

# Appetitoso: A Search Engine for Restaurant Retrieval based on Dishes

Gianni Barlacchi<sup>1,2</sup>, Azad Abad<sup>1</sup>, Emanuele Rossinelli<sup>3</sup>, Alessandro Moschitti<sup>1,4</sup>

<sup>1</sup>Department of Information Engineering and Computer Science, University of Trento

<sup>2</sup>TIM Semantics and Knowledge Innovation Lab, Trento

<sup>3</sup>Kloevolution S.r.l.

<sup>4</sup>Qatar Computing Research Institute

{gianni.barlacchi,e.rossinelli,amoschitti}@gmail.com

azad.abad@unitn.it

## Abstract

**English.** Recent years have seen an impressive development and diffusion of web applications to food domains, e.g., Yelp, TripAdvisor. These mainly exploit text for searching and retrieving food facilities, e.g., *restaurants, caffè, pizzerias*. The main features of such applications are: the location and quality of the facilities, where quality is extrapolated by the users' reviews. More recent options also enable search based on restaurant categorization, e.g., *Japanese, Italian, Mexican*. In this work, we introduce Appetitoso<sup>1</sup>, an innovative approach for finding restaurants based on the dishes a user would like to taste rather than using the name of food facilities or their broad categories.

**Italiano.** Recentemente si è assistito ad un impressionante sviluppo e diffusione di applicazioni web per il dominio del cibo, e.g., *Yelp, TripAdvisor*. Queste sfruttano principalmente il testo per la ricerca e il recupero di punti di ristoro, e.g., *ristoranti, bar, pizzerie*. Le caratteristiche principali usate dalle applicazioni sono: la posizione e la qualità delle strutture che servono il cibo, dove la qualità è estrapolata dalle recensioni degli utenti. Opzioni più recenti consentono anche la ricerca in base alla categoria del ristorante, e.g., *Giapponese, Italiano, Messicano*. Questo articolo introduce Appetitoso, un nuovo modo di trovare punti di ristoro sulla base dei piatti che il cliente vuole gustare invece che sul nome del ristorante o le sue grossolane categorie.

## 1 Introduction

In late 2000's, we assisted to the explosion of TripAdvisor<sup>2</sup>, the world's largest travel site, which offers advice about hotel and restaurants. In few years, it has revolutionized the restaurant industry, allowing its users to search restaurants by location and broad food categories, e.g., Mexican, Italian, French, etc., and the reviews and ratings provided by other users.

However, the user expectation has evolved over-time: looking for restaurants is not enough anymore, people are now considering finer-grained properties of food, e.g., a particular way to cook a dish along with its specific ingredients. Thus, there is a clear gap between what the market proposes and the emerging trends.

In this work, we present Appetitoso, a search engine that seeks for restaurants based on dishes. This approach is designed to help users to find their restaurants having already a specific dish preference in mind, using fine-grained properties of the dish.

Appetitoso integrates state-of-the-art search engines, such as BM25, with a domain specific knowledge base describing properties and similarity relations between different Italian dishes. This knowledge is very useful, e.g., in our experiments, we show that it greatly boosts dish retrieval. Appetitoso is available as a mobile phone application (e.g., Android and iOS) and website, released in 2014 for two languages, English and Italian. It is an end-to-end application for finding restaurants offering the desired dish. We evaluated it using a set of 666 popular queries typed by its users of in the cities of Rome, Milan and Florence.

In the reminder of this paper, in Section 2, we report related work on systems for automatic food recommendation, In Section 3, we introduce Appetitoso, its knowledge base and food search engine. Section 4, we describe our experiments on

<sup>1</sup><http://www.appetitoso.it>

<sup>2</sup><http://www.tripadvisor.com>

restaurant retrieval on Italian language and finally, in Section 5, we provide our conclusion.

## 2 Related Work

Nowdays, the importance of data analysis is becoming fundamental in many fields. From telecommunications to social media, the huge amount of available data allows scientists and researchers to address previously unsolved problems (Barlacchi et al., 2015). The food domain represents one of the field in which emerging big data techniques demonstrated to be very promising and able to impact the every daily life of people. In recipe recommendation, for instance, Teng et al. (2012) proposed an approach based on networks of ingredients, which has been built from a dataset of recipes. In order to capture both ingredient relations and users’ knowledge for combining ingredients in new recepies, they created two separate networks used for recipe recommendation.

Moreover, Ahn et al. (2011) explored the impact of flavor compounds on ingredient combinations through a network-based approach. An interesting application was developed by IBM with Chef Watson<sup>3</sup>, which is part of the cognitive computing applications developed by the company. The system models the chemical compounds of different ingredients together with textual information extracted from thousands recipes for suggesting new ones using new ingredient combinations.

