Master's Thesis

Forecasting Population at Gathering Places with Microblog Posts Referring to Future Events

（未来のイベントに言及するマイクロブログ投稿を用いた人が集まる場所の人口予測）

Department of Information and Communication Engineering
Graduate School of Information Science and Technology
The University of Tokyo

48-196430　Ryotaro Tsukada

Supervisor
Professor Masashi Toyoda

January 28, 2021

本論文は東京大学大学院情報理工学系研究科に修士号授与の要件として
提出した修士論文である．

Abstract

Large events with many attendees cause surge of population in the traffic network around the gathering places. To avoid accidents or delays due to this kind of unexpected behavior of public gatherings, it is important to predict the level of population change in advance of the event. Thus, this study aims to forecast population at general gathering places.

However, our preliminary experiment showed that historical population information alone is insufficient to forecast population at general gathering places when non-recurrent events are held there. Although a simple solution to this problem is to utilize the event schedules published by venue managers or event organizers, this approach cannot be scaled to many events or places for the following two reasons: (1) the number of places for which event schedules are published is limited and (2) some public gatherings (e.g., *hanami*, mass demonstration, etc.) cannot be found in such schedules.

To address this problem, we utilize microblog posts referring to future events as an indicator of event attendance. On the basis of this idea, we propose regression models that is trained with microblog posts and historical population information to accurately forecast population at gathering places.

Experiments on next 24-hour population forecasting using real-world traffic and Twitter data demonstrated that these two heterogeneous data sources have a complementary role in reducing the prediction errors over those of the baseline models (autoregressive and long short term memory) by 20% – 50%. Furthermore, our analysis on feature importances and attention weights suggested that our approach of using microblog posts had an advantage of explainability of forecast results, which are practically useful when users take measures against population surges.

Contents

List of Figures

Tables

Chapter 1

Introduction

1.1 Background

Large events such as baseball games or concerts attract huge crowds of people. They cause surges of people around their venues. Such unexpected behavior of people has various negative impacts not only on the event attendees themselves, but also on passers-by. For example, if a train is more crowded than usual when it is packed with people returning from a concert, passengers may feel physically and mentally stressed, and the increased time spent aboard the train results in economic losses. Similar problems happen around general gathering places: If tourists are not informed in advance about population surges at sightseeing spots, they may be dissatisfied with the trip if their plans are disrupted by the congestion. In addition, such unusual gatherings of people can even cause fatal accidents. On 2014 New Year's Eve celebration event in Shanghai, the crowd became uncontrollable and 35 people died in a stampede [1].

Population forecasting plays a key role in solving these problems. For instance, if the time, place, and degree of future population can be predicted, train passengers can take a different line where trains are less crowded, and tourists can make plans so that they can avoid congestion. Against this background, previous studies [2,3] have proposed frameworks that utilize data collected from location-based services and predict population on the city scale. However, the predictions of these methods are limited to a few hours in advance because the longer the forecast horizon is, the greater the effect of external factors (e.g., large events) becomes [4]. Another study has utilized railway route-search logs to make a prediction seven days in advance of events [5]. This method, however, is only applicable to stations. To give people a time to take measures against population surges at general gathering places, the forecasting should be over a long enough period and applicable to various places.

1.2 Challenges

In this study, we tackle the problem of longer-term population forecasting at general gathering places. However, as we show later (Chapter 4), it is difficult to forecast population in the vicinity of gathering places where non-recurrent events are held. This is mainly because a prediction model cannot capture the surge of people caused by non-recurrent events without any prior information about them. To address this, we focus on the fact that some microblog posts refer to future events. Such posts are valuable for automatically extracting the time, place, and type of the target events. For example, if there is a post saying "Protest in front of the National Diet Building on April 14!" indicates that a mass demonstration will take place in front of the National Diet Building on April 14. On the basis of this idea, we propose to predict population in the future from both historical population information and microblog posts referring to future events (Chapter 5). We demonstrated the superiority of our models relative to baseline models in experiments on real-world traffic and Twitter data (Chapter 6).

1.3 Contributions

Our contributions are as follows.

- We established the unprecedented and general problem setting of population forecasting at gathering places. We empirically showed that the traditional approach of population forecasting (i.e., the approach based on the historical population information alone) does not work in that problem setting.

- We proposed a method of population forecasting at gathering places utilizing microblog posts referring to future events. Since this method does not require event schedules, it can cover many types of gathering places.

- With our prediction method, we reduced the prediction errors of the baseline methods that only utilized historical time series of population by 20% – 50%. We showed through post-hoc analyses that our method captured important clues in microblog posts, suggesting that our approach has potential for explaining the prediction results, which leads to practical applications.

1.4 Thesis Structure

The structure of this thesis is outlined as follows.

Chapter 2 The problem we study in this thesis is defined and basic knowledge about our NN-based model is introduced.

Chapter 3 Related studies of this thesis are presented. By discussing their contributions and limitations, the relation between them and this thesis is clarified.

Chapter 4 Our preliminary experiment that showed historical population information alone is insufficient to forecast population at gathering places is detailed.

Chapter 5 A method for forecasting population at gathering places with microblog posts referring to future events is proposed. The two models of its implementation are also described.

Chapter 6 The experiments to demonstrate the superiority of our method in various task settings and their results are presented and discussed.

Chapter 7 The overall contribution of this thesis is summarized.

Chapter 8 The possible future directions of this thesis are discussed.

Chapter 2

Preliminary Knowledge

In this section, we firstly formulate the population forecasting problem and then introduce the Transformer [6] and the Set Transformer [7], which are the main components of one of our proposed models.

2.1 Problem Definition

In this section, we formulate the problem of population forecasting at gathering places.

Definition 1 (Sequence of population at a gathering place on the target day). Sequence of population $\boldsymbol{y}_{v,d}$ at a gathering place v on the target day d is a subset of corresponding spatio-temporal population data. It is a time series for the day d:

$$\boldsymbol{y}_{v,d} = \{y_{v,1}, y_{v,2}, \cdots, y_{v,n}\}.$$

Definition 2 (Available resources for the forecasting). The resources used to forecast the population on the day d must be the information available as of $d-1$. We refer to such information as X_{d-1}. For example, a model that utilizes the historical population information alone is the case of $X_{d-1} = \{\boldsymbol{y}_{v,d-l}, \cdots, \boldsymbol{y}_{v,d-2}, \boldsymbol{y}_{v,d-1}\}$.

Problem. Given X_{d-1}, predict $\boldsymbol{y}_{v,d}$.

2.2 Transformer

In this section, we explain the concept of the Transformer, which constitutes the basis of one of our proposed models. The Transformer is applied to the state-of-the-art neural machine translation systems. Typically, it takes an input sentence and outputs the translated one through its encoder-decoder architecture. Our proposed model, however, only utilizes the encoder part to obtain context-aware token embeddings. Thus, in what follows, we focus on the two key ideas regarding the encoder: the self attention and the positional encoding.

