtaining information and suggestions compatible with the user-citizen-organization profile that required information. Starting from the natural language requests that users make, the process of interpretation of the expressed need and the process of generating multi-format answers are activated. For example, the system is able to cross several databases in order to provide technical users with an answer which combines natural language and graphic visualization tools, allowing them to find answers to requests such as: "Which pipelines are at risk according to the weather conditions and the stability of the territory? Which of these have a greater impact on users in the event of an accident?". A distinctive feature of DECiSION is the adoption of strategies that prioritize responses favoring sustainable choices in the user's decision-making process, taking into account the time and space constraints which the user faces.

The system is composed of various modules that allow the semantic analysis of the request. Then, through the analysis of heterogeneous data sources, the system synthesizes the answers in dynamic, personalized and contextualized formats. These answers contain information "enriched" by new knowledge, generated through intelligent automatic reasoning processes. In Sect. 2, we review some related work in the area of Information Filtering, Sect. 3 describes the framework, while in Sect. 4 we focus on the key feature of DECiSION, the capability of understanding the user request and turning it into queries for system knowledge base. Section 5 describe some preliminary experiments with the natural language component of the framework, before concluding the paper with some final comments.

2 Motivations and Related Research

When a user faces a decision, he has to evaluate many options, for example a set of items from an Internet store. Often he hasn't all the necessary knowledge to choose the best option. A support system is necessary for evaluating all the options available and for helping the user to choose the best option, by taking into account preferences and behaviors. Information Filtering are appropriate for this task, because they are able to select a set of items considering different relevant personalized aspects for every single user, using the information stored usually in a user profile. Modern filtering systems, called recommender systems, can adopt different strategies for selecting relevant items for the user such as collaborative, content-based, knowledge based, context-aware techniques [3]. In particular, content-based approached exploits item descriptions in order to infer general properties of preferred that can be used to support the user in future choosing tasks. For instance, a movie recommender could learn user preferences from the content of items the user has liked in the past in order to provide suggestions based on preferred movie genres, actors or directors. Research in this area has many practical applications. Here we discuss emerging applications or new trends in traditional domains, usually aiming at improving commercial systems.

IP television (IPTV) services provide users with multimedia content (movies, news programs, documentaries, TV series) via broadband Internet networks. Bambini in a remarkable paper describes the adoption of both content and collaborative recommendation methods in the production environment of the one of the largest European IPTV providers [4]. News recommendation is a domain where content-based methods find their natural application. MESH (Multimedia sEmantic Syndication for enHanced news Services) is a research project that designed a framework for intelligent creation and delivery of semantically-enhanced multimedia news information [5]. Another interesting domain of application is recommendation of financial investment strategies, a complex and knowledge-intensive task where filtering techniques are recently used [6]. In the DECiSION project, we try to design a general framework that can be used in different domains. The proposed framework integrated several features:

1. Interpretation of the needs expressed by the user request;
2. Searching into heterogeneous sources of both structured and unstructured data;
3. Providing personalized answers based on the user profile and contextual information.

In this paper we focus on item 1. and 2., by describing the solutions adopted in the framework for the comprehension of the user need and automated query of the service catalog of the system.

3 The DECiSION Framework

The architecture of the framework is described in Fig. 1, which shows the logical-functional model, further developed and detailed through the different project research activities. The process which accounts for the system's ability to interpret data and requests is managed in block "a". It involves the processing of heterogeneous data

sources in order to extract the content necessary for the semantic interpretation of the data. The key idea here is to include in a knowledge base all the services which could provide data useful for the decision-making task. All these services are enriched by the indexing component (block "b") which is a layer that operates "normalization" and "enrichment" of annotations built to provide the semantic interpretation.

Actually blocks "a" and "b" are builds the semantically-annotated catalog service which will be the target of the query component (block "c").

