Standardisation of Hotel Descriptions

Nuno Miranda[1] Ricardo Raminhos[1] Pedro Seabra[1]
José Saias[2] Teresa Gonçalves[2] Paulo Quaresma[2]

[1] Viatecla SA, Almada, Portugal
{nmiranda,rraminhos,pseabra}@viatecla.com
[2] Dep. Informática – Universidade de Évora, Évora, Portugal
{jsaias,tcg,pq}@di.uevora.pt

Abstract. The description of tourism products (hotel, aviation, rent-a-car and holiday packages) is strongly supported by natural language expressions. Due to tourism dynamics and the extent of its offers, manual data management is not a reliable nor scalable solution: descriptions are structured in different ways, possibly comprising different languages, complementing and/or overlapping one another. This paper presents a prototype that automatically classifies and extracts relevant knowledge from real operational hotel descriptions retrieved from the KEYforTravel tourism application framework. Captured knowledge is represented in a normalised format enabling the development of new business services.

1 Introduction

The Laboratory of Excellence .NET in Évora is the result of a protocol between ViaTecla, University of Évora and Microsoft, to introduce the experience and academic knowledge in a business context. In this sense, the Laboratory is a multi-disciplinary space with the presence of teachers, researchers and students (bachelor, master and doctorate). *Standards for Tourism Product Descriptions* is its first project, aiming for the automatic extraction of relevant features from the tourism products currently residing on KEYforTravel [7] (K4T) application.

The K4T is a tourism application framework developed by ViaTecla that provides a rapid and effective response, through the interconnection of the various participants of the tourism industry, assuring not only the view and exchange of information between them, but also the various areas of the product selling process. Since K4T gathers tourism information and products from heterogeneous sources, it is crucial to offer a standardised way of presenting its offers. The *Standards for Tourism Product Descriptions* project addresses this challenge.

One specific problem deals with hotel descriptions. Tourism operators have the hard and time-consuming task of manually carrying out the survey of hotel characteristics: they read the description provided by the hotel and insert its characteristics (equipment, services and location) into a database or a Web template. With the increasing globalisation of the tourism market, this situation becomes impractical since a single tour operator may offer thousands of hotels. The project's first practical application aims at processing that information by automatically extracting useful information and standardising the hotel description.

This paper is organised as follows: Section 2 describes system's architecture, Section 3 evaluates it and Section 4 points out conclusions and future work.

2 System's Architecture

The system aims to receive an hotel description and produce a standardised version of it. It was designed using a divide and conquer strategy where several small tools that focus on specific and simpler problems were interconnected. There are four main tools: a *Sentence Classifier*, an *Entity Extractor*, an *Ontology Instantiator* and an *Ontological Translator*, that where packed into a Web Service for the system to be available online. This architecture is depicted on Figure 1.

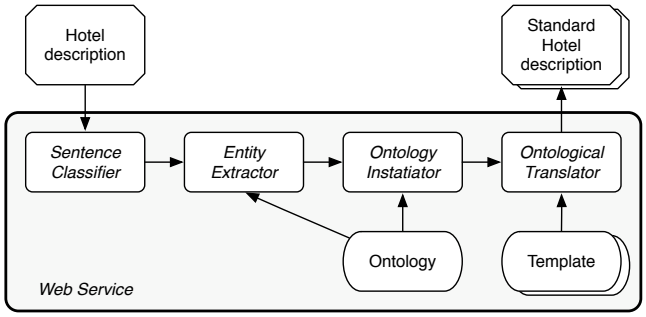

Fig. 1. System architecture diagram.

Sentence Classifier. This prototype's module is responsible for examining and classifying chunks of natural language text. It receives the crude description of the hotel and divides it into sentences. The sentences are then individually examined and automatically classified into a set of predefined classes such as "Equipment", "Service" and "Location". Each sentences can belong to more than one class, such as "Equipment" and "Service", provided it contains elements of both classes, or none of them, if it doesn't present any evidence. This classification aims at filtering the sentences by type to ensure a specific treatment to each one by the *Entity Extractor* module.

The *Sentence Classifier* was built using a Machine Learning approach. Since sentences can belong to more than one class, we have a multi-label classification problem. This kind of classification problem is typically solved by dividing it into a set of binary classification problems, where each concept is considered independently. Using a bag-of-words [6] representation for sentences, three different classification algorithms were applied and their decisions combined. The committee was comprised by a decision tree (C4.5 [5] algorithm), a Naïve Bayes classifier [3] and a support vector machine (SMO [4], the sequential minimal optimisation algorithm). This module's prototype was built using WEKA [9].

Entity Extractor. Having the sentences classified and grouped by type, the system tries to extract useful entities. This is the goal of this module that comprises two steps: finding useful entities and dealing with misspelled words.

Due to the fact that hotel descriptions are given in natural language without any pattern or consistency, the same entity can be described not only by a single term but by a set of synonyms, or even by an abbreviation. It can also be the case that the entity reference is misspelled or have failures in diacritics. This last hypothesis is very common since raw descriptions are frequently translated to different idioms and loose the regional diacritics.

