Information Extraction for Standardization of Tourism Products

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

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

Abstract. Tourism product descriptions are strongly supported on natural language expressions. Appropriate offer selection, according to tourist needs, depends highly on how these are communicated. Since no human interaction is available while presenting tourism products online, the way these are presented, even when using only textual information, is a key success factor for tourism web sites to achieve a purchase. Due to the large amount of tourism offers and the high dynamics in this sector, manual data management is not a reliable or a scalable solution. This paper presents a prototype developed for automatic extraction of relevant knowledge from tourism-related natural language texts. Captured knowledge is represented in a normalized format and new textual descriptions are produced according to available marketing channels. At this phase, the prototype is focused on hotel descriptions and is already using real operational data retrieved from the KEYforTravel tourism platform.

1 Introduction

Online presence of tourism related web sites has accompanied the exponential growth of the Internet. As an example, US online tourism revenues between 2001–2004 have increased on average 29% per year [5]. An important subset of these revenues, correspond to commercial web sites (e.g. *Expedia*) where users can browse and search tourism offers, inspect details, perform and fulfill reservations by themselves, following a self-booking tool approach.

Even today, when multimedia is increasing its presence on the web, natural language textual descriptions still remain by far the mostly used format for e-marketing and promotion applied to tourism products (i.e. hotel, aviation, rent-a-car, holiday packages). Descriptions presented to the user are provided by external service connectors (e.g. *GTA* or *HotelBeds* for hotel products) or updated manually by the tourism online operator. Offers are structured differently (complementing or overlapping each other) and mostly consists on simple enumerations of available services and equipments. Usually these textual descriptions are presented to the end user as obtained from the service provider without prior preparation.

This work presents a prototype developed for the automatic extraction of relevant knowledge from tourism-related textual expressions that enables the creation of appropriate descriptions according to user profile in order to provide a suitable segmented e-marketing and promotion process. For now, the prototype is focused on the hotel descriptions subset using real operational data retrieved from the KEYforTravel [22] tourism platform.

The paper is organized as follows: Section 2 presents the application domain and Section 3 introduces the Information Extraction and Text Classification tasks. Section 4 describes the System Architecture and Experiments and Evaluation are carried out on Section 5. Finally, Section 6 presents some conclusions and points out possible future work.

2 The application domain

Hotel descriptions are commonly available in natural language text. It usually contains information about:

- hotel services commonly shared by all tourists,
- equipment made available on each room and
- location information, normally relative to some well-known point of interest (e.g. street, monument or metro station).

Each description tries to summarize (in the minimum amount of text) the multiple features and benefits made available by the hotel. Together with price and availability factors, they are responsible for attracting customers.

Although all relevant information is comprised within the description, it is mostly used for presentation purposes. In this way, all individual knowledge isn't potentiated for searching and offer refinement purposes (e.g. search for all hotels with jacuzzi and swimming pool located nearby a metro station in London).

Hotels requiring strong market projection are usually present in one or more hotel service aggregators. Commercial tourism web sites can use multiple of these services to present hotel offers world-wide, resulting in possible multiple descriptions for the same hotel. Depending on each connector business focus, some hotel features may be found more or less relevant, resulting in disjoint descriptions. When detected, one of these descriptions is usually elected as primary and becomes the only one to be considered for hotel presentation (all others descriptions are discarded, as well as their complementary information, even if not present in the primary description). Further, for commercial tourism sites that provide world-wide hotel offer (in the order of several thousands) it is not possible to individually manage each hotel description. Thus, many commercial sites choose to present directly to the user the description made available by the hotel connector. This results in three main problems:

- there is no differentiation between tourism online operators sharing the same connectors
- descriptions are not targeted to the user/market segmentation

– descriptions are not controlled nor normalized. This results on different, heterogeneous descriptions presented altogether.

Building a system that extracts all relevant hotel features and normalizes them it is possible to address the previous posed problems.

Hotel descriptions are stored in the KEYforTravel platform, that gathers them from several external service connectors (like *GTA* or *HotelBeds*). Thus, the system goal is to normalize information from different sources and aggregate it in KEYforTravel clients in a standardized way (both in a structured way like tables or a natural language one).