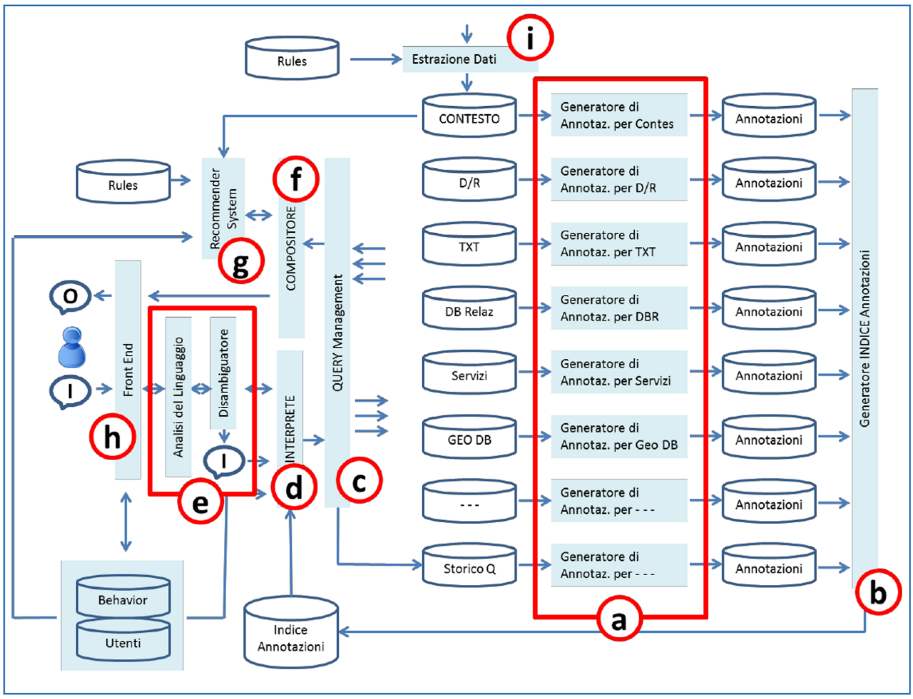


Fig. 1. DECiSION logical-functional architecture

For the implementation of these blocks, a specific module has been developed, which allows to store the OWL-S annotations in a graph database Neo4j (www.neo4j.com).

For instance, if a service that provides demographic data about customers must be included in DECiSION, it is described by OWL-S annotations, then indexed and included in Neo4j. In the same way, if traffic statistics or any statistics about other phenomena that fall within the scope of the decision must be included in the system, some services that provide this kind of data must be added to the knowledge base. In other works, blocks "a" and "b" are wrappers that connect the framework with data sources. The problem is how to understand, from the user query, which data sources contain the data needed to solve the decision task.

Components “c”, “d” and “e” manage this processes, which is devoted to the interpretation of the user request (i.e. understanding the intent of the user query) and its automated translation into queries on the available data sources. In this paper, we refer to this process as “service discovery”, which will be described in more detail in the next section.

The aim of the language analysis, disambiguation and interpretation processes is to extract the information required. Once this has been done, the query management module automatically builds queries to obtain useful information for the composition of the answer. Block “f” collects the results of queries to provide the response. The process is integrated with a recommendation. Indeed, the answer is personalized in terms of content and layout through block “g”, the recommendation component [7]. It aims to support the user decision-making process with additional suggestions which are relevant to the identified request purpose, e.g. more services available. This component is based on algorithms which favor socially responsible recommendations. Finally, the front-end component, block “h”, gives the answer in a simple and effective way through mobile applications, chatbots or user-friendly device interfaces.

Understanding natural language and using natural language to simplify dialogue with information access systems is one of the main challenges of research in the field of human-machine interaction. In particular, one of the most relevant problems addressed in the DECiSION project is that of the association of themes and meanings with the keywords /expressions used to formulate an information need. The paper focuses on this aspect of the framework.

4 Question Interpretation for Service Discovery: The Translator Component

The process that translates the user information need (natural language query) exploits a pipeline to analyze the text in order to extract its lexical, syntactical and semantic characteristics. The component, called Translator, carries out:

- A syntactical text analysis, where each node is a word and the edges show the dependency between them, e.g. subject, predicate, object;
- A disambiguation of the text, assigning correct meaning to each word;
- The recognition within the text of entities with a proper noun (people, places, organizations and so on);
- An entity annotation, exploiting of their relations and linking it to a concept contained in the database.

First, the user query (e.g. “find customers in London that might leave our company next year”) is processed by basic NLP operations (tokenization, stopword elimination, named entity recognition). Semantic interpretation of word is performed by word embeddings, a set of modeling techniques in which words of a vocabulary are mapped into vectors of real numbers. This makes it possible to store both semantic and syntactical information of words [8] starting from a non-annotated corpus and building a vector space in which word vectors are closer if words occur in the same linguistic contexts, that is, if they are recognized as semantically more similar.

To learn the embedding the most used techniques are Word2Vec [9] and GloVe [10]. We don't need to train our model, as there are many models already trained on different corpus. We used Italian word embeddings trained on Wikipedia (http://hlt.isti.cnr.it/wordembeddings), and for English the embeddings are obtained with GloVe, trained on Wikipedia as well (nlp.stanford.edu/projects/glove). Then, we need to combine word embeddings in order to obtain a vector that identifies an entire sentence. This task is known as sentence embedding. Several methods are available for obtaining the vector of a sentence, ranging from simple additional composition of the word vectors to sophisticated architectures such as convolutional neural networks and recurrent neural networks.