Among the different kinds of data, text surely represents one of the richest sources of information from which we can extract a wide range of statements about food. The use of text in food domain has been widely explored showing promising results with different models, ranging from the measurement of sentiment in food reviews (Kang et al., 2012) and relation extraction (Wiegand and Klakow, 2013; Wiegand et al., 2012), to the prediction of attribute reviews in recipes (Druck, 2013).

## 3 Appetitoso

We introduce the idea of searching a dish and then finding the restaurant that can offer it. Thus, the aim of our search engine, Appetitoso, is to find the best restaurants offering dishes relevant to the user’s request. Starting from a query with food-related content, e.g., *bistecca alla fiorentina* (t-bone steak), the system retrieves places that sat-

<sup>3</sup><https://www.ibmchefwatson.com>

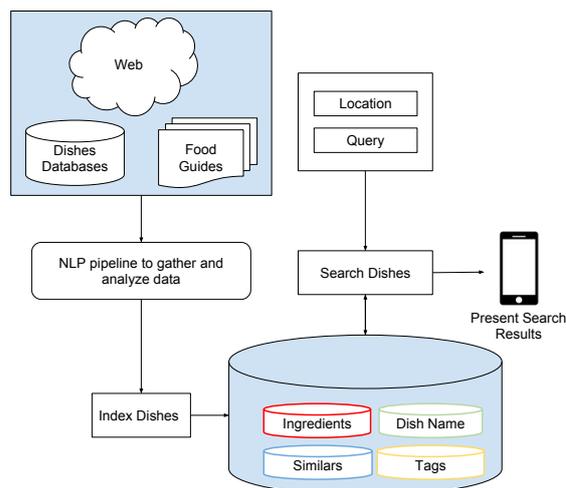


Figure 1: Architecture of Appetitoso.

isfy the constraint on the location and, at the same time, prepare the desired dish or similar dishes.

Appetitoso retrieves restaurants from a semistructured database, Food Taste Knowledge Base (FKB), which contains text descriptions of dishes and restaurants: we in part manually inserted them or gathered them from various sources such as foodblogs, restaurants reviews and food guides. The search processes is divided in two phases: first, the user has to type the query and a location, e.g., the address of a target place or the current user position captured by GPS. These are both sent to the Appetitoso’s search engine, which retrieves a list of related dishes from FKB. The results are grouped by dish name and shown to the user in different *course* categories, i.e., *antipastolentree*, *primol*/first course, *secondol*/second course, dessert. The input location is used to restrict the search area of interest, relying on the restaurant position available in FKB.

The second phase of the searching process is devoted to select the best restaurant. Once the user chooses a dish from the list above, Appetitoso provides a list of restaurants that offer such food speciality. Indeed, all the restaurants offering that dish (possibly named in many ways) are stored in FBK. Additionally, Appetitoso provides a *DishScore*<sup>4</sup> for each restaurant, which is a measure of the goodness of the dish in that restaurant. Fig. 1 shows the high-level architecture of the sys-

<sup>4</sup>We only inserted restaurant that have a good reputation in FBK. In order to generate the *DishScore*, we trained a logistic regression over 5 different review scores, e.g., 1 star, 2 star etc. We used various features, e.g., Tripadvisor and food guide scores. This description is however beyond the purpose of the current paper.

tem. In the next section, we illustrate our FKB, which enables accurate retrieval of similar dishes.

### 3.1 The Food Taste Knowledge Base (FKB)

A quick analysis of Italian menus clearly show that, in many cases, the name of a dish is not enough to understand its content, which means that names do not support an accurate similarity measures between dishes. Thus, we created FKB, which also organizes dishes in a hierarchical structure, where each node is connected to others in case there is a similarity between them.

For instance, *Bucatini alla amatriciana* (bucatini with amatriciana sauce) can be extended from *Spaghetti alla amatriciana* (spaghetti with amatriciana sauce) since the only difference between the two dishes is the type of pasta (*spaghetti* vs. *bucatini*). In this case, we marked the first dish as a template for the second one. The relation is one-to-many: one dish can be a template for many others but it can be only assigned to one template.