2.2.1 Self Attention

The motivation for introducing the self attention is to highlight a small but important part of an input text. As described later in Chapter 5, our proposed model takes event-related microblog posts and outputs the number of people at the corresponding place. For example, an input post looks like the following:

> 8/20（日）14時 巨人対横浜DeNA 東京ドーム スカイシート5 タオル引換券付は売切れ、余ったチケットの買取受付中。

In this post, only the words “巨人” and “14時” are necessary for prediction, and the others are less important. Here, the self attention tells which words are important to the successive components.

Specifically, an input sentence is processed as a sequence of word embeddings X and then it is linearly transformed as:

$$Q = W^Q X,\ K = W^K X,\ V = W^V X\,.$$

Here, W^Q, W^K, W^V are learnable parameters. On the basis of this transformation, the context-aware representation of X is obtained as the weighted sum of V:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d}})V\,.$$

2.2.2 Positional Encoding

The formulation above ignores the order of words. Since the word order plays a key role in solving NLP tasks, the Transformer encodes it by adding the positional encoding to the input. The i-th dimension of a positional encoding is defined by the following:

$$\begin{aligned} PE_{(\mathrm{pos},2i)} &= \sin(\mathrm{pos}/10000^{2i/d}) \, , \\ PE_{(\mathrm{pos},2i+1)} &= \cos(\mathrm{pos}/10000^{2i/d}) \, . \end{aligned}$$

Here, pos is the position of a token and d is the dimension of tokens. This is a sine wave and it injects information about positions of tokens to the model.

2.3 Set Transformer

In this section, we introduce the Set Transformer, which is an extended version of the Transformer. The Set Transformer is proposed to solve *set-input problems*. As mentioned later in Chapter 5, the problem of population forecasting at gathering places using microblog posts is an instance of set-input problems. In what follows, we show the definition of set-input problems and the architecture of the Set Transformer.

2.3.1 Set-Input Problem

Many existing problems addressed by deep learning are *instance-based* problems, that is, a fixed-dimensional input tensor is mapped to its target value. For some applications, however, an input takes the form of *set-structured data*. For example, representation learning of a recipe from a set of its ingredients [8] is an instance of set-input problems, where a set of input tensors is mapped to its target label for the set.

There are two requirements that a model for set-input problems should satisfy:

1. It should be *permutation invariant*.

2. It should process sets of any number of elements in them.

Taking the problem of retrieving the *max* value from a set of numbers as the simplest form of set-input problems, we can exemplify these requirements as follows:

1. $\max(\{1,2,3\}) = \max(\{3,2,1\}) = 3$
2. $\max(\{1\}),\ \max(\{1,2,3\}),\ \max(\{1,\cdots,100\})$

2.3.2 Architecture of Set Transformer

Zaheer et al [9] proposed the DeepSet, which is a neural network architecture that satisfies these requirements. It embeds each element in an input set into an instance space by a feed-forward neural network, and then it aggregates the embeddings by a pooling operation, such as mean, sum, max, etc. The Set Transformer further improved the DeepSet by introducing the attention mechanism to capture the highly-order interaction between elements in a input set.

The overall architecture of the Set Transformer is an encoder-decoder architecture, as illustrated in 1. In what follows, we briefly explain each component of the Set Transformer (SAB, ISAB, and PMA).

2.3.3 Set Attention Block (SAB)

Given matrices $X, Y \in \mathbb{R}^{n \times d}$ of two sets of d-dimensional vectors, the Multihead Attention Block (MAB) is defined as follows:

$$\mathrm{MAB}(X, Y) = \mathrm{LayerNorm}(H + \mathrm{rFF}(H)),$$

where

$$H = \mathrm{LayerNorm}(X + \mathrm{Multihead}(X, Y, Y)).$$

Here, rFF is a row-wise feed-forward layer and LayerNorm is layer normalization [10]. The MAB is the same as the encoder part of the Transformer, which is introduced in the previous section, without the positional encoding. It turns a

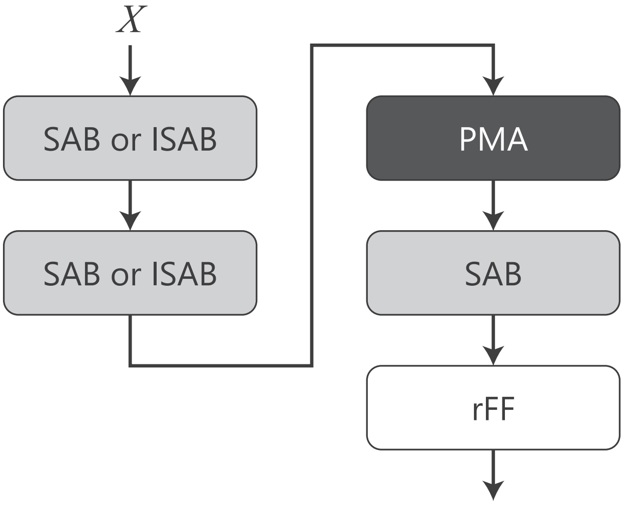


Figure 1 Overview of the Set Transformer.

disadvantage of the Transformer's order-insensitive property into an advantage for realizing a permutation-invariant module. Based on this formulation, we define the Set Attention Block (SAB) as follows:

$$\mathrm{SAB}(X) = \mathrm{MAB}(X, X)\,.$$

An SAB applies self-attention between the items in an input set and outputs a set of equal size. It naturally follows that the SABs are insensitive to the order of elements in an input set, since they are equivalent to the Transformer's encoder without the positional encoding.

2.3.4 Induced Set Attention Block (ISAB)

The time complexity of processing SABs is $\mathcal{O}(n^2)$, resulting in poor performance for large sets ($n \gg 1$). To mitigate this problem, the Induced Set Attention Block (ISAB) is introduced. An ISAB with m inducing points I is defined as follows:

$$\mathrm{ISAB}_m(X) = \mathrm{MAB}(X, H) \in \mathbb{R}^{n \times d}\,,$$

where

$$H = \mathrm{MAB}(I, X) \in \mathbb{R}^{m \times d}\,.$$

Here, inducing points I are trainable parameters. By introducing I, the time complexity of ISABs is reduced to $\mathcal{O}(mn)$, where m is a small hyperparameter.

2.3.5 Pooling by Multihead Attention (PMA)

The Pooling by Multihead Attention (PMA) is proposed as a pooling operation of feature vectors. Let $Z \in \mathbb{R}^{n \times d}$ be the set of feature vectors fed by an encoder, PMA with k seed vectors is defined as follows:

$$\mathrm{PMA}_k(Z) = \mathrm{MAB}(S, \mathrm{rFF}(Z))\,,$$

where $S \in \mathbb{R}^{k \times d}$ is a trainable set of k seed vectors.