Here the one proposed by Arora et al. is used, as it provides better performance than baselines on various textual similarity tasks and can even overcome other sophisticated methods [11]. This method represents the sentence by a weighted average of the word vectors. Once all the NLP steps are applied, we can move on the definition of the service discovery process.

All descriptions of services and output parameters are processed in the same way as the user request. From the descriptions of the output parameters it is possible understand what a service precisely provides: these textual descriptions are useful because they can be matched against what the user expects to get. The identification of a service takes place through three phases:

1. Computation of semantic similarity between request and service description;
2. Computation of semantic similarity between request and descriptions of the output parameters of a service;
3. Increase of the similarity score based on the number of recognized input parameters.

Let's consider the following query: "find places to enjoy in London"; suppose we have two services in the target repository (Table 1).

Table 1. Service descriptions

Service	Description
Service (s1)	searchTourism
Description	Search attractions to have fun near a specified place
Output description	Name and address of the attraction
Input	City (LOCATION), country (COUNTRY_CODE)
Service (s2)	searchBank
Description	Search a bank near a specified location
Output description	Name and address of the bank
Input	City (LOCATION), country (COUNTRY_CODE)

The first step applied is the computation of cosine similarity between sentence embeddings of service description and query.

After the first calculation we move on the descriptions of the output parameters of each service. We may have a service that meets the request, but that returns many

different outputs. In the above example, both services return name and address of a place, but the term “attraction” is more similar to “enjoy” than the term “bank”. As for the input parameters, we simply match the number and type of parameters are taken into account.

In the above example, both services have same similarity with the user request. In fact, only one parameter (type: location) is found in the user request and matched against parameter descriptions of the services. From these services, we expect the module to return search Tourism as a suitable service to satisfy the request.

5 Preliminary Experiments

In this section, we evaluate the effectiveness of the service discovery component compared with a simple keyword-based approach. An extended experiment will be performed that will involve all the DECiSION components.

We build a dataset of 64 user queries, 32 in English, 32 in Italian, in order to assess the effectiveness of the component in different languages. For each user query, a set of relevant services that we expect to be identified was defined. Table 2 shows some examples of entries in the dataset.

Table 2. User request and relevant services

Request	Relevant services
Tell me current weather in Florence	GetWeather, GetWeatherByZip, GetWeatherLatLng
Find a station in Milan	SearchStation
What should I visit in London	SearchTourism

As a baseline, we consider the simple keyword search approach implemented by Lucene (lucene.apache.org). We used Lucene to index service descriptions in the dataset. The result of a search on a Lucene index is a list of documents, previously indexed, that contain one or more words constituting the search query.

As for the metrics, we used two common metrics in the area of Information Retrieval: Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR). To understand how to calculate MAP we first define Precision, P@k and Average Precision. Precision is the fraction of the documents retrieved that are relevant to the user’s information need:

$$Precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|}$$

P@k is Precision that takes into account only the top-k results returned by the system.

Average Precision is the average of P@k, defined as follows:

$$AveP = \frac{\sum_{k=1}^{n} P@k \cdot rel(k)}{|num\ relevant\ documents|}$$

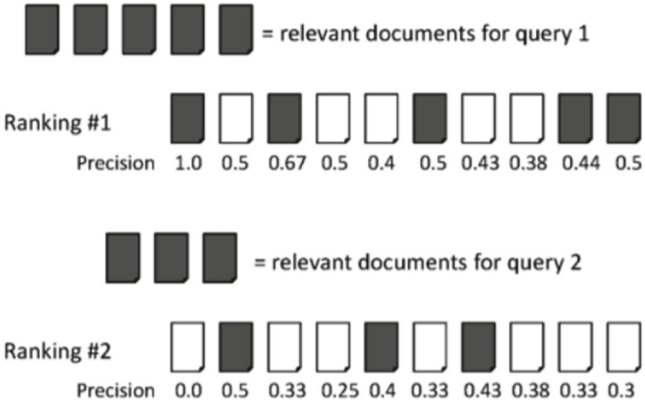
where rel(k) = 1 if the document k is relevant, 0 otherwise.

MAP is the Average Precision across multiple queries:

$$MAP = \frac{\sum_{k=1}^{|Q|} AveP_{Qk}}{|Q|}$$

where Ave P_{Qk} is the average precision of query Q_k.

