9th International Conference on Prestigious Applications of Intelligent Systems – PAIS@ECAI2020
Santiago de Compostela, Spain

# Point-Of-Interest Semantic Tag Completion in a Global Crowdsourced Search-and-Discovery Database

**Nikolaos Lagos**[1] and **Salah Ait-Mokhtar** and **Ioan Calapodescu**

**Abstract.** Applications that process Point-of-Interest data are omnipresent nowadays. They range from digital maps to recommender systems for places to visit, and personal assistants. The success of such applications critically depends on the quality of the ingested data. However, corresponding databases, especially when they are crowdsourced, are often incomplete. Existing work on automatic data completion approaches has only partially considered the task for Points-of-Interest (POI). Such entities have a number of distinctive properties - notably multiscript names, geo-spatial identity, and temporally defined context -, which make the task more challenging. Here we present an approach to automatically complete POI semantic tags in a crowdsourced database. We perform experiments on multi-lingual data from Foursquare, a global location-based social network, and observe that (i) POI names are strong predictors of POI semantic tags: a character-based LSTM model trained only on POI names gives 72.5% worst-case micro-precision and 50.38% micro-F1 scores, (ii) appropriate use of spatio-temporal data leads to consistent improvements (iii) using a structured representation of time gives higher precision and requires less computation time than string-based LSTM variants, however, the higher precision is achieved at a cost of lower recall and micro-F1, (iv) an LSTM model trained on semi-structured strings representing time, is competitive to fully structured inputs in terms of recall.

## 1 INTRODUCTION

Points of Interest (POIs) can be described by a number of semantic tags e.g. Falafel Restaurant, Bowling Alley etc. In Foursquare, users select the tags from a hierarchical list of more than 900 categories[2].

Semantic tags are not only used to guide humans but also as input data to several applications such as recommender systems, trip planners, and/or artificial intelligence (AI) based personal assistants. The success of such applications critically depends on the quality of the ingested data, and most importantly the completeness of supporting databases [2, 13].

In the last few years, with the advent of Location-Based Social Networks (LBSNs), such as Foursquare, a lot of POI databases are crowdsourced, which makes the problem of data completion even more important. A number of works have been proposed to automatically categorise POIs [6, 9, 19, 20, 23] but suggested techniques are usually applied on a small sample of POI categories and focus on limited geographical areas (i.e. cities or regions from a single country). Moreover, previous work is based on user data and most specifically the notion of check-ins i.e. visits to a POI explicitly declared by the user where geo-coordinates and time of visit are recorded. Gaining access to such user data may become more difficult with new privacy-related laws, such as EU's General Data Protection Regulation (GDPR) being applied (c.f. for a detailed analysis see Section 2).

The main contributions of this work are as follows.

- To our knowledge this is the first study of a multi-lingual, global search-and-discovery database related to POI tag completion. We formally define the problem and present a corresponding analysis.
- We propose that, as POIs are inherently not only geo-spatial but also temporally defined entities, we should exploit temporal information extracted from POI attributes e.g. opening hours. Specialised processing is thus proposed.
- In contrast to previous work, we use only publicly available data about POIs. This work can thus complement previous techniques based on user check-ins, but can also be independently used in the case that no such data is available.

The rest of the paper is organised as follows. We review related work in Section 2. We define the problem in Section 3 and describe our data completion method in Section 4. Experiments are presented in Sections 5 and 6. Section 7 includes the conclusions of this work.

### 1.1 Industrial context

Our company Naver, provides, among other things, location-based services. Good quality POI data is thus of major importance. In this context, in Naver Labs Europe, we have been exploring automatic multi-lingual methods for completing and correcting POI semantic tags found in Foursquare's database, a global crowdsourced location-based social network[3]. The scope of our work is to support:

- A user searching for specific type of POIs in the vicinity of her/his position. If POIs are not categorised under the appropriate type, the user can not easily search for them and they will not be included in the search results. In addition, proper POI categorisation could also help in recommending possible alternatives.
- A user writing a review on a POI. Selecting the right POI metadata when completing the review is time-consuming. We want to automatically recommend appropriate metadata, including POI categories, to facilitate the task as much as possible.

## 2 RELATED WORK

Most of the work on Point-of-Interest categorisation has taken place in the context of Location-Based Social Networks. There are two main approaches to the problem.

[1] Naver Labs Europe, France, email: nikolaos.lagos@naverlabs.com
[2] https://developer.foursquare.com/docs/resources/categories as of 3rd October 2018
[3] We got access to this data thanks to an agreement between Naver Labs and Foursquare.

The first one requires access to check-in data and uses only such data as input to the prediction model [20, 19, 9]. This includes for instance POI unique identifiers, user unique identifiers, the time and duration of the check-in, the number of check-ins, the latitude/longitude of the user's position, and sometimes users' demographic information (e.g. age range, gender). Based on this information, most of the existing work, attempts to categorise POIs in very coarse-grained categories (e.g. home vs. work, or nightlife/bar vs. restaurant) with the no. of categories to predict ranging from 3 to 15.

The second one is represented by He et al. [6] and Zhou et al. [23], where the authors, in addition to check-ins, also try to use more fine-grained information about the POIs. In the case of He et al. [6] this includes general tags that may be related to categories but also to other information e.g. "godzilla". Zhou et al. [23] are the first that use the POI name and address tokens or more particularly token embeddings pre-computed on a domain and language-specific corpus. However, they consider POI data as standard textual content, and perform token-based processing. However, corresponding databases are often global, multi-lingual, and multi-script. This requires a different approach to token-based representations. Although character-based models are not new, we are the first ones to use them in this context. In addition, we believe that there is a spatio-temporal aspect inherent to the identity of POIs, including their opening/access times. We thus propose different ways to represent temporal information.

