Multiple-Taxonomy Question Classification for Category Search on Faceted Information[*]

David Tomás and José L. Vicedo

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante, Spain
{dtomas,vicedo}@dlsi.ua.es

Abstract. In this paper we present a novel multiple-taxonomy question classification system, facing the challenge of assigning categories in multiple taxonomies to natural language questions. We applied our system to category search on faceted information. The system provides a natural language interface to faceted information, detecting the categories requested by the user and narrowing down the document search space to those documents pertaining to the facet values identified. The system was developed in the framework of language modeling, and the models to detect categories are inferred directly from the corpus of documents.

1 Introduction

From its very beginning, there have been two main paradigms to information search on the web. The first one, *navigational search* (represented by websites like the *Open Directory Project*), helps people to narrow down the general neighborhood of the information they seek using topical directories or taxonomies. The second paradigm, *direct search* (present in websites like *Google*), allows users to write their queries as a set of keywords in a text box to perform *information retrieval* (IR).

Faceted search [9] is a new approach that has recently emerged. This paradigm aims to combine navigational and direct search, allowing users to navigate a multiple-dimensional information space by combining text search with a progressive narrowing of choices in each dimension. Faceted search systems assume that the information is organized into multiple independent facets, rather than a single taxonomy. For instance, we could define for a restaurant guide attributes such as *Cuisine*, *City* or *Features*. These attributes are *facets* that help the users to navigate through them selecting the *values* desired, for instance *Mexican* for *Cuisine*, *Madrid* for *City* or *Online Reservation* for *Features*.

This paradigm is complemented by *category search* [8], which is not a direct search against the information recorded but a search in the space of facet values.

[*] This work has been partially supported by the framework of the project QALL-ME (FP6-IST-033860), which is a 6th Framenwork Research Programme of the European Union (EU), and the Spanish Government, project TEXT-MESS (TIN-2006-15265-C06-01)

While direct search retrieves a set of records[1] that can be further refined using a faceted search approach, category search provides results that are themselves entry points into faceted navigation. In the restaurant guide example, a user would query the system with requests such as "Madrid" or "Italian" to restrict the results to restaurants in this *City* or with this *Cuisine*.

Current interfaces to category search are limited to keyword search on facet values (*categories*). In this paper we propose a novel approach to category search. We face the challenge of identifying facet values requested in natural language questions from the user. We tackle this problem from the point of view of *question classification*, which is the task that, given a question, maps it to different semantic classes. This task has been largely used in the context of *question answering* (QA) [3], where it tries to assign a class or category from a fixed taxonomy to the question in order to semantically constrain the space of valid answers.

While traditional question classification systems are limited to single taxonomy categorization, we introduce the idea of *multiple-taxonomy question classification*. In the context of category search, our question classification system accepts a natural language question from the user and detects the different facets (taxonomies) and values (categories) implicitly requested in the query. The values assigned narrow down the set of relevant documents to those pertaining to the categories identified. Following the previous example, a question like "I'm looking for a Tukish restaurant in Madrid", would set the value of facets *Cuisine* to *Turkish* and *City* to *Madrid*, in order to retrieve only restaurants that fulfill these two constraints.

Our system makes use of *language modeling*. A language model is built for each category based on the document set. To identify categories in the question, the probability of generating it is calculated for each category through its language model. Then several heuristics are applied to determine the final classification. Thus, unlike traditional category search systems we do not limit our search to the list of possible values of the facets, but take advantage of the statistical regularities of the documents classified under these categories. We follow the intuition that words occurring in documents assigned to a category are related to it. Going back to the restaurant guide example, documents describing restaurants categorized as *Mexican* would probably contain words like "burrito", "fajita" or "taco". Moreover, in documents describing restaurants with features like *Reservation*, words like "book" or "reserve" would be common. Thus, our system can interpret a request like "I want to book a table to eat a burrito" and infer that the user is asking for *Cuisine* and *Features* facets with values *Mexican* (triggered by "burrito") and *Reservation* (triggered by "book") respectively.

In the rest of this paper, Section 2 reviews related work. Section 3 describes the language modeling framework. Section 4 depicts the sytem architecture, paying attention on the question processing and the identification of facets and their values. Section 5 describes the corpus employed in the experiments carried out,

[1] Although information search covers different formats like images or sounds, our research is focused on textual information.

the results obtained and the error analysis derived from these results. Finally, conclusions and future work are discussed in Section 6.