Chapter 3

Related Work

3.1 Congestion Prediction Using Location Data

Location data collected from GPS-equipped mobile phones or cars are widely used as means of congestion prediction at gathering places. A typical approach to forecasting congestion is to exploit the spatio-temporal dependencies in such location data. For example, a traffic congestion occurring at 8 AM in Region X will affect the traffic state at 9 AM in Region Y when the congestion propagates through the road networks between X and Y. To automatically capture such dependencies in city-scale congestion data, Zhang et al. [11] proposed CNN-based architecture to predict crowd flows in an analogous manner to the video next-frame generation task. They released the code and data for the improved version of their method [4], enabling many researchers to compare their results of congestion prediction. Fan et al. [2] pointed out that the previous studies on congestion prediction treated congestion triggered by large events as outliers. To tackle this problem, they proposed an online version of the Markov chain model trained with GPS-based short-term human movement data to forecast city-scale human movements during large events.

A more direct approach to forecasting future human movements is to use query logs from transit or map apps. These query logs reflect individuals' intentions regarding their future movements and some works collectively utilized them. Konishi et al. [5] utilized the route-search query logs from a transit app as users' future schedules. Since route-search queries contain the future target time specified by users, they were able to forecast population surges at railway stations triggered by large events such as fireworks up to seven days in advance of them. However, due to their dependence on railway route-search queries, this method is only applicable to population forecasting at stations. Liao et al. [3] proposed a deep learning method based on the fact that the number of the search queries for a gathering place on map apps increases just before the event. This method does not depend on railway route-search queries and can cover not only stations but also general gathering places. However, their forecast horizon was limited to two hours since map search queries do not contain the target time information, in contrast to railway route-search queries. Another direction of forecasting congestion triggered by such events is to explicitly utilize the event schedules published by venue managers or event organizers. Rodrigues et al. [12] collected the event schedule of a venue from its official website. They utilized the textual information in it as an additional feature and showed that utilizing event schedules significantly reduced the prediction errors of the number of taxi pickups near the venue. However, this approach is limited for venues with their event schedules available and events that do not appear on any official schedules (e.g., mass demonstrations) are totally ignored.

The existing methods directly associate these location data with spatio-temporal points in a city. In contrast, our models can combine these spatio-temporal data with the microblog posts that are only weakly associated with gathering places. These posts are easily available for many places from earlier days prior to events.

3.2 Event Extraction from Microblog Posts

Microblog services such as Twitter have been widely used as social sensors for capturing information on real-world events.

Many researchers are interested in the real-time property of microblog posts for detecting ongoing events. Sakaki et al. [13] trained a classifier that classified a given post is related to an earthquake. They incorporated it into a system that issued an earthquake alert when the number of earthquake-related posts burst. Zhang et al. [14] tackled the problem of extracting local events from massive geotagged posts. They focused on a representative post in a spatio-temporal burst of event-related geotagged posts and demonstrated the effectiveness of their method on real-life datasets.

Another line of research is future or past event extraction by resolving temporal expressions in posts. Ritter er al. [15] extracted structured open-domain event data from inclusive but noisy information stream, i.e., Twitter. They adopted a sequence-labeling approach consisting of a named entity tagger, a part-of-speech tagger, and a time-expression resolver trained with noisy Twitter corpora, and released an auto-generated high-precision event calendar service. Yamada et al. [16] extracted the local event information from Twitter posts to help tourists make trip plans. They normalized venue names to collect posts containing event information and extracted the event name and duration from those posts. Jatowt et al. [17] proposed a visual analytics framework for future and past events based on time-referring expressions in microblog posts.

Inspired by these studies, we utilize microblog posts that contain both normalized place names and time-referring expressions as an indicator of future attendance.

3.3 Congestion Estimation and Prediction Using Microblog Posts

Some studies attempt to utilize microblog posts to estimate or predict congestion in the real world.

Onishi and Nakashima [18] analyzed the mutual interaction between population in the real world and the number of microblog posts in the virtual world. They tried to explain the mutual interaction between the population and the number of microblog posts through the parameters of the model trained with real-world data. However, their model is limited to estimating the current population by using the number of microblog posts and cannot be applied to future population prediction, which we study in this thesis. For human mobility estimation during unprecedented events, Miyazawa et al. [19] applied topic modeling methods on geotagged Twitter data and integrated the topic features with GPS data to train a regression model. Their experiments showed that the textual contents of geotagged posts contributed to the improvement of human mobility estimation.

He et al. [20] pointed out that there are posts that mention traffic information and proposed a method to predict the future traffic volume for longer periods. Their method utilizes posts that are created within the target area of the prediction. Terroso-Sáenz et al. [21] proposed a human mobility prediction model that considered both user- and crowd-based mobility. They built a multi-component framework to collect information about users' current activity from geotagged Twitter posts. On the assumption that subway operators did not know the schedule of all events around stations, Ni et al. [22] tackled the problem of forecasting passenger flow under non-recurrent event conditions. Their proposal consists of a hashtag-based event detection algorithm and multi-modal forecasting approach leveraging both historical transit data and geotagged posts. These methods, however, are not applicable to the prediction task we study in this thesis for the following two reasons: (1) Twitter removed the function of geotagging in 2019[*1] and (2) users' current location does not always suggest information about their future location, especially when the forecast horizon is set to 24 hours or more.

Dao et al. [23] addressed the problem of longer-term forecasting of congestion by exploiting social networking data. They classified Twitter posts that contain certain keywords related to road conditions (e.g., congestion, rainfall, accidents, etc.) and adopted a data fusion technique of weighting corresponding external factors based on the classification labels. Alkouz and Al Aghbari [24] combined Twitter and Instagram streams to detect and predict traffic jams. They extracted locations from the text and geotag of posts and showed that the fusion of the two data streams improved both the detection and prediction performance.

In order to accurately predict traffic flows while handling non-recurrent traffic events such as accidents and road closures, Essien et al. [25] focused on posts

[*1] https://twitter.com/TwitterSupport/status/1141039841993355264

from specific Twitter accounts that provides information about road conditions and incorporated them to their LSTM-based prediction model. Although they showed that information from specific accounts was useful, many potential accounts remain unused. In contrast, we do not limit the accounts from which posts are collected, resulting in much more coverage of potential accounts as an indicator of future traffic conditions.

These studies showed that using textual contents from microblog streams is promising for traffic forecasting. We thus further extend these approaches and propose a general method that does not depend on the geotagging function and specific accounts.

Chapter 4

Preliminary Experiment

As we describe in Chapter 3, using historical population information is a typical approach to population prediction. In this section, we show that forecasts based on this information alone are far from satisfactory, especially at gathering places.

4.1 Dataset

We used "Konzatsu-Tokei®" Data.[*2] It consists of the estimated numbers of people in square (250 m × 250 m, standardized in JIS X0410 [26]) grids of Japan that were aggregated every hour from Sep. 2015 until Nov. 2018. We used the last three months of data for testing and the remaining for training. We focused on the 1,500 most populated grids in Tokyo as of Aug. 2018. Those grids contain large gathering places such as baseball stadiums and concert halls, and transportation hubs such as main terminals and arterial roads.

[*2] "Konzatsu-Tokei®" Data refers to people flows data collected by individual location data sent from mobile phone under users' consent, through Applications* provided by NTT DOCOMO, INC. Those data is processed collectively and statistically in order to conceal the private information. Original location data is GPS data (latitude, longitude) sent in about every a minimum period of 5 minutes and does not include the information to specify individual. * Some applications such as "docomo map navi" service (map navi / local guide).