3 Information Extraction and Text Classification

This section introduces the information extraction task and the text classification problem along with the all-purpose classification algorithms used in this work.

3.1 Information Extraction

Information Extraction is a type of information retrieval whose goal is to automatically extract structured information (categorized, contextually and semantically well-defined data) from a certain domain, from unstructured machine-readable documents. This has been an active area of research, exhibited in a series of Message Understanding Conferences (MUCs) [6, 23] and more recently in ACE evaluations [12].

Detecting entities in natural language text typically involves disambiguating phrases based on the words in the phrase, and the text surrounding the candidate entity. Explored approaches include hand-crafted rules [7], rule learners [1]and other machine learning approaches (e.g. [2]). Another line of research generates probabilistic models. Hidden Markov Models (HMMs) are popular sequential models that have been used in the context of IE [4],as well as other frameworks like Conditional Markov Models [9] and Conditional Random Fields [10].

3.2 Text classification

Text Classification is a well studied task with many effective techniques. Nowadays, the most popular and successful algorithms for text classification are based on machine learning techniques. Several algorithms have been applied, such as decision trees [20], linear discriminant analysis and logistic regression [18], Naïve Bayes classifier [14] and Support Vector Machines (SVM)[8].

Learning systems have the advantage of flexibility since the only required human effort is to provide a consistent set of labeled examples. Originally, research in text classification addressed the binary problem, where a document is either relevant or not w.r.t. a given category. In real-world situation, however, the great variety of different sources and hence categories usually poses multi-class classification problem, where a document belongs to exactly one category

selected from a predefined set. Even more general is the case of multi-label problem, where a document can be classified into more than one category. This kind of classification problem is typically solved by dividing it into a set of binary classification problems, where each concept is considered independently.

Documents must by pre-processed to obtain a structured representation to be fed to the learning algorithm. The most common approach is to use the bag-of-words representation [17], where each document is represented by the words it contains (their order and punctuation are ignored). Normally, words are weighted by some measure of word's frequency in the document and, possibly, the corpus. In most cases, a subset of words (stop-words) is not considered, because they do not have discriminating power over different classes; some works reduce semantically related terms to the same root applying a lemmatizer.

Decision Tree. A text classifier represented by a decision tree is a tree in which internal nodes represent words, branches represent values which the words may have and the leaves represent classes. It classifies a document verifying recursively (from the root), the node's word and traversing the branch with the document word's value until reaching a leaf; the document belongs to the class labeled by the leaf. The induction of a decision tree is achieved applying a divide and conquer strategy: it verifies if all examples have the same label and, if not, selects a word w and divides the document set on subsets with the same value for w, putting each subset into distinct subtrees. This process is repeated recursively for each of the subtrees until all leaves contain only instances of the same class, which is then chosen as the document label. The key step of the algorithm is the choice of the word w on which the partition is made. This choice is usually made according to a criterion of mutual information or entropy.

Naïve Bayes Classifier. Naïve Bayes classifier is a probabilistic classifier that sets the class of a given document by choosing the class that maximizes the probability of the document, given its attributes. This probability is calculated applying the Bayes theorem and assuming that each attribute is independent from the others. Even naively assuming attribute independence, this algorithm has shown good results on text classification.

Support Vector Machines. Support Vector Machines was motivated by theoretical results from the statistical learning theory: it joins a kernel technique with the structural risk minimization framework. *Kernel techniques* comprise two parts: a module that performs a mapping from the original data space into a suitable feature space and a learning algorithm designed to discover linear patterns in the (new) feature space. The *kernel function*, that implicitly performs the mapping, depends on the specific data type and domain knowledge of the particular data source. The *learning algorithm* is general purpose and robust. It's also efficient, since the amount of computational resources required is polynomial with the size and number of data items, even when the dimension of

the embedding space (the feature space) grows exponentially [19]. A mapping example is illustrated in Fig. 1a).