With a different objective but in a related context, Jiang et al. [7] apply machine classification techniques to the problem of fusing different POI databases under a common classification hierarchy, the North American Industry Classification System (NAICS). Their study involves only a few American towns and they do not use POI attributes as input features.

A number of works have been carried out on location prediction in social streams, e.g. Twitter. The main research interest is in using noisy and short text for classification. For instance, Cano et al. [4] uses tweets to infer volatile POI classes according to specific temporary events happening at a specific location. Interested readers may refer to Zheng et al. [22] for a comprehensive survey of the domain. Despite superficial commonalities, this subject is different from the one studied in this paper.

## 3 PROBLEM DEFINITION

Our goal is to complete POIs' semantic tags in the dataset. For instance, a typical place found in Foursquare's database is "Παραδοσιακό μπουγατσατζίδικο Μπαντής"[4] with opening times "6:30-15:00" and latitude/longtitude "40.647039/22.938023". The only category attributed to the POI in the database is *Bougatsa Place*. However, missing, but pertinent, tags include *Breakfast Spot*, *Pastry Shop*, and *Snack Place*. The objective is to complete such missing tags.

We consider POI semantic tag completion as the problem of completing a specific attribute of the dataset, the one that represents POIs' categories, based on data from the remaining attributes.

Formally, a POI $p$ should consist of an attribute that includes an ideal, complete, category labelset i.e. set of relevant labels $L \subset \Lambda$, where $\Lambda = l_1, ..., l_m$ is the set of all possible labels, and other attributes represented by the set $A$. In this work, the set of labels attributed to each POI in the dataset i.e. the observed labelset, is incomplete. So, if $L_o$ represents the observed labelset then $L_o \subset L \subset \Lambda$ where $L_o$ can be the empty set. For instance, revisiting our example, the observed labelset $L_o = \{Bougatsa\ Place\}$ while $L = \{Bougatsa\ Place, Breakfast\ Spot, Pastry\ Shop, Snack\ Place\}$.

[4] English translation:Traditional bougatsa place Badis.

We denote by $\boldsymbol{y} = (y^1, ..., y^m)$ an m-dimensional binary vector where $y^i \in [0, 1]$ such that $y^i = 1$ if and only if $l_i \in L$. Accordingly, the m-dimensional binary vector $\boldsymbol{y_o} = (y_o^1, ..., y_o^m)$ with $y^i \in [0, 1]$ has $y_o^i = 1$ if and only if $l_i \in L_o$. We assume that $S \cup T \cup R = A$ where $S$ stands for the set of spatial attributes, $T$ the set of the temporal ones, and $R$ the rest of the attributes. Considering again our example, the spatial attributes latitude and longtitude would be instantiated by the corresponding values. The temporal attribute would include the opening times, while $R$ the name of the POI.

We denote by $\boldsymbol{x}, \boldsymbol{x_R}, \boldsymbol{x_S}, \boldsymbol{x_T}$ the vectors that represent correspondingly $A$, $R$, $S$, and $T$, such that the observed $p$ in the dataset, is defined as

$$p = \{\boldsymbol{x}, \boldsymbol{y_o}\} = \{\boldsymbol{x_R}, \boldsymbol{x_S}, \boldsymbol{x_T}, \boldsymbol{y_o}\} \quad (1)$$

and we are trying to complete $p$ such that

$$p = \{\boldsymbol{x}, \boldsymbol{y}\} \quad (2)$$

We formulate our goal as a multi-label classification problem where we want to find a classifier $h : X \rightarrow Y$ where $X$ is the input space (all possible attribute vectors) and $Y$ the output space (all possible labelset vectors), such that $\boldsymbol{y} = h(\boldsymbol{x})$.

We assume that the attribute containing opening times is included in $\boldsymbol{x_T}$ and the one with geospatial coordinates, e.g. latitude/ longitude, in $\boldsymbol{x_S}$. Opening times and geospatial coordinates should have non-null values. Optionally, $\boldsymbol{x_T}$ may also represent attributes describing the times that different services are available (e.g. in the case of restaurants, kitchen opening times may differ from bar opening times and/or happy hours).

## 4 DATA COMPLETION METHOD

To find $h$, we follow a standard approach and transform our problem into finding a real-valued vector function $f : X \rightarrow S$ that allows to indicate the relevance of a label $l_i$ in relation to the input i.e. $f(\boldsymbol{x}) = (f(\boldsymbol{x}, \boldsymbol{l_1}, ), f(\boldsymbol{x}, \boldsymbol{l_2}), ..., f(\boldsymbol{x}, \boldsymbol{l_m}))$ where $f(\boldsymbol{x}, \boldsymbol{l_i})$ is the confidence of $l_i \in \Lambda$ being a correct label for $\boldsymbol{x}$ and $m$ is the number of labels. Actually this corresponds to an estimation of $p(y^i|\boldsymbol{x}) : y^i \in [0, 1]$. Note that ideally, observed outputs should be completely specified vectors, however in our context the training instances are only partially complete, so of the form $(\boldsymbol{x^i}, y_o^i)$. We follow the Binary Relevance method, thus learn $m$ binary models, each specialised into predicting whether one label is correct or not, independently from the other labels. For an unseen $\boldsymbol{x}$, the predicted labels are then the union of the predictions of all the binary models.

To learn the binary models we execute the following steps:

- Attribute selection: In initial configurations we have followed the procedure proposed by Biessmann et al. [2] where the attribute to be completed is selected as the target and the rest of the attributes are used as input features. However, based on further experiments, we have elaborated this step into selecting the attributes that are the most representative for our task (for instance ignoring aggregation meta-attributes, such as total no. of likes, and attributes with highly sparse values such as twitter ids). This step reduces the search space and actually the model not only converges faster but also leads to slightly better precision and recall.
- Vectorisation: This step includes transforming the attributes in a form that can be treated by the imputation models. In addition to the traditionally distinguished types of categorical and sequential data, we also explore specialised vectorisers for spatial and temporal data. We will give more details in section 4.1.