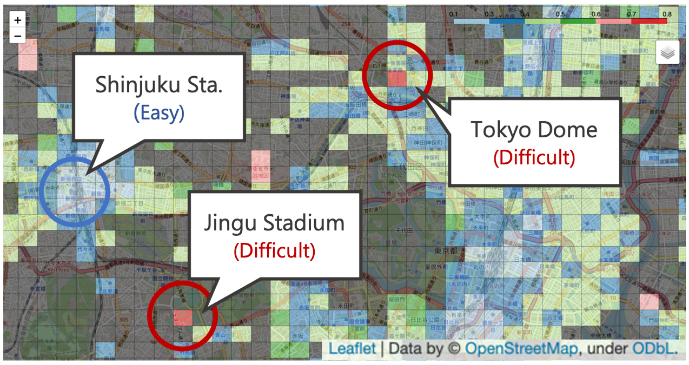


Figure 2 Prediction errors for the 1,500 most populated grids in Tokyo. Grids with higher WAPEs are shown in red and those with lower WAPEs are in blue.

"Konzatsu-Tokei®" ©ZENRIN DataCom CO., LTD.

4.2 Prediction Method

We trained an autoregressive (AR) model for each grid. It makes a prediction $\hat{X}_t$ for a time step t by linear regression using actual values over the last week $\{X_{t-24\times7}, \cdots, X_{t-1}\}$. In the rest of this thesis, the time interval is 60 minutes.

4.3 Evaluation Metric

To make the results of different grids comparable, we use the weighted absolute percentage error (WAPE):

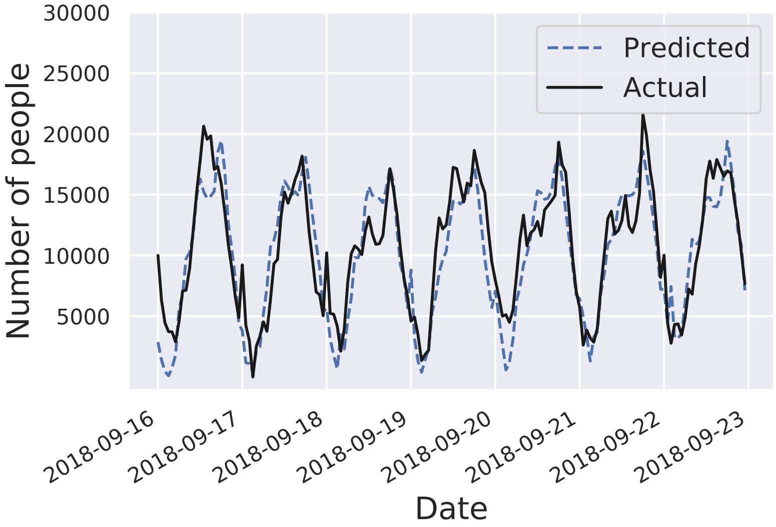
$$\mathrm{WAPE} = \frac{1}{N}\sum_{t}^{N}|\frac{\hat{X}_t - X_t}{\tilde{X}}|, \tag{1}$$

where X_t and $\hat{X}_t$ are respectively the actual and predicted values at a time step t, N is the total number of time steps evaluated, and $\tilde{X}$ is the mean of X_t.

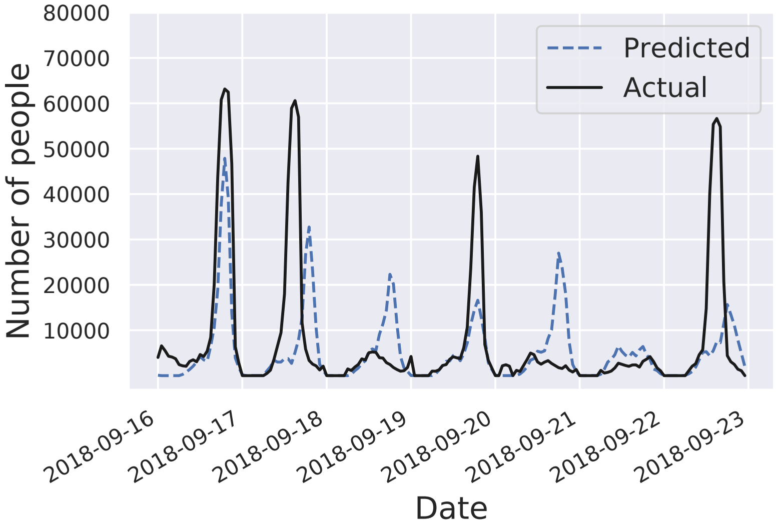
4.4 Prediction Results and Discussion

The prediction results are shown in Figure 2, where grids with higher WAPEs (i.e., difficult to predict) are shown in red and those with lower WAPEs (i.e., easy to predict) are in blue. The grids with lower WAPEs are distributed around main terminals such as Tokyo Station and Shinjuku Station. On the other hand, the grids with higher WAPEs are distributed around large gathering places such as Tokyo Dome and Jingu Baseball Stadium.

To further investigate the prediction error variance, we focused on the two grids that contain Shinjuku Station and Tokyo Dome. Figure 3 shows the actual and predicted time series of each of the two grids for the week from Sep. 16, 2018. A recurrent pattern during commuting hours is observed at Shinjuku Station. The AR model captured this pattern, resulting in high performance. However, at Tokyo Dome, population surge irregularly occurs due to the events held there. The AR model failed to adapt to these surges, resulting in poor performance. The causes of the surges were baseball games held at Tokyo Dome on Sep. 16, 17, and 19.



(a) Shinjuku Station



(b) Tokyo Dome

Figure 3 Predicted and actual number of people at Shinjuku Station and Tokyo Dome for a week from Sep. 16, 2018.

“Konzatsu-Tokei®” ©ZENRIN DataCom CO., LTD.

Chapter 5

Proposed Method

As we showed in the previous section, historical population information alone is insufficient to forecast population in the vicinity of large gathering places. This is because the model is not aware of the days when events with many attendees (e.g., baseball games and concerts) are held.

A simple solution to this problem is to use the event schedules published by venue managers or event organizers. However, the number of gathering places for which official schedules are published is limited. To make predictions for various types of places and events, it is desirable to be able to automatically collect information about future events without relying on such schedules.

In this study, we focus on the fact that information about future events is posted on microblogs. To be specific, we propose a method that utilizes heterogeneous data consisting of microblog posts about events and historical population information on gathering places (Section 5.1). There are various kinds of gathering places, which we study in this thesis, from sports venues to sightseeing spots. This difference sets up the various forecasting situation for each place. For example, population at sports venues drastically influenced by sports games, which cause the surge of thousands of people. In contrast, population at sightseeing spots changes gradually due to moderate factors such as seasonal events. In addition to these differences, these gathering places have various key properties regarding forecast: the amount of available posts, the frequency of events, etc. To handle these different forecasting situations, we propose two models: one is a tree-based model (Section 5.2) and the other is an NN-based model (Section 5.3).