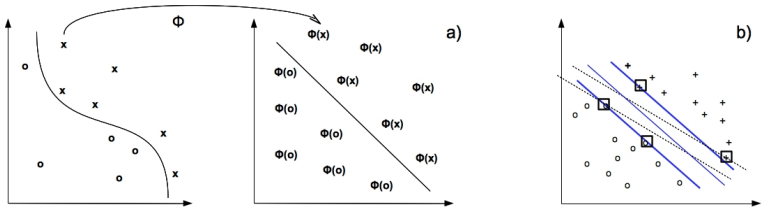

Fig. 1. The SVM approach: kernel transformation and search for maximum margin.

The *structural risk minimization* (SRM) framework creates a model with a minimized VC (Vapnik-Chervonenkis) dimension [21]. This theory shows that when the VC dimension is low, the expected probability of error is also low, which means good performance on unseen data (good generalization). In geometric terms, it can be seen as a search to find, between all decision surfaces that separate positive from negative examples, the one with maximum margin (the one having a separating property that is invariant to the most wide translation of the surface). This property is enlighten in Fig. 1b) for a 2-dimensional problem.

4 System architecture

The system aims to receive an hotel description and produce a standardized 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 Fig. 2, with the information flow between tools also represented.

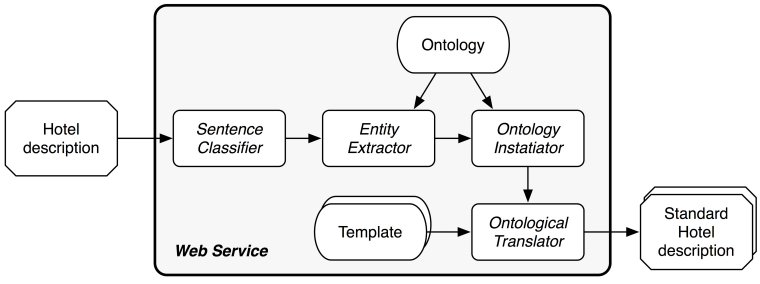

Fig. 2. System architecture diagram.

4.1 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 sentence can belong to more than one class, provided it contains elements of those 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 machine learning approach and because sentences can belong to more than one class, we are in presence of a multi-label text classification problem. After yielding the set of experiments (section 5.1) the classifiers were built using a Naïve Bayes classifier with the bag-of-words representation of the sentences with binary word weights.

The classifiers were built with WEKA [25], a software package developed by New Zealand Waikato University that implements a collection of ML algorithms.

4.2 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 abbreviations. It can also be the case where 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 was accomplished by defining a set of some 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 similarity between words not yet extracted and the ones considered relevant to the sentence's type is measured using the Levenshtein distance [11].

4.3 Ontology Instantiator

To retain the entities extracted from the texts, and aiming to provide the basic structure and organization of the involved concepts, an ontology for hotel domain was developed. This ontology comprises 89 classes e 88 relations.

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.

The hotel ontology was conceived using the Web Ontology Language [24] (OWL), a language designed to be used by applications that need to process Web information content. It facilitates machine interpretability by providing additional vocabulary along with a formal semantics, that besides defining structure, 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 the specific problem at hands.

Using the ontology, this module generates an OWL instance populated with the extracted entities jointly with their attributes and semantic relationships. This instance is then accessed using the Jena Semantic Web Framework [3].

4.4 Ontology Translator

This module is responsible for turning the extracted information attractive and easy to read by humans giving it different flavors according to the preference of the tour operator or the target audience (e.g. corporate versus leisure clients).

This module uses a XML template giving the skeleton for the final information representation and filling it with the extracted information.

5 Experiments and Evaluation

This section presents the experiments done to build the *Sentence Classifier* and the outputs generated by the system when presented with an hotel description.

5.1 Experimental setup

To build the *Sentence Classifier* we made several experiments with different bag-of-words term weighting representations and different classification algorithms:

- binary, word count and tfidf [17] normalized to unit length term weighting;
- decision tree [16], naïve Bayes [13] and support vector machine (SMO, sequential minimal optimization [15]) algorithms;

The algorithms were run with their default parameters and the model was evaluated using a 10-fold stratified cross-validationprocedure with 95% confidence level significance tests.