Since there is no defined way to assess the similarity between two dishes: they may be similar as they are made by similar ingredients or because they are cooked in the same way, we built the FKB hierarchy with a semi-automatic approach. We used name similarity to select similar candidates, which are then manually annotated by food experts. We manually populated FKB with data collected from the web, food guides and foodblogs. Every dish belongs to a restaurant is represented by means of the following information:

- ID: unique identifier for the dish.
- Name of the dish: the name of the dish as reported in the restaurant menu.
- Ingredients: list of the principal ingredients. When the ingredients are not provided by the restaurant, we use a list of common ingredients for the dish (e.g., ingredients from online recipes).
- Tags: list of tags useful to characterize the dish. The tag list does not include ingredients but only categorical information that can help to characterize the dish (e.g., meat or fish).
- Similar dishes: list of similar dishes defined according to our hierarchy described above.
- Template: ID of the template dish, if it is present.
- Restaurant: information about the restaurant that cook this dish.
- DishScore: a value that indicates the goodness of the dish. It is calculated taking into account many factors such as the reputation of the restaurant in



Figure 2: Connection between similar fish dishes.

cooking that dish, the number of mentions in food guide and the sentiment extracted from foodblogger articles and restaurant reviews.

This hierarchical organization is very powerful and allow us to easily keep track of similarities that are not explicit. Fig. 2 shows an example of connections between similar dishes. It is worth to mention that Appetitoso aims to suggest only restaurants that own a good reputation in cooking target dishes, i.e., restaurants in Rome that are famous for *pasta alla carbonara*. Consequently, this limits the number of dishes contained in the FKB and thus on the territory coverage. On the other hand, it makes it possible to create a manually checked resource.

### 3.2 Dish Retrieval

Italy has long and variegated traditions on preparing food: it is possible to find different kinds of cuisine even in nearby cities. This makes the Italian food incredibly varied and fascinating, but, at the same time, difficult to interpret from a linguistic viewpoint. As a consequence the underlying retrieval problem cannot be addressed by just using a simple word matching approach. Indeed, even if a dish is conceptually the same of another, different restaurants (e.g., in different locations) have their own way to call it.

To tackle the problem above, we verified the hypothesis that a search engine can achieve a better result if we consider further information such as ingredients and tags. This approach significantly improves the quality of the retrieved list comparing to the simple word matching approach.

More specifically, we applied BM25 (Robertson et al., 1995) to FKB. Given a dish query,  $Q$  and a representation of a candidate dish,  $D$ , BM25 ranks the latter according to the following score:

$$s(Q, D) = \sum_{i=0}^n \frac{IDF(q_i) \cdot ((k+1) \cdot TF(q_i, D))}{\left(k \cdot (1 - b + b \cdot \frac{|D|}{avgD}) + TF(q_i, D)\right)},$$

where  $k$  and  $b$  are two free parameters that modify respectively the impact of term frequency (TF) and the document length through the term  $\frac{|D|}{avgD}$ ,  $|D|$  is the document length and  $avgD$ , i.e., the average of  $D$  over the whole dataset. Finally,  $IDF(q_i)$  is the Inverse Document Frequency for the query term  $q_i$ , computed as:

$$\log \left( 1 + \frac{(N - DF(q_i) + 0.5)}{(DF(q_i) + 0.5)} \right),$$

where  $N$  is the total number of documents in the collection, and  $DF(q_i)$  is the document frequency of the term  $q_i$ .

Additionally, we created four different indexes<sup>5</sup> with the information contained in FKB, i.e., the (i) dish name, (ii) ingredients, (iii) tags and (iv) similar dishes. Each list is built using the words describing the four items above. Thus, when we query a dish, we first retrieve four different set of results and then, since they have different importance, we combine them together assigning different weights, where the latter are set using cross-fold validation.

## 4 Experiments

Our experiments aim at demonstrating the effectiveness of our models on the task of dish retrieval. We used the well known metrics: Precision at rank 1 (P@1), Mean Reciprocal Rank (MRR) and Mean Average Precision (MAP). P@1 indicates the percentage of queries with a correct answer (e.g., the desired dish) found in the first position. The MRR is computed as follows:  $MRR = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \frac{1}{rank(q)}$ , where  $rank(q)$  is the position of the first correct answer in the retrieved list. For a group of queries  $Q$ , MAP is the mean over the average precision scores for each query:  $\frac{1}{|Q|} \sum_{q=1}^{|Q|} AveP(q)$ .