2 Related Work

The approach presented in this paper is related to the fields of question classification and faceted information. Question classification has been mainly employed in the field of QA. A majority of systems use hand-crafted rules to identify expected answer types [5]. To overcome the lack of flexibility of these systems, several machine learning approaches have been successfully applied, like *Maximum Entropy* (ME) [1] and *Support Vector Machines* (SVM) [11]. These systems require different levels of linguistic knowledge and tools to learn the classifiers. To avoid this dependence, our approach to question classification is based on statistical language modeling. Unlike other similar approaches [6], we do not need to obtain a training set of questions to build the models as they are built directly from the classified documents.

In [4], Moschitti and Harabagiu presented a novel approach to the task of question classification. Instead of mapping the question to an expected answer type, they assigned document categories from a single taxonomy to questions. They used a set of training questions related to five categories of the *Reuters-21578* text classification benchmark. The idea was to classify questions into these categories in order to filter out all the answers occurring in documents that do not pertain to the category detected. Our system also performs the task of assigning document categories to natural language queries, but extending the classification task to multiple different taxonomies.

In the field of category search on faceted information, current systems [8] perform the task of mapping keywords from the queries to categories. In this sense we go beyond keywords to deal with natural language questions. Moreover, we do not limit the search to category values but take advantage of the faceted documents to infer knowledge to map categories to questions.

3 Language Modeling Framework

Our approach follows the ideas described in [7] for statistical language modeling applied to IR, i.e., to infer a language model for each document and to estimate the probability of generating the query according to each of these models. For a query $q = q_1 q_2 \ldots q_n$ and document $d = d_1 d_2 \ldots d_m$ we want to estimate $p(q|d)$, the probability of the query q given the language model of document d. One important advantage of this framework over previous approaches is its capability of modeling not only documents but also queries directly though statistical language models. This makes it possible to set retrieval parameters automatically and improve retrieval performance through utilization of statistical estimation methods.

In our experiments we follow a unigram language modeling algorithm based on *Kullback-Leibler* (KL) *divergence* [10]. Documents are ranked according to the negative of the divergence of the query language model from the document language model. We employed *Dirichlet prior* as smoothing method. Previous studies [10] suggest that this method surpasses other smoothing strategies when dealing with short queries. This property is interesting as we estimate probabilities for n-grams in the question (this process is described in detail in Section 4).

In our system, in order to detect the categories that occur in a question, we previously grouped all the documents by the values of their facets, obtaining clusters of documents for every category. Then we infer a language model for each of these clusters to estimate the probability of generating the question according to each of these models. We then rank the clusters of documents according to the negative KL-divergence described above and choose the best candidate following some heuristics further detailed in the next section. In this framework, collection statistics such as term frequency, document length and document frequency are integral parts of the language model and are not used heuristically as in many other approaches. Length normalization is implicit in the calculation of the probabilities and does not have to be done in an *ad hoc* manner. In our experiments all the words in the language were indexed, since we do not want to be biased by any artificial choice of stopwords.

4 System Description

Our approach deals with questions on documents that are organized into multiple independent facets, rather than a single taxonomy. The system carries out the task of identifying the set of facet values that occur in a natural language request on faceted documents.

We first group all the documents by the values of their facets obtaining clusters of documents for every category in the corpus. For instance, we would get clusters *Chinese* and *Thai* for facet *Cuisine*, or *Takeout* and *Outdoor Dinning* for facet *Features*. Neither stemming nor stopword removal is carried out in this stage. After this previous process, a language model is derived for each of these clusters. When a question is sent to the system, all the n-grams of the question are extracted and the probability of generating each one according to the models is estimated, following the aforementioned language modeling framework. We use the *Lemur toolkit*[2] to perform this task.

As we said before, the underlying idea of our approach is that content of the documents is related to the categories assigned to these documents. For example, words like "pizza" or "risotto" will commonly appear in the description of restaurants with value *Italian* assigned to facet *Cuisine*. A request like "I want a pizza" will promote the cluster of *Italian* cuisine in the language modeling retrieval framework over other clusters, thus detecting this category in the question. So we do not only search on the values of the facets to detect categories, but also the contents of the documents classified in these categories.

[2] http://www.lemurproject.org

Another assumption is that, as facets are orthogonal sets of values [2], we consider that one n-gram from the question can only determine one facet, i.e., "pizza" can not determine values on different facets as *Cuisine* or *City* at the same time. Otherwise, it could be possible to obtain different values for the same facet (*Italian* and *Greek* for facet *Cuisine*), but in our system hard classification is performed allowing only one possible value per facet.