- Imputation: The probability of $l_i$ being a correct label given $\boldsymbol{x}$ is computed in this step. As explained in the previous paragraph, our problem is casted as a supervised machine learning problem. Details are provided in section 4.2.

## 4.1 Vectorisation

**Categorical variables**. We represent them with embedding vectors, as usually reported in the literature [5, 2].

**Sequential variables**. Biessmann et al. [2] report that character-based representations are more robust for a similar setting to ours (i.e. sparse data and multiple languages). In addition, Joulin et al. [1] and Biessmann et al. [2] mention that character n-grams can perform better than simple, unigram, character-based LSTMs. After experimentation we have adopted trigram character based LSTMs that we train along with the classifier.

**Temporal variables**. In the case of POIs, temporal variables describe opening times.[5] Opening times represent recurrent intervals of time. Exceptionally they may vary over different seasons and/or specific periods of the year, such as Christmas. We can consider though that overall they exhibit periodicity. In this study we compare three alternative ways of representing and vectorising opening times.

- As a categorical variable (bucket-based). Transforming the string into one-hot vector according to the intervals during which the POI is open. For instance, if a POI opens every Monday at 9am and closes at 9pm, then if we decide the granularity of the intervals to be 3-hour ones daily, Monday would be represented with the following vector $[0, 0, 0, 1, 1, 1, 1, 0]$. The granularity of the intervals can be chosen based on a histogram of the values. We have decided to use 1/2 hour intervals for each day and concatenate them to represent all days of the week (where the week is repeated over the year). The concatenated vector is used as input.
- As a periodic variable. Opening time is periodic over two dimensions: 7-day week and 24h day intervals. As Bishop states, such quantities can conveniently be represented using an angular (polar) coordinate and as points of a circle [3]. Consequently, to appropriately transform days, opening and closing times, we create two vectors that hold the corresponding Cartesian coordinates. For instance, if the vector that represents instances of day-time hours found in our training data is $\boldsymbol{h}$, then each $h^i \in \boldsymbol{h}$ is transformed to two dimensions: $k^i = \sin(2\pi \times \frac{h^i}{24})$ and $y^i = \cos(2\pi \times \frac{h^i}{24})$ where $k^i \in \boldsymbol{k}$ and $y^i \in \boldsymbol{y}$. Then we use $\boldsymbol{k}, \boldsymbol{y}$ as input vectors instead of $\boldsymbol{h}$. Days and minutes are transformed in a similar manner.
- As a sequential variable. As mentioned above, time is written as a formatted string i.e. a semi-structured sequence of characters. Thus, a possibility is to consider time as a sequential variable and use the last state vector of a unigram-character based LSTM to represent the string value of the opening times (cf. 5.1.2).

**Spatial variables**. Geographical coordinates are the most important spatial attributes that characterise a POI. For instance, latitude and longitude are two of the most frequently used geographical coordinates. The predominant way of modelling coordinates is to discretise the input space [16, 21]. This could take the form of a grid separated into a fixed number of cells. Usually in this case the form and granularity of the cells has to be selected appropriately. In our context, POI categories are usually country-specific (and sometimes city or region-specific). After initial exploration, we have decided to map our geo-coordinates to countries. We can then model the corresponding data as categorical variables. A downside of that representation is that the model is not able to learn geographical regions in a data-driven way (e.g. [14, 18]). We leave exploring dynamic ways of learning appropriate representations for future work.

[5] As a POI may offer several different services, different opening times for each service may be included in the data e.g. kitchen opening times vs bar opening times.

## 4.2 Imputation

Once we vectorise our attributes, as explained in the previous section, we use a concatenation layer to combine them. So if $a$ is a POI attribute such that $a \in A$ and $\phi_a(x_a) \in \Re^{D_a}$ is the attribute specific vectorisation function, where $D_a$ denotes the dimensionality associated with the attribute $a$, then the final input vector is a concatenation of all vectorised individual attributes:

$$\tilde{\boldsymbol{x}} = [\phi_1(a_1), \phi_2(a_2), ..., \phi_n(a_n)] \tag{3}$$

where $n$ denotes the number of attributes. We feed this to:

$$\boldsymbol{h} = relu[\boldsymbol{W^h}\tilde{\boldsymbol{x}} + \boldsymbol{b^h}] \tag{4}$$

After applying a dropout layer, we then calculate:

$$p(\boldsymbol{y}|\boldsymbol{h}, \boldsymbol{\theta}) = sigmoid[\boldsymbol{W}\boldsymbol{h} + \boldsymbol{b}] \tag{5}$$

where $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{W^h}, \boldsymbol{b^h})$ are learned parameters of the model. $sigmoid(s)$ denotes the element-wise logistic function $f(s_i) = \frac{1}{1+e^{s_i}}$. The parameters $\boldsymbol{\theta}$ are learned by minimising the binary cross-entropy loss function. The multi-label model outputs for each label a probability score. To get from that the corresponding set of labels, a constant can be applied as threshold (usually this is 0.5) [11].

Note that, in the general case of data completion, partially observed labels can be considered as part of the inputs. We do not follow this approach, we rather focus on an extreme case of label incompleteness where no such partial observations are available.

# 5 EXPERIMENTS

## 5.1 Set up

### *5.1.1 Data*

As mentioned in the previous sections, we run our experiments on ≈900K POIs extracted from a large database provided by Foursquare. The focus of the experiments is on (i). understanding the capability of the model to predict POI categories, and (ii). exploring the influence of different spatio-temporal attribute representations on the data completion task. Details are provided below.