Due to the fact that FKB contains multiple instances of the same dish, we evaluated the collapsed list of results by considering the dish name. It is worth to mention that the name of the dishes are not standard, thus some dishes are the same still having slightly different names. To make them more similar, we normalized name forms by removing space, articles and punctuation. We considered a set of 547 popular queries typed by users

<sup>5</sup>We use Lucene (McCandless et al., 2010)

Model	City	MRR	MAP	P@1
<b>Baselines</b>				
String Matching (on entire names)	Milan	53.28	53.28	53.28
	Rome	71.23	71.23	71.23
	Florence	44.87	44.87	44.87
	<b>All</b>	<b>56.46</b>	<b>56.46</b>	<b>56.46</b>
BM25 (on names only)	Milan	69.75	65.44	68.18
	Rome	63.86	60.32	58.90
	Florence	42.31	40.94	37.18
	<b>All</b>	<b>58.64</b>	<b>55.56</b>	<b>54.75</b>
<b>Our Model</b>				
Appetitoso (names, ingredients tags, similar names)	Milan	95.35	85.69	93.43
	Rome	87.40	76.23	84.93
	Florence	83.55	75.38	78.21
	<b>All</b>	<b>88.76</b>	<b>79.10</b>	<b>85.52</b>

Table 1: Ranking evaluation for different models

in Milan (396 queries), Rome (73 queries) and Florence (78 queries). The number of retrieved dishes varies for the different queries with averages of 22.8, 22.3 and 37.4 for Florence, Milan and Rome, respectively. For each retrieved dish, we manually annotated the relevance respect to the input query. It should be noted that the same dish is associated (in FKB) with all the restaurants offering. Thus, restaurants retrieval is a side effect of dish retrieval.

We considered two baselines for evaluating our model, namely, String Matching and BM25. The first is base on simple string matching between the query and the dish names. The second is BM25, which can be applied to dish names only or to the 4 indexes described in described in Sec. 3.2.

Table 1 shows the results of the baselines and our model by cities and overall (All). Appetitoso largely outperforms String Matching and BM25 applied to names only, e.g., up to 32 and 24 absolute percent points in MRR and MAP, respectively.

## 5 Conclusion

In this paper we presented Appetitoso, a semantic search engine for food. The aims of the search engine is to provide the users with a way of searching restaurants by dishes rather than just using the restaurants' address or cuisine type. We show that, given the complexity of dish naming, semistructured database for dishes largely improve BM25. Overall, Apetitoso shows good performance, e.g., achieving 88.76% in MAP. In the future, we would like to include more complex unstructured data such as the description of the dishes and also explore the possibility of word embeddings for the food domain.

## References

- Yong-Yeol Ahn, Sebastian E Ahnert, James P Bagrow, and Albert-László Barabási. 2011. Flavor network and the principles of food pairing. *Scientific reports*, 1.
- Gianni Barlacchi, Massimo Nicosia, and Alessandro Moschitti. 2014. A retrieval model for automatic resolution of crossword puzzles in italian language. In *The First Italian Conference on Computational Linguistics CLiC-it 2014*, page 33.
- Gianni Barlacchi, Marco De Nadai, Roberto Larcher, Antonio Casella, Cristiana Chitic, Giovanni Torrìsi, Fabrizio Antonelli, Alessandro Vespignani, Alex Pentland, and Bruno Lepri. 2015. A multi-source dataset of urban life in the city of milan and the province of trentino. *Scientific data*, 2.
- Gregory Druck. 2013. Recipe attribute prediction using review text as supervision. In *Cooking with Computers 2013, IJCAI workshop*.
- Yi Fang, Naveen Somasundaram, Luo Si, Jeongwoo Ko, and Aditya P Mathur. 2011. Analysis of an expert search query log. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 1189–1190. ACM.
- Hanhoon Kang, Seong Joon Yoo, and Dongil Han. 2012. Senti-lexicon and improved naïve bayes algorithms for sentiment analysis of restaurant reviews. *Expert Systems with Applications*, 39(5):6000–6010.
- Yuanhua Lv and ChengXiang Zhai. 2011. When documents are very long, bm25 fails! In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 1103–1104, New York, NY, USA. ACM.
- Michael McCandless, Erik Hatcher, and Otis Gospodnetic. 2010. *Lucene in Action, Second Edition: Covers Apache Lucene 3.0*. Manning Publications Co., Greenwich, CT, USA.
- Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *NIST SPECIAL PUBLICATION SP*, 109:109.
- Chun-Yuen Teng, Yu-Ru Lin, and Lada A Adamic. 2012. Recipe recommendation using ingredient networks. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 298–307. ACM.
- Michael Wiegand and Dietrich Klakow. 2013. Towards the detection of reliable food-health relationships. *NAACL 2013*, page 69.
- Michael Wiegand, Benjamin Roth, and Dietrich Klakow. 2012. Data-driven knowledge extraction for the food domain. In *KONVENS*, pages 21–29.