An example of MAP computation on two queries follows.



$$AveP_{Q1} = \frac{1 + 0.67 + 0.5 + 0.44 + 0.5}{5} = 0.62$$

$$AveP_{Q2} = \frac{0.5 + 0.4 + 0.43}{3} = 0.44$$

$$MAP = \frac{0.62 + 0.44}{2} = 0.53$$

MRR metric is based on the idea that the system gets a score for each query, which is the reciprocal of the position (in the ranked list of results) of the service correctly returned:

$$MRR = \frac{\sum_{i=1}^{N} \frac{1}{pi}}{N}$$

For instance, suppose we have 3 queries in the evaluation set; suppose that the correct answer s returned in position #3 for the first query, position #2 for the second query and position #1 for the third query. Then

query 1:  $pi = 3$

query 2:  $pi = 2$

query 3:  $pi = 1$

$$MRR = \frac{\frac{1}{3} + \frac{1}{2} + 1}{3} = 0.61$$

We compare the results of the Translator with those of Lucene, as well as with a those obtained by a different version of the Translator which does not consider input / output parameters when computing query-service similarity. Results are reported in the following Table 3:

Table 3. Comparison of results.

	English		Italian		Overall	
	MAP	MRR	MAP	MRR	MAP	MRR
Lucene	0.61	0.64	0.54	0.57	0.58	0.61
Translator (no input)	0.79	0.81	0.85	0.86	0.82	0.84
Translator	0.89	0.91	0.86	0.88	0.88	0.90

From the results, we can see that the reduced version of the Translator, although working on the same data as Lucene (only textual description of the services), achieves much better performances. The reason for the improved performance is the adoption of a semantic-based approach that allows us to compute a more precise similarity between sentences than simple keyword-based search. We can see also that better results are obtained on English. This is probably due to the better accuracy of the English NLP pipeline.

The main outcome of the evaluation is that the full version of the Translator achieves a MAP value of 0.88, which shows that it is able to identify in a very good way most relevant services. Also, MRR value is high, which shows that the system is able to identify in most cases in the first position of the ranking a service able to satisfy the request. This aspect is very interesting for the overall performance of the framework, because it is more likely that the services retrieved first will be invoked.

6 Conclusions and Outlook

This paper describes the DECiSION framework, an innovative information-seeking system capable of supporting users in a decision-making process, integrating different heterogeneous data sources. The framework is able to interact with the user by using natural language.

The paper focused on the Natural Language component of the platform, the Translator, which is the key to understand the user information need and to invoke the services, available in the catalog of platform, which can support the decision task. Although the project is still an ongoing work, the preliminary evaluation shows promising results. As a future work, we will integrate in the framework the component devoted to combine the results retrieved by the translator and to answer to the user request. Furthermore, we will design an extended in-vivo evaluation with specific decision-making task that will be performed by real users by exploiting the DECiSION system in order to assess if the system actually support the decision-making process.

Acknowledgments. The research described in this paper has been funded by Regione Puglia under the programme INNONETWORK - POR Puglia FESR-FSE 2014–2020, Project "DECiSION" (code: BQS5153).

References

1. Pirolli, P.: Powers of 10: modeling complex information-seeking systems at multiple scales. Computer (Long Beach, California) **42**: 33–40 (2009). https://doi.org/10.1109/MC.2009.94
2. Marchionini, G., White, R.W.: Information-seeking support systems. Computer (Long Beach, California) **42**, 30–32 (2009). https://doi.org/10.1109/MC.2009.88
3. Ricci, F., Rokach, L., Shapira, B.: Recommender systems: introduction and challenges. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 1–34. Springer, Boston, MA (2015). https://doi.org/10.1007/978-1-4899-7637-6_1
4. Bambini, R., Cremonesi, P., Turrin, R.: A recommender system for an IPTV service provider: a real large-scale production environment. In: Ricci, F., Rokach, L., Shapira, B., Kantor, Paul B. (eds.) Recommender Systems Handbook, pp. 299–331. Springer, Boston, MA (2011). https://doi.org/10.1007/978-0-387-85820-3_9
5. Picault, J., Ribière, M., Bonnefoy, D., Mercer, K.: How to get the recommender out of the lab? In: Ricci, F., Rokach, L., Shapira, B., Kantor, Paul B. (eds.) Recommender Systems Handbook, pp. 333–365. Springer, Boston, MA (2011). https://doi.org/10.1007/978-0-387-85820-3_10
6. Musto, C., Semeraro, G., Lops, P., De Gemmis, M., Lekkas, G.: Personalized finance advisory through case-based recommender systems and diversification strategies. Decis. Support Syst. **77**, 100–111 (2015). https://doi.org/10.1016/j.dss.2015.06.001
7. de Gemmis, M., Lops, P., Musto, C., Narducci, F., Semeraro, G.: Semantics-aware content-based recommender systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 119–159. Springer, Boston, MA (2015). https://doi.org/10.1007/978-1-4899-7637-6_4
8. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. **3**, 1137–1155 (2003)
9. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality (2013)
10. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
11. Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings (2016)