**Categorisation hierarchy** Our dataset includes 779 POI categories from the categorisation hierarchy of Foursquare (Table 1[6]). As the classification hierarchy is based on crowdsourced data, the parts of the dataset that include more POI instances are represented with more categories, resulting in it being heavily imbalanced. For instance, the most well developed category is *Food*, with 336 categories distributed in 5 levels. The least developed one is *Residence* having only 4 subcategories, over 2 levels.

[6] https://developer.foursquare.com/docs/resources/categories as of 3rd October 2018

**Table 1.** Category Distribution in the dataset

| Root Category | Levels | Categories in Path |
|---|---|---|
| Food | 5 | 336 |
| Shop & Service | 3 | 150 |
| Professional & Other Places | 3 | 74 |
| Outdoors & Recreation | 4 | 78 |
| Arts & Entertainment | 3 | 49 |
| Travel & Transport | 3 | 41 |
| College & University | 3 | 26 |
| Nightlife Spot | 3 | 25 |
| Event | 2 | 7 |
| Residence | 2 | 4 |

**Semantic tag distribution**. We have extracted a dataset of POIs having spatio-temporal attributes from the existing Foursquare database. We used the most well developed root category, *Food*, as seed. The resulting dataset includes about 900K POIs. The distribution of the categories is similar to the one found in the original hierarchy, as shown in Table 1[7].

The label cardinality (i.e. average number of labels per POI) is 1.37, while the label density is 0.0018. The POIs that have strictly fewer than two tags are 63% of the overall set of POIs. These numbers further illustrate the complexity of the problem: the cardinality is relatively low due to incompleteness, while the density is also low, meaning there is a relatively high number of distinct labels. To better understand these numbers we provide the overall distribution of semantic tags in Table 2. As shown, the dataset is skewed in terms of the POI instances attributed to each category, with the first 10 top categories having more than half of the POIs attributed to them. The long queue of sparsely represented categories could also be an explanation of the low density. It also worths noting that:

| Category | % tags |
|---|---|
| Fast Food Restaurant | 10.35% |
| Café | 8.28% |
| Pizza Place | 7.70% |
| Coffee Shop | 7.66% |
| Sandwich Place | 5.77% |
| Restaurant | 5.08% |
| American Restaurant | 4.91% |
| ... | ... |
| Ice Cream Shop | 2.03% |
| Breakfast Spot | 1.93% |
| ... | ... |
| Diner | 1.65% |
| ... | ... |
| Food | 0.24% |

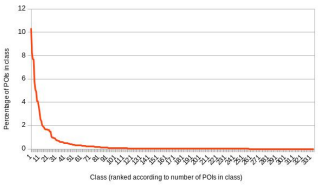


**Table 2.** Percentage & distribution of semantic tags

- Semantic tags attributed to the POI can be from different levels of the category hierarchy. For instance the root category *Food*, the second level category *Pizza Place*, and the third level category *Ice Cream Shop*, are all included as semantic tags in the dataset. This illustrates that there is no constraint on the tag(s) the user can input i.e. they can come from any level of the hierarchy.
- The hierarchy is counter-intuitive at places. For instance, *Restaurant* and *American Restaurant* are siblings rather than having a hierarchical relation between them. This could be explained from the crowdsourced nature of the resource. A direct consequence is that similar POIs tend to be labelled differently from users, not only because all relevant labels may not be found by the user, but also because differences between categories may be fuzzy.
- Spatio-temporal attributes should be more distinctive for some classes than others. For instance, *Breakfast Spot*s should open early and close earlier than *Bar*s, while *Diner*s should be found more frequently in the US rather than in other countries.

**Point-of-Interest attributes**. POI attributes include its name, the latitude and longitude, and its opening times, transformed into the different representations discussed in Section 4.1. It is important to note that contrary to completely freely crowd-sourced POI databases such as OpenStreetMaps [10], the format of these resources is normalised. Latitude and longtitude are written in the standard form, normally with >10-decimal point precision (e.g. latitude:55.76942424341726, longtitude: 44.948036880105064). Opening times are represented in the form "*day_1; opening_time_in_minutes_from_midnight_1; closing_time_in_minutes_from_midnight_1 | day_2...*". The days and the opening time intervals can have any order (e.g. day_2 i.e. Tuesday can be included before day_1 in the string representing the opening times). Comments are available for some of the POIs but as they are relatively sparse we chose not to use them in the experiments.

**Multi-linguality**. An important characteristic of Foursquare's data is that it covers the whole globe. As a consequence, it is highly multi-lingual. Previous work has focused only on a couple of different languages, with each language represented in a separate dataset and thus potentially processed with appropriately tuned models (e.g. specific hyper-parameter tuning). In the case of POIs, automatically creating such datasets is far from simple: POI string attributes, especially their names, are short, so automatic language identification techniques tend to be less accurate; POI names tend to occasionally contain a mix of two or more different languages e.g. Mr. Panino 北京小; and POI names are proper names in a number of cases e.g. KiKi. This aspect influenced our choice of using character-based models[8]. It is difficult to quantify the number of different languages and alphabets included in our dataset as automatic language identifiers are not reliable on short strings, such as POI names. As a proxy, we have analysed the, relatively sparse, comments related to POIs[9]. We have detected 46 languages in total, although 38.5% of the comments are written in English and the top 10 most frequently used languages cover 76% of the comments. We have found 9 different alphabets ranging from Russian to Chinese. The POIs found in the dataset come from 100 different countries.

**Silver standard**. To generate training and test data, we used approximate stratified sampling. The goal was to maintain the distribution of positive and negative examples of each label by considering each label independently. Consequently, we allocated the dataset of 890K POIs proportionally into 70% for training, 20% for development, and 10% for testing purposes. We would like to highlight here, that actually the dataset is a *silver standard*, as the sets of labels attributed to each POI are incomplete. This is an important limitation that has to be kept in mind when interpreting the experimentation results.

[7] Even if we used *Food* as seed, we find also the rest of the root categories in the dataset. The reason is that POIs can be categorised using multiple root labels, although at least one of the labels must have as seed *Food*.