5.2 Results

We are interested achieving maximum recall, at the cost of lower precision, because false positives are treated later by the *Entity Extractor* of that class. Nevertheless, Table 1 shows recall, precision and F_1 measures for each class, term weight and algorithm.

From the results we can see that the highest recall is achieved by the naïve Bayes classifier with the binary term weight. Bold-face values are significantly better than that setting while italic-face are significantly worse. Also, F_1 values for that setting are only significantly worse than SVM algorithm for the `Equipment` class.

		Services			Equipment			Location		
term weight	classifier	rec	prec	F_1	rec	prec	F_1	rec	prec	F_1
binary	nBayes	.987	.795	.876	.969	.657	.776	.919	.703	.792
	SVM	*.816*	**.912**	.854	*.843*	**.951**	**.883**	*.724*	**.893**	.789
	dTree	*.601*	.863	*.696*	*.724*	**.961**	.802	*.589*	**.955**	.713
word	nBayes	*.844*	**.952**	.890	*.693*	**.990**	.800	*.510*	**.988**	*.658*
	SVM	*.830*	**.905**	.858	*.858*	**.951**	**.893**	*.711*	**.892**	.779
	dTree	*.608*	.858	*.699*	*.689*	**.954**	.775	*.623*	**.956**	.736
tfidf	nBayes	*.868*	**.969**	.911	*.749*	**.974**	.835	*.451*	**.960**	*.592*
	SVM	*.846*	**.904**	.866	*.865*	**.942**	**.893**	*.700*	**.891**	.771
	dTree	*.742*	.799	*.760*	*.716*	**.932**	.789	*.691*	**.919**	.773

Table 1. Precision, recall and F_1 values for each category.

5.3 System Evaluation

With the *Sentence Classifier* defined, and taking into account its future use, several tests were carried out using hotel descriptions residing on the KeyforTravel application. Fig. 3 shows an example of hotel description (in Portuguese) and the result of running the system with three different flavors of the *Ontology Translator*: a corporate and leisure templates for Portuguese and a leisure one for English.

For this example, the system was able to identify the hotel location and all the available equipment (8 ontology instances) and services (6 ontology instances) and generate descriptions focusing on different sets of attributes according to the corporate or leisure flavor. For instance, while corporate description centers attention on internet access, meeting room and air conditioning, the leisure description spots the satellite TV, olympic pool and sauna.

6 Conclusions and Future work

This paper presents a methodology for extracting useful information from textual descriptions of tourism products and standardizing it. This method was applied to hotel descriptions domain.

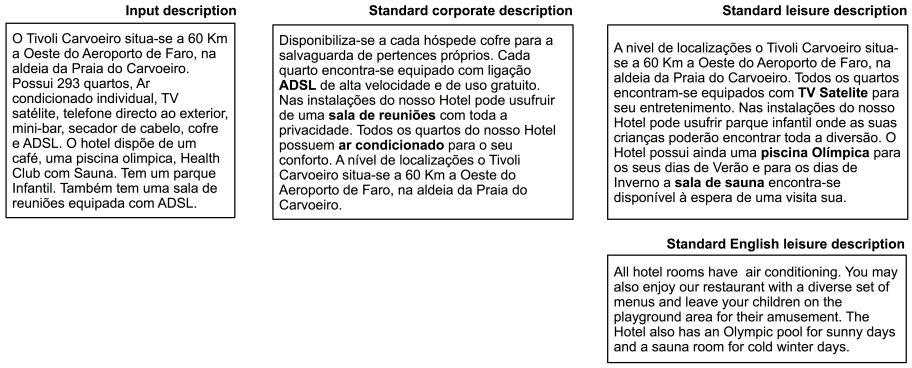

Fig. 3. An hotel description and three flavor descriptions generated by the system.

Results show that the main objective has been reached since it is possible to extract useful information, standardize it and create computable objects from plain text descriptions.

Concerning information extraction, this approach is able to detect relevant types of hotel descriptions, namely hotel services, equipment and location and, for each one, extract the useful features that describe them.