5.1 Microblog Posts as an Event Indicator

As an indicator of future population at large gathering places, we utilize microblog posts that contain time-referring expressions and a place name. We consider that a post which is useful for predicting population on a future date d at a gathering place v should meet all of the following conditions:

- It contains a time-referring expression to d
- It contains the place name v
- It was created before d

To increase the coverage of event-related posts that meet these conditions, we expanded the search query to cover variations of time-referring expressions and place names. See Section 6.1.2 for detail.

When we have these posts, a naive approach to population forecasting at gathering places is to train a regression model with the number of these matched posts and the population on the corresponding date. However, this approach is far from satisfactory since there is no clear correlation between the number of posts and the population (Figure 4). We investigated the contents of matched posts and concluded that this variance of population can be explained in two ways: (1) Posts that do not relate to the actual population (e.g., announcement of a concert DVD's release) meet the conditions or (2) the event attracts less attention on microblogs due to the attributes of its attendees, etc. Therefore, it naturally follows that we have to focus on the contents of these matched posts to fully exploit them as an event indicator.

However, the sets of posts collected for each target day have two difficult properties to handle: (1) the numbers of posts are different for each target day and (2) the order of input posts are not related to the population on the target day. This is

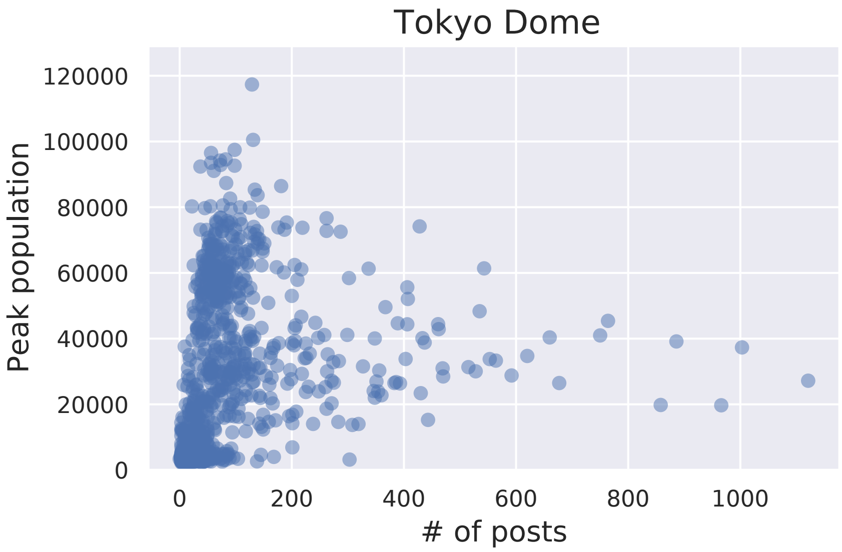


Figure 4 Number of matched posts and peak population at Tokyo Dome.

"Konzatsu-Tokei®" ©ZENRIN DataCom CO., LTD.

an instance of set-input problems. In the rest of this section, we explain the details of these two models and how they address this set-input property.

5.2 Forecasting with Gradient Boosting Regressor

For a tree-based model, we adopt the gradient boosting regressor (GBR) [27]. The motivation for adopting GBR is as follows:

- It can handle data of mixed type. Thanks to this property, we can mix the features from microblog posts and time series of historical population without considering their difference in scale.

- It can be trained with small data. This is preferable since some gathering places are mentioned in few microblog posts, as described later in Section 6.1.2.

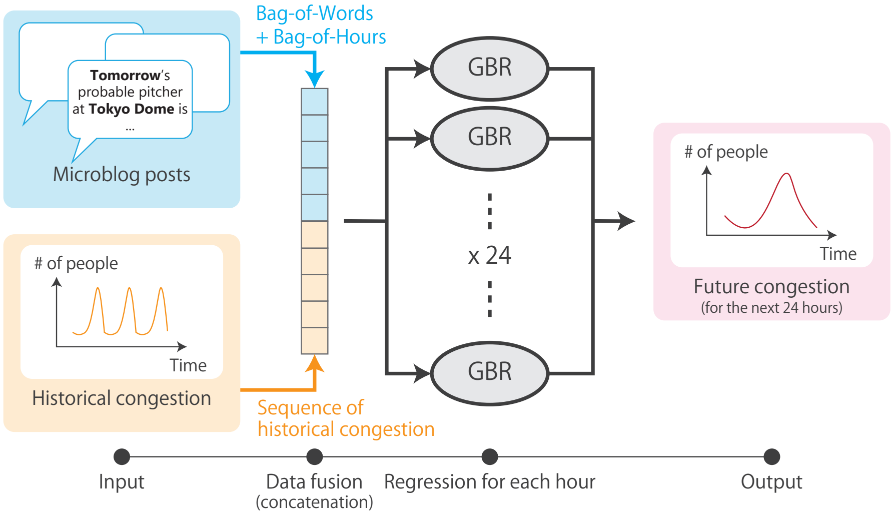

Tomorrow's probable pitcher at Tokyo Dome is ...
Microblog posts
# of people
Time
Historical congestion
Bag-of-Words + Bag-of-Hours
Sequence of historical congestion
GBR
GBR
x 24
GBR
# of people
Time
Future congestion (for the next 24 hours)
Input
Data fusion (concatenation)
Regression for each hour
Output

Figure 5 Overview of our tree-based model.

Figure 5 shows the overview of our tree-based model. Since GBR alone cannot be applied to set-input problems, we adopt the aggregated representation of posts. To be specific, we concatenate the unique posts into a single document and use the bag-of-words representation of it. The bag-of-words representation can handle any number of posts and it is invariant to the permutation of posts. To capture the short-term trend in the time series, we simply concatenate the bag-of-words vector with hourly data on the number of people for $24 \times n$ time steps. We further concatenate the bag-of-*hours* vector with them to leverage hour-referring expressions that take various forms. The bag-of-hours vector is an analogy of the bag-of-words vector. It is a 24-dimensional vector and its i-th dimension is the number of expressions referring to the hour i that appear in the posts set. We covered several variations of hour-referring expressions. See Section 6.1.2 for detail. We then feed the concatenated vector into a regression model. Our model makes a prediction by regression for each hour, taking as input the concatenated bag-of-words, time-series, and bag-of-hours vector, and outputting the number of people at the corresponding hour.