[8] Intuitively, character-based models tend to be more robust in highly multi-lingual setting than token (or word) based ones.

[9] The analysis has been carried out with a proprietary language identifier that performs comparably, if not better, to Langid.py

#### 5.1.2 Evaluated Models

We have developed a number of model variations that correspond to the different representations discussed in Section 4.1. We compare those to the following state-of-the-art imputation methods.

- **BRknna** [15]. Binary Relevance multi-label classifier based on k-Nearest Neighbors method. k is set to 1 based on the results of a grid search (k: {1,6}).
- **Datawig_hash** [2]. State-of-the-art method for data imputation with categorical data, as reported by Biessmann et al. [2]. This model encodes POI names and opening times as hashed character n-grams, while the country is represented as a categorical variable. For fair comparison, we closely follow the model hyperparameters, regularization and optimization techniques introduced in [2], with only small changes to allow multi-label imputation (the original version performs only single-label imputation).
- **Datawig_LSTM** [2]. LSTM based state-of-the-art method, as introduced by Biessmann et al. [2]. This model encodes POI names and opening times with unigram-character LSTMs, and country is represented as a categorical variable.
- **Ours**. The model presented in this paper. The input variations include: (i) **Base**. POI names as trigram-character based LSTMs. When used alone, we consider it as a baseline for the rest of the models. (ii) **t_sincos**. Temporal information as a periodic variable. (iii) **t_30**. Temporal information as one hot vector based on intervals of 30 min (buckets). (iv) **t_LSTM**. Temporal information as sequence (unigram-character LSTMs). (v) **s_geo**. Geocoordinates as vectors representing countries.

For our model variations, we are using an architecture with one hidden dense layer, followed by a dropout layer, and the output layer. We use the Rectified Linear Unit as the activation function of the hidden layer. The dropout rate is set to 0.3. The loss we use is binary crossentropy. We have set an early stopping criterion for the training based on a pre-defined threshold that takes into account the delta of the loss between two consecutive epochs. For all sequential features we applied a length of 50. For the LSTM layer we set the dimensions of the embedding layer vector space to 128 and the number of the LSTM hidden units to 128. The LSTM has a recurrent droupout rate of 0.3. Experiments were run on a single GPU instance (1 GPU with 16GB VRAM, 4 CPUs, with 256GB RAM). Training was performed with a batch size of 32. We used the Adam optimiser with the default parameters recommended in [8].

## 5.2 Silver standard evaluation

The results reported below are on the test dataset, which has not been used for training or validation purposes. We have calculated for each model the average micro precision, micro recall and micro F1, excluding outlier values[10], over 10 runs[11]. Results are shown in Table 3[12].

#### 5.2.1 POI names as category predictors

Our baseline predicts POI categories with 72.55% precision and 50.38% F1 scores. The micro-F1 is also higher than the state-of-the-art methods, which indicates that (i) the name is a strong predictor of the category (ii) vectorising appropriately the name and optimising the model accordingly is extremely important.

**Table 3.** Average performance (%) over 10 runs (except for BRknna). Best results are in bold. Standard deviation is also reported.

| Model | Micro-prec. | Micro-rec. | Micro-F1 |
|---|---|---|---|
| BRknna [15] | 41.35 | 39.90 | 40.61 |
| Datawig_hash [2] | 74.89±0.63 | 35.59±0.68 | 48.24±0.50 |
| Datawig_LSTM [2] | 69.03±0.54 | 37.47±0.77 | 48.56±0.59 |
| Ours | | | |
| Base | 72.55±0.54 | 38.60±0.8 | 50.38±0.63 |
| +s_geo | 72.96±0.5 | 40.06±0.64 | 51.72±0.61 |
| +t_LSTM | 72.62±0.47 | 39.81±0.64 | 51.42±0.56 |
| +t_30 | 74.96±0.8 | 36.50±0.71 | 49.10±0.58 |
| +t_sincos | 74.20±0.51 | 38.38±0.54 | 50.60±0.47 |
| +s_geo+t_LSTM | 73.56±0.35 | **41.61±0.4** | **53.15±0.38** |
| +s_geo+t_30 | **74.96±0.72** | 37.38±0.98 | 49.88±0.80 |
| +s_geo+t_sincos | 74.61±0.34 | 39.50±0.56 | 51.65±0.48 |

#### 5.2.2 Improvements using spatio-temporal data

Appropriate addition of spatio-temporal information results in models with higher micro-precision and micro-F1 scores. The difference between the baseline and the models including spatio-temporal information is consistent. The average difference in terms of F1-score between the baseline and the best performing model is ≈2.7 absolute percentage points in micro-F1 and ≈2.4 points in micro-precision. Over the hundreds of millions of POIs included in the database of Foursquare these differences are significant.

In the following paragraphs we take a more detailed look at how the addition of each type of information impacts the results.

**Spatial data** In Table 4 we can see the categories for which performance differences, when compared to the baseline model, are most important. The results related to injecting spatial data seem to be self-explanatory: categories with limited training data and strongly correlated to the location of the POI are mainly included. Examples include, Australian Restaurant and Brasserie (POI type found mainly in France and the Francophone world). An interesting category is Pet Café: in that case the model has learned that Cafés and Tea Shops in Japan that have the character 猫 (meaning 'cat') in the name are probably categorised as Pet Cafés (the baseline favours *Café* instead). Although most of the time spatial data have a positive impact, their effect is generally limited in terms of micro scores, as the categories that are improved are mainly part of the long tail. In practical terms though this case is very important. Long tail categories are more rare and thus more difficult to find in an explicit search.

**Temporal data** In the case of the temporal model there are performance differences that intuitively make sense. For instance, the opening times of Convenience Stores, Grocery Stores, and Bars, should be particularly different compared to restaurants. However, other gains are more difficult to explain, such as in the case of the category *Portuguese Restaurant*. Analysing the corresponding predictions, we observe the following.