The possibility of having that information structured enables the creation of new added-value services, such as offer refinement search. Further, it provides the automatic creation of suitable descriptions for different market segments anticipating new steps towards a more effective client differentiation.

Regarding future work, and since there is room for improvements on system's various modules, we hope to increase its overall performance.

On the other way, its our aim to construct an hotel database, automatically populated and maintained by the application. This shall be used as a first "out-of-the-box" hotel repository, thus reducing substantially the effort on setting up an hotel infrastructure for commercial business.

Finally, we intend to address the areas of multi-language support (currently only Portuguese descriptions are supported) and normalization of other tourism products, such as rent-a-car and holiday packages.

References

1. Aitken, J.S.: Learning information extraction rules: An inductive logic programming approach. In: van Harmelen, Frank (eds.) ECAI'02 – 15th European Conference on Artificial Intelligence. pp. 355–359. Lyon, France (2002)
2. Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora. pp. 100–110 (1999)
3. Development, H.: Jena – A Semantic Web Framework. *http://jena.sourceforge.net* (March 2010)

4. Freitag, D., McCallum, A.: Information extraction with hmm structures learned by stochastic optimization. In: AI'00 – 17th National Conference on Artificial Intelligence. pp. 584–589. AAAI Press (2000)
5. Grau, J.: Travel Agencies Online. eMarketer (2005)
6. Grishman, R.: Information extraction: Techniques and challenges. In: SCIE'97 – International Summer School on Information Extraction. pp. 10–27. Springer-Verlag (1997)
7. Hobbs, J.R., Bear, J., Israel, D., Tyson, M.: Fastus: A finite-state processor for information extraction from real-world text. In: IJCAI'93 – 13th International Joint Conference on Artificial Intelligence. pp. 1172–1178 (1993)
8. Joachims, T.: Transductive inference for text classification using support vector machines. In: ICML'99 – 16th International Conference on Machine Learning (1999)
9. Klein, D., Manning, C.D.: Conditional structure versus conditional estimation in nlp models. In: ACL'02 – Conference on Empirical methods in Natural Language Processing. pp. 9–16 (2002)
10. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML'01 – 18th International Conference on Machine Learning. pp. 282–289 (2001)
11. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10, 707–710 (1966), originally publish in Russian
12. Martin, A., Przybocki, M. (eds.): 2003 NIST Language Recognition Evaluation (2003)
13. McCallum, A., Nigam, K.: A Comparison of Event Models for Naive Bayes Text Classification. In: AAAI'98 – Workshop on Learning for Text Categorization (1998)
14. Mladenić, D., Grobelnik, M.: Feature selection for unbalanced class distribution and naïve Bayes. In: ICML'99 – 16th International Conference on Machine Learning. pp. 258–267 (1999)
15. Platt, J.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In: Schölkopf, B., Burges, C., , Smola, A. (eds.) Advances in Kernel Methods – Support Vector Learning, pp. 185–208. MIT Press (1999)
16. Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA (1993)
17. Salton, G., Wang, A., Yang, C.: A vector space model for information retrieval. Journal of the American Society for Information Retrieval 18, 613–620 (1975)
18. Schütze, H., Hull, D., Pedersen, J.: A comparison of classifiers and document representations for the routing problem. In: SIGIR'95 – 18th ACM International Conference on Research and Developement in Information Retrieval. pp. 229–237. Seattle, US (1995)
19. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004)
20. Tong, R., Appelbaum, L.: Machine learning for knowledge-based document routing. In: Harman (ed.) TREC'02 – 2nd Text Retrieval Conference (1994)
21. Vapnik, V.: Statistical learning theory. Wiley, NY (1998)
22. ViaTecla: KEYforTravel platform. *http://www.keyfortravel.com* (March 2010)
23. Voorhees, E. (ed.): MUC7, 7th Message Understanding Conference. Science Applications International Corporation (SAIC), Fairfax, Virginia (1998)
24. W3C: OWL Web Ontology Language Guide. textithttp://www.w3.org/TR/owl-guide (March 2010)
25. Witten, I., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco, US, 2nd edn. (2005)