For this method, we set the hyperparameters of GBR to the best values from the

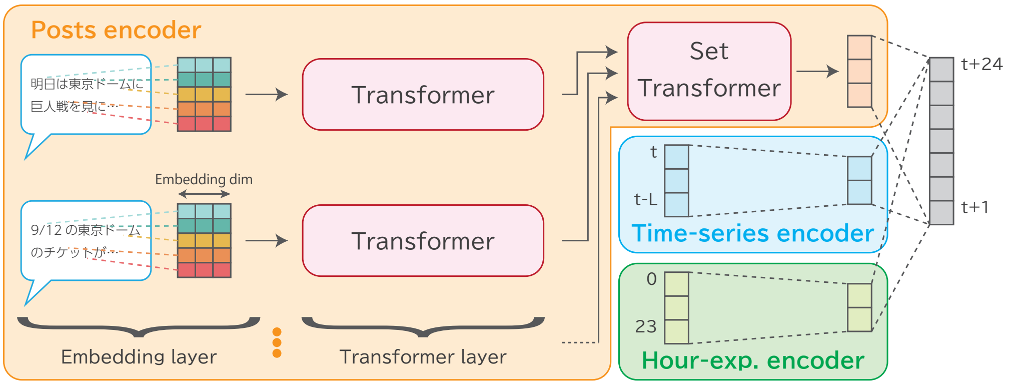


Figure 6 Overview of our NN-based model.

grid search results on the validation set.

5.3 Forecasting with Set Transformer

For an NN-based model, we adopt the Set Transformer (ST) [7]. The motivation for adopting ST is as follows:

- It can process each post separately. In contrast to the bag-of-words approach that transforms a set of posts into a single vector, this approach is expected to be able to consider the complex interactions between posts in the set.
- We can design an encoder for each data source. Although the tree-based model processes the different types of inputs equally, an NN-based model can be customized for fusing the heterogeneous data efficiently. Recent studies on traffic forecasting using NN-based data fusion technique have reported its effectiveness [3, 12].
- For places mentioned in many posts, an NN-based model is expected to perform better because of their large training data.

Figure 6 shows the overview of our NN-based model. ST is an attention-based permutation-invariant module and can be applied to set-input problems. The model consists of the posts encoder, the time-series encoder, and the time-expressions encoder. In the posts encoder, ST encodes a set of post vectors into a single vector. Here, each post is encoded with Transformer [6]. We prepend a `<CLS>` token to each post, and use the Transformer's embedding of the `<CLS>` token as the post vector. In the time-series encoder, to capture the short-term trend in the time series, the hourly-based data on the number of people for $24 \times n$ time steps are fed into a dense layer. Here, n is a hyperparameter that determines the number of days for which the historical time series is used. In the time-expressions encoder, to explicitly focus on the valuable parts of text, i.e., hour-referring expressions, the bag-of-*hours* vector is fed into a dense layer. Following the existing data fusion approach to traffic forecasting [3], we concatenated the embedded representations of the three data source (i.e., posts, counts of time expressions, and time series) and fed them to the succeeding dense layers. Figure 7 shows the overall network architecture of our NN-based model.

As we show later in Section 6.2.2, some of target places we study did not have enough training samples to train the textual encoder (i.e., the number of posts for each target day was small for those places). To reduce the forecast errors for such places, we also tested the fine-tuned version of this model. Specifically, we first trained a model with the concatenated dataset of all places, and then fine-tuned it for each target place with its dataset.

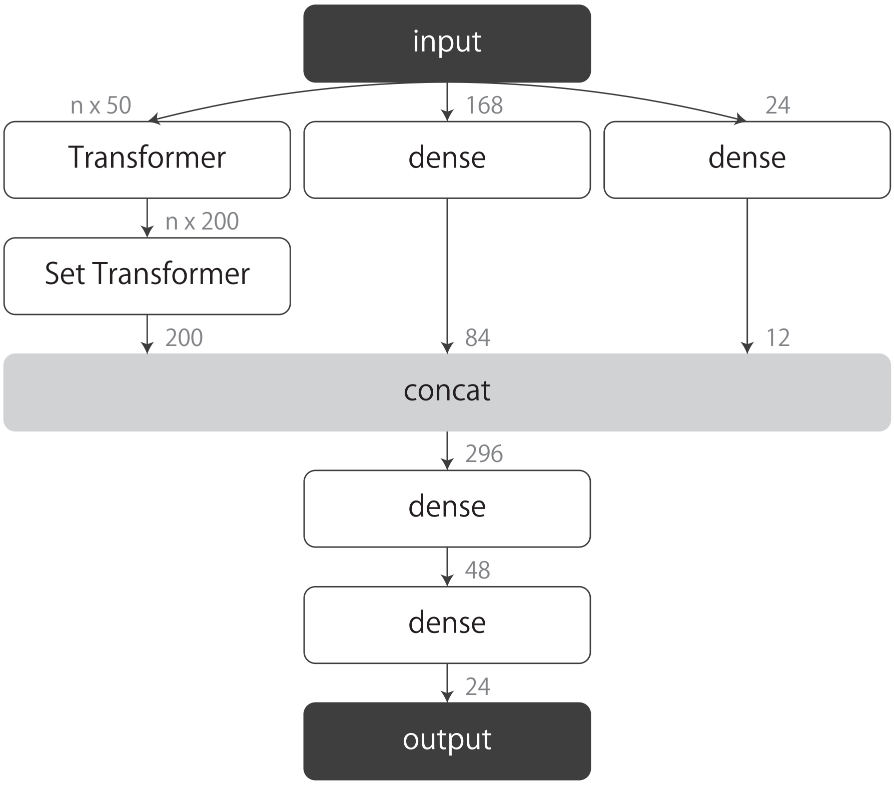


Figure 7 Architecture of our NN-based model. The input splits into three parts: microblog posts (left), historical time series (center), and a time-expressions vector (right).

Chapter 6

Experiments

We examined the effectiveness of our models in experiments with real-world data. Our experiments were designed to answer the following questions:

Q1 Can our models identify event days? (Section 6.3)

Q2 How accurately can our models forecast the population in different settings? (Section 6.4)

Q3 What are the important features? (Section 6.5)

6.1 Preprocessing

Our dataset consisted of two heterogeneous parts: spatio-temporal population data and microblog posting data. In what follows, we detail the preprocessing steps to build the dataset.

6.1.1 Spatio-Temporal Population Data

For the spatio-temporal population data, we used "Konzatsu-Tokei®" Data, which is described in Chapter 4, collected from Dec. 2014 to Nov. 2018. We used the first two years for training (Dec. 2014 – Nov. 2016), the last one year for testing (Dec. 2017 – Nov. 2018), and the remaining one year for developing (Dec. 2016 – Nov. 2017). We chose 15 gathering places in Tokyo and Kanagawa, Japan, from six different event categories (i.e., sports venues, concert halls, exhibi-

Table 1 Description of gathering places.

Type	Place	Capacity
Sports venue	Nissan Stadium	72,327
	Tokyo Dome	55,000
	Ajinomoto Stadium	50,000
	Jingu Baseball Stadium	35,133
	Yokohama Stadium	30,000
	Chichibunomiya Rugby Stadium	24,871
Concert hall	Pacifico Yokohama	18,000
	Yokohama Arena	17,000
	Nippon Budokan	14,471
Exhibition Hall	Tokyo Big Sight	-
Park	Ueno Park	-
	Yoyogi Park	-
Protest venue	National Diet Building	-
Sightseeing spot	Motohakone	-
	Hakone-Yumoto Station	-

tion halls, parks, protest venues, and sightseeing spots) as shown in Table 1. The sports venues and concert halls were the nine largest venues by capacity as of Dec. 2014. Similarly, the exhibition hall was the largest hall by floor space. The parks were the main parks in Tokyo where events were regularly held. For each place, we chose a grid that covers the place and considered the number of people in the grid as the population around the place. Since parks were too wide to be covered by a gird, we chose a grid that covers the area in the park where events were held. For the famous sightseeing area, Hakone, we chose two grids that covers the main spot (Motohakone) and the gateway station (Hakone-Yumoto Station), respectively.