- The baseline occasionally has problems with seemingly easy predictions such as "Nando's Mall of the North", although Nando's is a well known Portuguese chain of restaurants with thousands of occurrences in the database (this happens especially when the

[10] Outliers are values that fall below Q1-1.5*IQR or above Q3-1.5*IQR where IQR is the interquartile range and Q1 and Q3 the first and third quartiles [17].
[11] Datawig_LSTM is averaged over 5 runs because of limited time.
[12] Please note that more details, such as a box plot visualisation of the results, can be found in our technical report at https://europe.naverlabs.com/publications/semantic-tag-completion.

**Table 4.** Top 10 categories in terms of performance difference to the baseline (in the brackets the delta in precision)

| s_geo | t_30 | s_geo+t_30 |
|---|---|---|
| Australian(0.85) | Portuguese(0.38) | Tex-Mex(0.41) |
| Pet Café(0.77) | Fish & Chips(0.31) | Diner(0.26) |
| Trattoria/Osteria(0.66) | Bubble Tea(0.26) | Taco Place(0.25) |
| Hong Kong(0.61) | Taco Place(0.24) | Italian(0.23) |
| Austrian(0.58) | Friterie(0.16) | Wings Joint(0.17) |
| Brasserie(0.55) | Convenience St.(0.11) | Thai(0.14) |
| Mongolian(0.54) | Noodle House(0.10) | Snack Place(0.13) |
| Cha Chaan Teng(0.45) | Chinese(0.09) | Sandwich Place(0.12) |
| Tex-Mex(0.43) | Grocery St.(0.08) | Bakery(0.11) |
| Unagi(0.39) | Bar(0.07) | Convenience St.(0.11) |

name of the POI includes a lot of characters). The regularity in the opening times though (i.e. 11:00-21:00 on weekdays and 09:00-23:00 on Friday and Saturday), seems to help the corresponding model to be more confident and generate a correct prediction.

- In other cases, the name indicates that it should be a Portuguese Restaurant but the opening times do not fit the corresponding distribution. For instance, Mando's, a Mexican Restaurant, is predicted as a Portuguese Restaurant by the baseline because of the obvious similarity of the name to Nando's. However, the POI is open daily 09:00-02:00. In that case, the t_30 model does not generate any predictions and thus no false positives.

The two points mentioned above are representative of the way in which temporal information also helps with other categories.

**Combining spatial and temporal data** Categories in this case have a strong geo-spatial character, constrained to a few countries, and in addition opening times differ among these countries. For instance, Tex-Mex Restaurant gains 0.21 points in micro-precision with the addition of spatial data because a large percentage of these POIs is found in the US and Mexico compared to other countries. In addition, the distribution of times is characteristic for each country and helps to further improve the predictions (Figure 1).

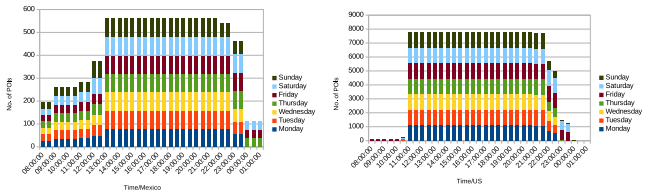


**Figure 1.** Tex-Mex opening time distribution in Mexico (left figure) and the US (right figure). Regularity in opening times and characteristic time ranges for each country on some days (especially Thursday) help to further improve the predictions.

### *5.2.3 Computation time*

The base (name-only) model took 27 hours and 33 epochs to converge. The base + s_geo + t_30 model took less time with 24.5 hours and only 19 epochs, while the base + s_geo + t_LSTM variant requires the most computation time, 73 hours and 39 epochs. This is a significant difference, especially in a production-oriented scenario where models have to be regularly updated.

## 5.3 Restrospective evaluation

While the results obtained on the silver standard are promising, the labels (i.e. semantic tags) attributed to each POI in the silver standard are incomplete. This could lead to the precision being incorrectly penalised, as values predicted by the model are counted as incorrect even if they are correct but absent from the test data due to incompleteness - in that sense the micro-precision values presented here are a worst-case scenario while the micro-recall ones may be optimistic.

To understand the potential impact of our method on the current database we created a small gold dataset by manually and carefully assigning complete sets of labels to 163 POIs. Label cardinality in this sample is 2.8 instead of 1.37 in the silver standard. There are several reasons for this difference: In some cases, it is obvious that semantic tags are missing from the silver standard e.g. McDonald's is tagged as a *Fast Food Restaurant* but not a *Burger Joint*. This is plainly illustrated in Figure 2, which shows the results of a search, using Foursquare's data, for POIs with the semantic tag *Noodle House* in a small part of Paris. The silver dataset includes only 20 results while 50 additional Noodle Houses are predicted by our system. Those results are counted as incorrect in our evaluation. However, we can see that the predictions clearly designate the 13th arrondissement, an area with a lot of Asian Restaurants, as having several noodle houses (the database includes only a couple). This sounds reasonable. Looking closer at the results, we see that the predictions are actually correct: La Table du Ramen is labelled as a *Chinese Restaurant* by users, while Ramen is a Japanese dish based on noodles. Pho Bida Vietnam is labelled as a *Vietnamese Restaurant*, however, Pho is a Vietnamese noodle soup, so the prediction seems to be more precise than the existing label.

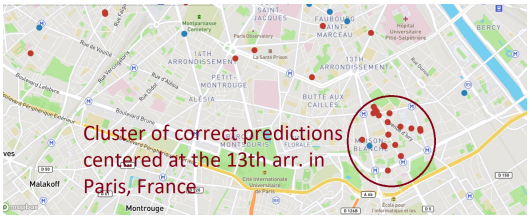


**Figure 2.** Example of predictions that are considered wrong in our silver standard evaluation due to label incompleteness. Red spots indicate POIs where the semantic tag *Noodle House* is completed by the classifier. Blue spots stand for Noodle Houses that already exist in the dataset.