To process the question in the system and obtain multiple-taxonomy classification, we define the following algorithm:

Compute similarity: Compare each n-gram ($n \leq 3$ in our experiments) from the question with each facet value in the corpus. If there is an exact match, the category value is assigned to that n-gram with the maximum ranking value (0, as we employ negative KL-divergence). These first steps perform the classic approach to category search. If no match is found, the similarity with the models estimated for every category is computed on each n-gram, and ranking values obtained are stored.

Stopword removal: Fisrt, In order to detect which n-grams in the query behave as stopwords (and should not be taken into account for the classification task), each unigram is treated in isolation, measuring the variance of the ranking values obtained in the previous step. We consider as stopwords all the unigrams that present close similarity values for every model of the clusters, indicating that they are almost equally distributed through the corpus and cannot discriminate between categories. An empirical variance threshold is established to detect these stopwords. All the remaining n-grams (bigrams and trigrams) that start or end with a unigram that is a stopword are also considered as stopwords.

Category selection: For all the n-grams not labeled as stopwords, the best category detected in the ranking process is stored with its corresponding weight. If two n-grams have the same best category, their weights are compared: the n-gram with the greatest weight keeps the category assigned while the other is discarded.

Category assignment: Finally, we get a set of n-grams with the assigned categories. These categories are finally associated to the question if the weight given by the similarity of the language model for these n-grams is over a threshold empirically set.

5 Experiments

This section describes the corpus of documents and set of questions used in the test carried out, the experiments and results obtained and the error analysis derived from these results.

5.1 Dataset

In order to test the system, we created a corpus of documents extracting information from *Lanetro*[3], a website specialized in tourist information about places

[3] http://www.lanetro.com

and events, that offers a faceted search interface for browsing the data. From this web we collected all the data related to restaurants. We obtained 2,146 documents in Spanish, one for each restaurant, with information about the street address, the telephone number and a brief textual description of the restaurant (about 50 words on average). Every document was originally classified in four different facets: *City* (the city where the restaurant is located), *Average price* (average price per person), *Features* (such as *Open Late*, *Romantic*, *Delivery Available*...) and *Cuisine* (*Chinese*, *Italian*, *Seafood*...). There are 77 possible values for facet *City*, 5 for facet *Average price*, 21 for *Features* and 87 for *Cuisine*.

In addition to the corpus of documents, we created a test set of questions in Spanish gathered from potential users outside our project. Eight different users were asked to formulate ten call centre-like free questions. The only restriction imposed to the users was that they must ask questions that should have as answers a restaurant or a list of restaurants. Thus we obtained eighty different natural language questions with a significant variety of utterances, from "Want a Kebab" to "I'm in Alicante. I'm vegetarian and I'm looking for a cheap place that fits my needs". All the questions in the test set were labeled by two assessors that assigned values to the facets present in the questions and detected those facets that did not occur.

Users were not informed about the facets involved in the experiments or its possible values, as users of real systems do not necessary know the multiple taxonomies present in this type of systems. This way we also wanted to test the robustness of the approach detecting existing features and values, and also discarding those that do not exist in the taxonomies.

5.2 Evaluation and Results

We tested the system on the eighty questions described above. We carried out three different experiments in order to compare the performance of the language modeling framework with other traditional IR empirical methods. For this purpose, we computed the similarity between the questions and the documents also with *TF-IDF* and *Okapi*. In the language modeling experiments, we used Dirichlet prior smoothing with prior parameter μ empirically set to 3500.

Three different measures of performance are defined. P represents the precision detecting categories in the questions, that is, the number of categories present in the questions that where correctly detected from the total number of categories present. $P_{\emptyset}$ indicates the precision of the system detecting absent facets in the questions, i.e., if there are restrictions on facets or not. This value is the number of facets not present in the questions that were correctly identified as absent. Finally, P_T is the total number of categories correctly detected in the questions plus the number of facets correctly detected as not present, divided by the total number of facets in the questions.

Table 1 presents the results obtained for the three approaches mentioned above: language modeling, TF-IDF and Okapi. The results are detailed for each of the facets. Language modeling achieved the best overall result in these experiments, with a value of $P_T = 0.6500$. It also achieved the best value of P for all the facets, while best values for $P_{\emptyset}$ are more distributed through the three approaches.

Table 1. Detailed results for *language modeling*, *Okapi* and *TF-IDF*.

