We do not need hasCity, hasState inside those entities

Exploratory data analysis has shown that the name doesn’t uniquely identify the site since there are a few site with multiple addresses (e.g. post codes) and each address has it’s own set of menu items.

For instance, the site name Bertucci's appears with 4 different addresses and although three of them appear to be of the same franchise having similar menu items with the same description and price, one of the three is missing the Margherita Pizza. Moreover the 4th one, the one in Illinois seems to be a completely different restaurant since it only has one menu item that is different from the menu items of the other 3 Bertucci’s.

The above observations have highlighted the need to combine the name with at least one more column before we can create a new instance of a site otherwise the data would incorrectly assume that because the name is the same all 4 Bertucci's sites are actually the same site.

Also it has rendered the assumption that all sites sharing the same name are part of the same franchise/chain incorrect.

Currency: even though the currency logically belongs with the price, we are making a modelling assumption that it actually is a property of the site, since a site cannot be selling items at different currencies. An even more generic assumption would be to make the currency a property of the country the site is in but since we only have data from the US and all currencies where specified are USD we don’t have more evidence that that assumption would be correct.

Items without a price will be processed a zero price items

No point in having a menu item or an address without a restaurant so

If restaurant

If menu\_item

2.3.

I’ve created the classStringToURI = dict() as a dictionary of the format

{'class':{'string': uri}}

The first key will be used to denote the class we are creating a URI for and the second would be a dictionary of key value pairs. That is because we may have the same value for a restaurant and a pizza for instance so we do not want to mix the two.

Some of the Cities are not correct, Manchester… Universities… etc.

It was also decided to make the address an entity of each own. A simpler approach would be to make all properties of the address properties of the restaurant class. However, having the address being its own class, makes the model more flexible, in case an address may house multiple restaurants (or generally businesses). There are a few cases in the data we have that the same address houses two places:

e.g. 400 S Orlando Ave, 32751, FL is the address of ‘Nypd Pizza’ as well as ‘Francesco's Ristorante & Pizzeria’.

In order to decide create a unique URI for the address we concatenate the address like with the state column. That is because there is the case of the following two addresses that share the same address line but see to be different addresses altogether.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 222 E Main St | Mount Kisco | US | 10549 | NY |
| 222 E Main St | Collegeville | US | 19426 | Rahns |

For the categories we performed a basic NLP for keyword extraction in order to identify the most frequent terms in the ‘categories’ column. The insights from the NLP results informed the creation of some subclasses of the restaurant class. The list of subclasses is not exhaustive and it was created after manually inspecting the extracted keywords. For the NLP we allowed up to 3 word per expression and also allowed for the same word to appear in multiple expressions. We also allowed for the restaurant subclasses in the ontology to be NOT be disjoined since a lot of restaurants in the csv have multiple categories.

We follow a similar logic for the pizza subclasses

Reasoning

Originally made price a functional property

For the average price I grouped by currency as well. In this case where all prices are in USD that grouping is redundant. However if there were prices in other currencies as well then this grouping would ensure that only same currency prices would be summed up. In such a case we would get multiple results (i.e. one per currency).