However, there are also cases that are difficult to judge even for a human, as some categories are inherently fuzzy. For instance, Dunkin' Donuts has been labelled as a *Donut Place*, but is it also a *Snack Place* or a *Fast Food Restaurant*? In addition, semantics between seemingly distinct labels vary according to the country. For instance, in the majority of countries, Starbucks is categorised as a *Coffee Shop*, however, in Brazil half of the instances are also tagged *Café*. This illustrates the difficulty of creating a gold standard.

Faced with this difficulty, we have also performed retrospective evaluation i.e. the output of the system is given to human judges for annotation, who then label completions as correct and incorrect [12]. The metric in this case is usually precision combined with the total number of completion or errors found. We evaluated in this manner our best performing model, Base + s_geo + t_LSTM, on 100 POIs that were marked as errors in the silver standard evaluation. In the case of multiple predicted labels, if all labels were correct then we

considered the whole prediction correct. In the case that m out of n predicted labels were correct we calculated a score that corresponds to the m/n ratio. Results are shown in Table 5. Projected completions account for more than 7% of the number of labels, i.e. 11900 additional labels, at a projected micro-precision reaching ≈86%.

We have to note here that we follow a strict evaluation approach i.e. the label has to be exactly the same as in the ground truth to be considered correct. However, in some cases the model predicts categories that are one level higher in the POI category hierarchy of Foursquare. For instance when the prediction is *Bar* while in the ground truth the label is *Cocktail Bar* or *Sports Bar*. In these cases the error is not potentially as important as predicting *Bakery* while the correct category is *Japanese Restaurant*. We analysed such cases in the retrospective evaluation. Out of the 100 silver standard errors, 7 were due to generalisation and 8 due to label specialisation (e.g. *Empanada Restaurant* was predicted instead of *Mexican Restaurant*), while 3 of them were errors in the silver standard. In future work, the notion of error importance should be introduced in our metrics to account for such differences.

## 6 BONUS APPLICATION: CATEGORY TREE REFINEMENT

While the main objective of this work is database completion, we found that it can also be used to provide visual (and statistical) cues to human curators to help them refine Foursquare's category hierarchy.

The second (and last) layer of activations in our Neural Network architecture can be understood as an embedding of the POI categories. We have used a combination of PCA and t-SNE for dimensionality reduction and matplotlib to visualize the corresponding result. In Figure 3 we can see the visualisation, after filtering out rare semantic tags (filtered out tags were attributed to less than 5 POIs in our dataset). Different colours represent different categories.

An illustrative example is given in the Figure, where we zoom into a specific area. In the zoom, in addition to colours, we project a number representing each category. The median of the coordinates of all POIs of a category is used to position the number in the 2D space. We can see that all the Japanese restaurants are clustered together, which conforms to the corresponding part of the manually created Foursquare hierarchy (see *Japanese Restaurant* subcategories). However, in addition, we find that *Ramen Restaurant* is very close to the *Noodle House* category, which is in turn in the middle of Asian restaurant related categories. In the Foursquare category hierarchy, no specific relation exists between *Ramen Restaurant* and *Noodle House*, or *Noodle House* and other Asian types of restaurants (i.e. the only common ancestor is the root category *Food*).

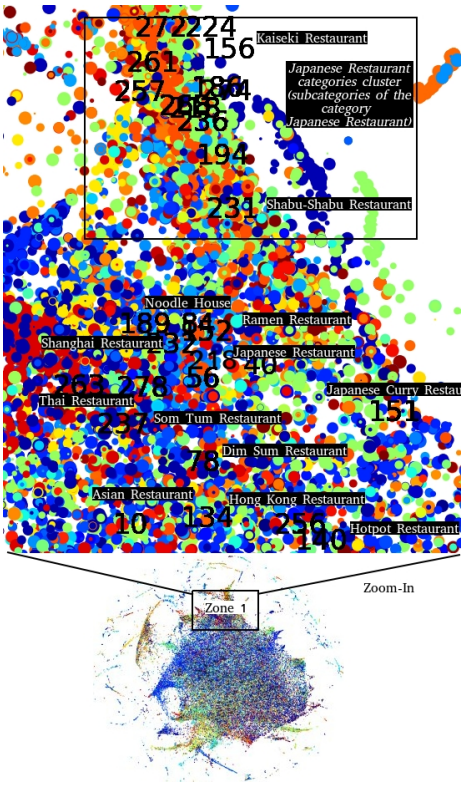


**Figure 3.** Visualisation of categories after PCA and t-SNE dimensionality reduction and zoomed-in visualisation of Japanese, Asian, and other related categories (zone 1). Different colours stand for different categories.

## 7 CONCLUSIONS

We have presented an approach to multi-lingual completion of POI semantic tags in a dataset from Foursquare, a global crowdsourced Location-Based Social Network. The data in the domain present a number of challenges: multiple tags can be correct for each POI; spatio-temporal data that characterise the POIs require specific preprocessing; and there are hundreds of categories that can be used to tag the POIs. In addition, the distribution of semantic tags is severely skewed and data related to POIs includes noisy information.

In this context, semantic tag completion is a multi-label classification problem. To tackle the problem we propose a neural-based approach where metadata from POI attributes, notably their names, geo-coordinates, and opening times, are used as inputs to the classifier. After extensive experiments we have observed that (i) POI names are strong predictors of POI semantic tags: a character-based LSTM model trained only on POI names gives 72.5% worst-case micro-precision and 50.38% micro-F1 scores, (ii) appropriate use of spatio-temporal data leads to consistent improvements (iii) using a structured representation of time gives higher precision and requires less computation time than a string-based LSTM variant, however, the higher precision is achieved at a cost of lower recall and micro-F1, (iv) an LSTM model trained on semi-structured strings representing time, is competitive to fully structured inputs in terms of recall.