There are cyclic trends in the time series data. They result from daily or weekly

Table 2 Variation of time-referring expressions.

Type	Expression
Absolute reference	平成 Y 年 M 月 D 日，20Y 年 M 月 D 日
Relative reference	M/D，M 月 D 日 来月 D 日，今月 D 日 明日，明後日

commuting and interfere with the population prediction. Thus, it is important to remove these trends from the time-series data analysis [28]. We removed them from both the input and target time series data by simply subtracting the mean time series of non-event days. To obtain the mean time series, we performed k-means clustering on the daily time series in the training data. Since most of the training data consisted of non-event days, we can regard the center of the largest-sized cluster as the representative time series of non-event days. Based on the observations of clustering results with the number of clusters k changed, we set k to three, which usually yielded clear clustering results of two event-day clusters and one large non-event-day cluster. Since the time series of population varies greatly between weekdays and holidays, this clustering-based detrending was applied separately to weekdays and holidays.

6.1.2 Microblog Posting Data

For the microblog posting data, we used posts that were extracted from our Twitter archive.[*3] To expand the coverage of event-related posts meeting the matching conditions (Section 5.1), we handled variations of both time-referring expressions and place names. For time-referring expressions, as shown in Table 2, we assumed

[*3] Our archive has been maintained since Mar. 2011 by continuously crawling with the Twitter API. It consists of timelines from about 2.5 million public users. Our crawling started with 30 famous Japanese users, and the set of users has been repeatedly extended by following retweets and mentions in their timelines.

Table 3 Variation of place names.

Place	Place name
Nissan Stadium	日産スタジアム，横浜国際総合競技場，横浜国際競技場
Tokyo Dome	東京ドーム，ビッグエッグ
Ajinomoto Stadium	東京スタジアム，味の素スタジアム，味スタ
Jingu Baseball Stadium	明治神宮野球場，神宮球場，神宮スタジアム
Yokohama Stadium	ハマスタ，浜スタ，横浜球場
Chichibunomiya Rugby Stadium	秩父宮ラグビー場
Pacifico Yokohama	横浜国際平和会議場，パシフィコ横浜
Yokohama Arena	横浜アリーナ，横アリ
Nippon Budokan	日本武道館
Tokyo Big Sight	東京国際展示場，ビッグサイト
Yoyogi Park	代々木公園
Ueno Park	上野恩賜公園
National Diet Building	国会議事堂
Motohakone	箱根
Hakone-Yumoto Station	箱根

that there were two patterns by which microblog users referred to a specific date: absolute reference and relative reference. Absolute references (i.e., a full combination of a year, a month, and a day) can resolve themselves to specific dates. In contrast, relative references (e.g., *tomorrow*) are resolved to specific dates along with the dates when the posts including them were posted. Some relative references that contained a month and a day but missed a year were resolved to the nearest date of the month and the day. For place names, we created a dictionary of synonyms that maps variants of place names to a formal name using Wikipedia's redirect data (Table 3). This dictionary contains places' official names, popular names, old names, and their abbreviations.

After eliminating duplicates (e.g., auto-generated posts by bots) from the matched posts, we followed the preprocessing steps of hottoSNS-w2v.[*4] Specifically, we removed URLs from these posts and normalized texts by the neologdn package.[*5] We then tokenized them by JUMAN.[*6] We further removed Japanese stop words[*7] and tags.[*8]

The position of keywords (a time-referring expression and a place name) that meet the matching conditions suggests valuable information for forecasting. However, that information is lost when only the original textual contents of posts are simply fed to a prediction model without any annotation. In fact, some of the posts we collected cannot be fully utilized without that information. For example, the following post contains multiple place names and time-referring expressions and the model cannot distinguish the target date corresponding to the target place from other dates:

> 【速報】嵐が5大ドームツアーを行うことが決定しました！【11月6日のナゴヤドームを皮切りに12月27日の東京ドーム公演まで全17公演】

This post is useful to predict the population on Dec. 27 at Tokyo Dome, but suggests no information about the population on Nov. 6 at Tokyo Dome. To handle this kind of posts, the model should focus on the relative position of the target place name and the target date. Thus, we mask the target place name and the target date with special tokens (`<TARGET_PLACE>` and `<TARGET_DATE>`, respectively) as follows:

> 【速報】嵐が5大ドームツアーを行うことが決定しました！【11月6日のナゴヤドームを皮切りに<TARGET_DATE>の<TARGET_PLACE>公演まで全17公演】

[*4] https://github.com/hottolink/hottoSNS-w2v

[*5] https://github.com/ikegami-yukino/neologdn

[*6] http://nlp.ist.i.kyoto-u.ac.jp/?JUMAN

[*7] https://github.com/apache/lucene-solr/blob/master/lucene/analysis/kuromoji/src/resources/org/apache/lucene/analysis/ja/stopwords.txt

[*8] https://github.com/apache/lucene-solr/blob/master/lucene/analysis/kuromoji/src/resources/org/apache/lucene/analysis/ja/stoptags.txt

By masking these keywords, we expect the model to identify useful posts.

To explicitly input the hour when an event will start, we created the bag-of-hours vectors (Section 5.2) for each date. We covered Japanese and English expressions of hours (e.g., 午後 6 時, 18 時, and PM 6) along with colon-separated notations (e.g., 18:00). To stabilize training, we normalized each bag-of-hours vector by dividing itself by its maximum element.

6.2 Experimental Settings

6.2.1 Baselines

We compared our method with two baselines: AR (same as in our preliminary experiment in Chapter 4) and long short term memory (LSTM) [29]. For LSTM, we followed the default hyperparameters of PyTorch's implementation [30]. Note that these baselines did not use the microblog posts. They predicted the population for the next 24 hours in an autoregressive manner (taking the previous output as input). To capture the weekly patterns of the historical time series, the hyperparameter n (Section 5.2) was set to $n = 7$, unless otherwise mentioned.

6.2.2 Evaluation Metrics

We evaluated the effectiveness of our models in two scenarios: a coarse-grained one and a fine-grained one.

The coarse-grained evaluation determined whether a given model can identify event days. As mentioned in Chapter 5, however, the official event schedule is not always available for many gathering places. Thus, we performed k-means clustering on the daily time series for each place and regarded the date as an event day if the daily time series belonged to the largest cluster and a non-event day otherwise. Here, the number of clusters k was set to $k = 3$, based on the observation on the training data: When k was set to $k = 3$, there were typically two smaller clusters that represented event days, and one large cluster that represented non-event days.