	LM			Okapi			TF-IDF		
Facets	P	$P_{\emptyset}$	P_T	P	$P_{\emptyset}$	P_T	P	$P_{\emptyset}$	P_T
City	**0.8621**	**0.5686**	**0.6750**	0.5862	0.4902	0.5250	0.1724	0.0588	0.1000
Average price	**0.4000**	0.9467	0.9125	**0.4000**	**0.9733**	**0.9375**	**0.4000**	0.7067	0.6875
Features	**0.3462**	**0.7407**	0.6125	0.2692	0.8148	**0.6375**	0.1923	0.6852	0.5250
Cuisine	**0.6250**	0.0625	0.4000	0.3125	0.0313	0.2000	0.0833	**1.000**	**0.4500**
Overall	**0.6111**	0.6698	**0.6500**	0.3796	**0.6745**	0.5750	0.1481	0.5896	0.4406

We can conclude from this results that language modeling offers more robustness and precision through all the facets. These values are coherent with the results obtained in [10] for the task of IR. Okapi obtains the second best overall precision ($P_T = 0.5750$) and also obtains the best results for facets *Average price* and *Features*. TF-IDF obtained the worst results with $P_T = 0.4406$.

5.3 Error Analysis

The first problem detected is due to verbosity in questions. A request such as "Could you recommend me a restaurant recently opened specialized in Asian cuisine?" presents many n-grams not related to any facet (like "recently opened"). These terms introduce noise in the detection of facet values.

Another problem is sparse facets, which severely harms the performance of statistical methods. There are facets that present many possible values (87 for *Cuisine*) and few documents classified in each value (there are only two restaurants offering *Vietnamese* cuisine in our corpus). This makes necessary the increase of data to predict the models.

As we said before, we do not perform preprocessing of the corpus, not even a stemming process. This way, terms like "burger" and "burgers" are considered completely different. Performing stemming on the data would solve this problem.

Finally, the way the set of test questions was built results a bit risky. We wanted to set a real open domain environment, so that questions were uttered freely with no restriction on facet values. Thus, many requests related to "anniversaries" or "weddings" occur, while no categories in our system match these requests. This results in an increase of the noise introduced into the system.

6 Conclusions and Future Work

In this paper we presented a novel approach to multiple-taxonomy question classification. The system proposed receives a natural language question and maps it to categories in different taxonomies. In our experiments we used this system as a natural language interface to category search on faceted data, allowing the user to formulate a free question to narrow down the set of candidate relevant documents to its query. We tested the system on a corpus of faceted documents describing restaurants. We gathered questions from potential users in other to build a corpus of real test questions.

The system was built on a language modeling framework that demonstrated to perform better than other traditional approaches to IR, like Okapi or TF-IDF. All the information to build the models was obtained from the corpus of documents. Thus, we do not need any linguistic knowledge or tools but statistical information. The results obtained are promising for this novel task as free questions made this type of multiple-classification a hard problem. We obtained a best performance $P = 0.6500$ in detecting categories for language modeling.

Error analysis revealed that some improvements must be done. Adding stemming to the preprocessing of the corpus and improving the stopword detection algorithm could easily rise the overall performance. More improvement in the system could be introduced by expanding the original unigram model approach to language modeling with a more complex one based on bigrams or trigrams.

As the system does not employ linguistic tools, there is room for much improvement in this field. For instance, we could use WordNet in order to expand the terms of the question and increase the recall of categories.

References

1. P. Blunsom, K. Kocik, and J. R. Curran. Question classification with log-linear models. In *SIGIR*. ACM, 2006.
2. W. Denton. How to make a faceted classification and put it on the web. http://www.miskatonic.org/library/facet-web-howto.html, November 2003.
3. X. Li and D. Roth. Learning question classifiers. In *International Conference on Computational Linguistics*, 2002.
4. A. Moschitti and S. Harabagiu. A novel approach to focus identification in question/answering systems. In *HLT-NAACL Workshop on Pragmatics of Question Answering*, 2004.
5. M. A. Pasca and S. M. Harabagiu. High performance question/answering. In *SIGIR*, pages 336–374. ACM, 2001.
6. D. Pinto, M. Branstein, R. Coleman, W. B. Croft, M. King, W. Li, and X. Wei. Quasm:Â a system for question answering using semi-structured data. In *2nd ACM/IEEE-CS Joint Conference on Digital libraries*, pages 46–55, 2002.
7. J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR*, pages 275–281. ACM, 1998.
8. D. Tunkelang. Dynamic category sets: An approach for faceted search. In *SIGIR Workshop on Faceted Search*. ACM, 2006.
9. K. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. In *CHI*. ACM, 2003.
10. C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, 2004.
11. D. Zhang and W. S. Lee. Question classification using support vector machines. In *SIGIR*. ACM, 2003.