The analysis of the results indicates that label incompleteness is a particular difficulty for the evaluation of the task itself: when we use existing (incomplete) data as ground truth, values predicted by the model are counted as incorrect even if they are correct but absent from the test data due to incompleteness. Therefore the reported precision values correspond to a worst-case scenario. In addition, especially in production, the notion of error importance should be introduced in our metrics. This would account for differences between errors due to predicting labels that are more general (e.g. Bar instead of Cocktail Bar), different but compatible (e.g. Noodle House instead of Vietnamese Restaurant), or incompatible ones (e.g. Bakery instead of Japanese Restaurant).

We plan to deploy the proposed solution as a tool for database curators and maintainers. This live test will provide us with additional human evaluation, and give us a better appreciation of the strengths and weaknesses of our approach.

**Table 5.** Results of retrospective evaluation.

| Model | No. of predictions marked as errors in the silver standard evaluation that are actually correct | Total no. of predictions (average) | Projected micro-prec. (silver) | Projected no. of correctly imputed semantic tags in the current test set (Original no.of labels) |
|---|---|---|---|---|
| Base + s_geo + t_LSTM | 47/100 | 85414 | 0.8592 (0.7356) | ≈11900 (151K) |

## References


[1] J. Armand, G. Edouard, B. Piotr, N. Maximilian, and M. Tomas, 'Fast Linear Model for Knowledge Graph Embeddings', *arXiv e-prints*, arXiv:1710.10881, (Oct 2017).

[2] F. Biessmann, D. Salinas, S. Schelter, P. Schmidt, and D. Lange, 'Deep learning for missing value imputation in tables with non-numerical data', in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pp. 2017–2025, New York, NY, USA, (2018). ACM.

[3] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 105–110, Springer-Verlag, Berlin, Heidelberg, 2006.

[4] A. E. Cano, A. Varga, and F. Ciravegna, 'Volatile classification of point of interests based on social activity streams', in *In Proceedings of the 10th International Semantic Web Conference, Workshop on Social Data on the Web (SDoW)*, (2011).

[5] Cheng Guo and Felix Berkhahn. Entity embeddings of categorical variables, 2016.

[6] T. He, H. Yin, Z. Chen, X. Zhou, S. Sadiq, and B. Luo, 'A spatial-temporal topic model for the semantic annotation of pois in lbsns', *ACM Trans. Intell. Syst. Technol.*, **8**(1), 12:1–12:24, (July 2016).

[7] S. Jiang, A. Alves, F. Rodrigues, J. Ferreira, and F. C. Pereira, 'Mining point-of-interest data from social networks for urban land use classification and disaggregation', *Computers, Environment and Urban Systems*, **53**, 36 – 46, (2015). Special Issue on Volunteered Geographic Information.

[8] D. Kingma and J. Ba, 'Adam: a method for stochastic optimization (2014)', *arXiv preprint arXiv:1412.6980*, **15**, (2015).

[9] J. Krumm and D. Rouhana, 'Placer: Semantic place labels from diary data', in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, pp. 163–172, New York, NY, USA, (2013). ACM.

[10] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org, 2017.

[11] G. Ouadie, *Ensemble multi-label learning in supervised and semi-supervised settings*, Ph.D. dissertation, Université de Lyon, 2017.

[12] H. Paulheim, 'Knowledge graph refinement: A survey of approaches and evaluation methods', *Semantic Web*, **8**, 489–508, (12 2016).

[13] N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich, 'Data lifecycle challenges in production machine learning: A survey', *SIGMOD Rec.*, **47**(2), 17–28, (December 2018).

[14] A. Rahimi, T. Baldwin, and T. Cohn, 'Continuous representation of location for geolocation and lexical dialectology using mixture density networks', in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 167–176. Association for Computational Linguistics, (2017).

[15] E. Spyromitros, G. Tsoumakas, and I. Vlahavas, 'An empirical study of lazy multilabel classification algorithms', in *Proc. 5th Hellenic Conference on Artificial Intelligence (SETN 2008)*, (2008).

[16] P. Tsangaratos, D. Rozos, and A. Benardos, 'Use of artificial neural network for spatial rainfall analysis', *Journal of Earth System Science*, **123**(3), 457–465, (Apr 2014).

[17] G. J. G. Upton and I. T. Cook, *Understanding Statistics*, 55–56, Oxford University Press, 1996.

[18] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng, 'When will you arrive? estimating travel time based on deep neural networks', in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, eds., Sheila A. McIlraith and Kilian Q. Weinberger, pp. 2500–2507. AAAI Press, (2018).

[19] Y. Wang, Z. Qin, J. Pang, Y. Zhang, and J. Xin, 'Semantic annotation for places in lbsn through graph embedding', in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pp. 2343–2346, New York, NY, USA, (2017). ACM.

[20] M. Ye, D. Shou, W-C. Lee, P. Yin, and K. Janowicz, 'On the semantic annotation of places in location-based social networks', in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pp. 520–528, New York, NY, USA, (2011). ACM.

[21] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, 'Dnn - based prediction model for spatio-temporal data', in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPACIAL '16, pp. 92:1–92:4, New York, NY, USA, (2016). ACM.

[22] X. Zheng, J. Han, and A. Sun, 'A survey of location prediction on twitter', *IEEE Transactions on Knowledge and Data Engineering*, **30**(9), 1652–1671, (Sep. 2018).

[23] J. Zhou, S. Gou, R. Hu, D. Zhang, J. Xu, A. Jiang, Y. Li, and H. Xiong, 'A collaborative learning framework to tag refinement for points of interest', in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 1752–1761, New York, NY, USA, (2019). ACM.