To find useful entities on the description of each type of sentence a pattern matching approach was used. This is accomplished defining a set of more or less complex regular expressions able to identify synonyms, abbreviations and "almost" well-written words (e.g. `television`, `T.V.` and `TV` or `mini-bar` and `mini bar`) for common terms used for describing that kind of information.

To cope with misspelled words, the Levenshtein distance [2] is then used as a function to measure the similarity between the words not yet extracted and the ones that are considered relevant to the sentence's type.

Ontology Instantiator. To maintain the entities extracted from the system, and aiming to provide the basic structure and organisation of the involved concepts, an ontology for hotel domain was developed. We used Web Ontology Language [8] (OWL) that besides defining the structure also considers possible semantic relationships between objects and attributes. Each ontology object contains the set of regular expressions and Levenshtein functions used by the *Entity Extractor* module. In this way, the *Entity Extractor* becomes independent of specific problem at hands.

Using the developed ontology, this module generates an OWL instance that contains the entities with their attributes and their semantic relationships. This instance is then accessed using the Jena Semantic Web Framework [1].

Ontology Translator. This module is responsible for turning the extracted Entities attractive and easy to read by humans. Using a XML template that gives the skeleton for the final information representation, the *Ontological translator* fills it with the extracted instances. This template can later be replaced by another according to the preference of the tour operator or the target audience (e.g. corporate versus leisure clients).

Although in the present architecture ontology instances are the input for the *Ontology Translator*, this normalized knowledge (easily computable) can be applied to substantially expand and improve search capabilities in tourism offers since each Service, Equipment or Location item can be used in the query itself, or as a parameter in the search results refinement process. Further, since knowledge is formalized using an hierarchical structure, it may be applied to graphically map related items as well as structure navigation.

3 Evaluation

During the development of the project, and taking into account its future use, several tests were carried out using hotel descriptions residing on the Keyfor-Travel application. Figure 2 shows an example of running the system with an hotel description (in Portuguese) and three standard descriptions: a leisure template for English and a leisure and corporate templates for Portuguese.

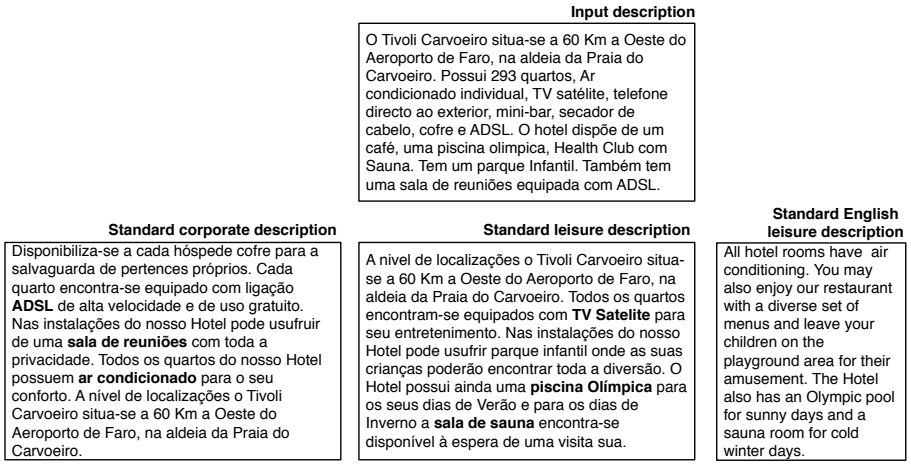

Fig. 2. Standard descriptions generated by the system.

4 Conclusions and Future Work

One can say that the project's main objectives were reached since it was possible to extract useful information, standardise it and create computable objects from plain text descriptions. Also, the used Machine Learning and string processing techniques revealed to be fully applicable to this domain. There is work to be done for normalising other tourism products and since there is room for improvements on its various modules, we hope to increase its overall performance.

In another context, one can also say that is possible to carry out joint projects between the University and business worlds and this is a good example of it.

References

1. HP Development. Jena – A Semantic Web Framework. http://jena.sourceforge.net.
2. V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710, 1966. Originally publish in Russian.
3. A. McCallum and K. Nigam. A Comparison of Event Models for Naive Bayes Text Classification. In *AAAI-98 Workshop on Learning for Text Categorization*, 1998.
4. J. Platt. Fast Training of Support Vector Machines using Sequential Minimal Optimization. In B. Schölkopf, C. Burges, , and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 185–208. MIT Press, 1999.
5. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.
6. G. Salton, A. Wang, and C. Yang. A vector space model for information retrieval. *Journal of the American Society for Information Retrieval*, 18:613–620, 1975.
7. ViaTecla. KEYforTravel platform. http://www.keyfortravel.com.
8. W3C. OWL Web Ontology Language Guide. http://www.w3.org/TR/owl-guide.
9. I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, US, 2nd edition, 2005.