Table 4 Number of event days in test data determined by the clustering and number of posts in training data.

Place	# of event days	# of posts
Nissan Stadium	15	6,229
Tokyo Dome	131	82,629
Ajinomoto Stadium	29	15,842
Jingu Baseball Stadium	87	7,842
Yokohama Stadium	72	20,892
Chichibunomiya Rugby Stadium	114	2,489
Pacifico Yokohama	53	21,310
Yokohama Arena	106	52,115
Nippon Budokan	114	41,507
Tokyo Big Sight	80	79,237
Yoyogi Park	21	27,129
Ueno Park	196	3,500
National Diet Building	111	3,081
Motohakone	100	26,319
Hakone-Yumoto Station	203	26,319

“Konzatsu-Tokei®” ©ZENRIN DataCom CO., LTD.

Table 4 shows the threshold and the number of event days in the test data for each place. Based on this criterion, we evaluated the results with precision and recall as a binary classification problem.

The fine-grained evaluation assessed the degree of predicted population. For this scenario, we used the weighted absolute percentage error (WAPE), same as the preliminary experiment (Chapter 4).

Table 5 Performance (precision and recall) of event-day detection.

	AR		LSTM		GBR		ST (fine-tuned)	
Place	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
Nissan Stadium	0.33	0.07	0.00	0.00	0.63	0.80	0.71	0.67
Tokyo Dome	0.87	0.20	0.80	0.18	0.86	0.77	0.78	0.69
Ajinomoto Stadium	0.00	0.00	1.00	0.03	1.00	0.48	0.87	0.45
Jingu Baseball Stadium	0.68	0.31	0.87	0.15	0.77	0.76	0.75	0.69
Yokohama Stadium	0.62	0.18	0.80	0.06	0.95	0.76	0.91	0.74
Chichibunomiya Rugby Stadium	0.68	0.40	0.64	0.62	0.89	0.68	0.56	0.74
Pacifico Yokohama	0.50	0.26	0.30	0.06	0.67	0.58	0.66	0.47
Yokohama Arena	0.40	0.25	0.50	0.19	0.51	0.51	0.52	0.41
Nippon Budokan	0.44	0.25	0.44	0.14	0.55	0.49	0.55	0.48
Tokyo Big Sight	0.48	0.56	0.56	0.29	0.44	0.76	0.56	0.62
Yoyogi Park	0.00	0.00	0.00	0.00	0.59	0.48	0.44	0.57
Ueno Park	0.76	0.67	0.83	0.56	0.74	0.77	0.73	0.56
National Diet Building	0.75	0.71	0.72	0.76	0.80	0.67	0.81	0.66
Motohakone	0.48	0.24	0.00	0.00	0.75	0.33	0.57	0.31
Hakone-Yumoto Station	0.72	0.67	0.82	0.41	0.78	0.58	0.68	0.52

“Konzatsu-Tokei®” ©ZENRIN DataCom CO., LTD.

6.3 Q1: Event-or-Not Prediction

First, we report the results of the coarse-grained evaluation. Table 5 shows the performance of event-day detection of each method, where the event days are considered to be positive examples. The recall values of the AR model and the LSTM model were nearly zero for many places. This implies that they failed to predict the presence of almost all of the events. As well, the precision and recall values of the AR / LSTM model were zero for some places. These uncommon low values are

Table 6 Performance (precision and recall) of non-event-day detection.

	AR		LSTM		GBR		ST (fine-tuned)	
Place	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
Nissan Stadium	0.96	0.99	0.96	1.00	0.99	0.98	0.99	0.99
Tokyo Dome	0.69	0.98	0.68	0.97	0.88	0.93	0.84	0.89
Ajinomoto Stadium	0.92	1.00	0.92	1.00	0.96	1.00	0.95	0.99
Jingu Baseball Stadium	0.82	0.95	0.79	0.99	0.92	0.93	0.91	0.93
Yokohama Stadium	0.83	0.97	0.81	1.00	0.94	0.99	0.94	0.98
Chichibunomiya Rugby Stadium	0.77	0.91	0.83	0.84	0.87	0.96	0.86	0.74
Pacifico Yokohama	0.88	0.96	0.86	0.98	0.93	0.95	0.91	0.96
Yokohama Arena	0.73	0.84	0.74	0.92	0.80	0.80	0.78	0.85
Nippon Budokan	0.72	0.85	0.70	0.92	0.78	0.82	0.78	0.82
Tokyo Big Sight	0.87	0.83	0.82	0.94	0.92	0.72	0.89	0.86
Yoyogi Park	0.94	0.99	0.94	0.99	0.97	0.98	0.97	0.96
Ueno Park	0.66	0.75	0.63	0.86	0.71	0.68	0.60	0.76
National Diet Building	0.88	0.89	0.89	0.87	0.86	0.93	0.86	0.93
Motohakone	0.76	0.90	0.73	1.00	0.79	0.96	0.78	0.91
Hakone-Yumoto Station	0.62	0.67	0.55	0.89	0.60	0.80	0.54	0.69

“Konzatsu-Tokei®” ©ZENRIN DataCom CO., LTD.

partly explained by the small numbers of event days at these places (Table 4). Since few events took place at these places, the input time series of the last week was likely to be a sequence of days when few people gathered and it did not contain any useful clues to forecast the time series on the event days. In contrast, our models consistently achieved much higher performance. By utilizing microblog posts referring to future events as an additional clue, our models successfully distinguished event days from non-event days.

We also report evaluation results focusing on non-event days (Table 6) to confirm that our models could predict the population on non-event days that composed the majority of the test data. Table 6 shows the detection performance of non-event days of each method, where the non-event days are considered to be positive examples. Our models again had high performance for most places. This shows its practicality.

Comparing the results for the different types of places, it can be seen that our models achieved higher performance on baseball stadiums, namely Tokyo Dome, Jingu Baseball Stadium, and Yokohama Stadium, while it had lower performance on concert halls like Yokohama Arena and Nippon Budokan. A possible reason for this difference is that different types of events are held at each place. For example, baseball games are predominantly held at baseball stadiums. The microblog posts that refer to the games typically contain characteristic words, such as the team name whose home stadium is the target place. This led to our method having higher performance at sports places. At concert halls, however, the posts that refer to concerts contain various proper nouns, such as performer names. Consequently, our models had poorer performance on concert halls. Later, we will present our feature importance analysis, whose results support this explanation.

6.4 Q2: Population Prediction

Next, we report the fine-grained results. Figure 8 shows prediction performance for population measured on all of the test data. Our models had lower errors than the baselines did for all places. The improvement was larger for sports places than concert halls, which is consistent with the event-or-not classification experiment reported in Section 6.3.

To identify the situations in which our tree-based and NN-based models had better performance respectively, we examined the relation between prediction errors of these models and numbers of matched posts for the corresponding place (Figure 9). Apart from the places mentioned in extremely few (less than 10,000) posts, for both our tree-based (GBR) and NN-based (ST) models, the more posts

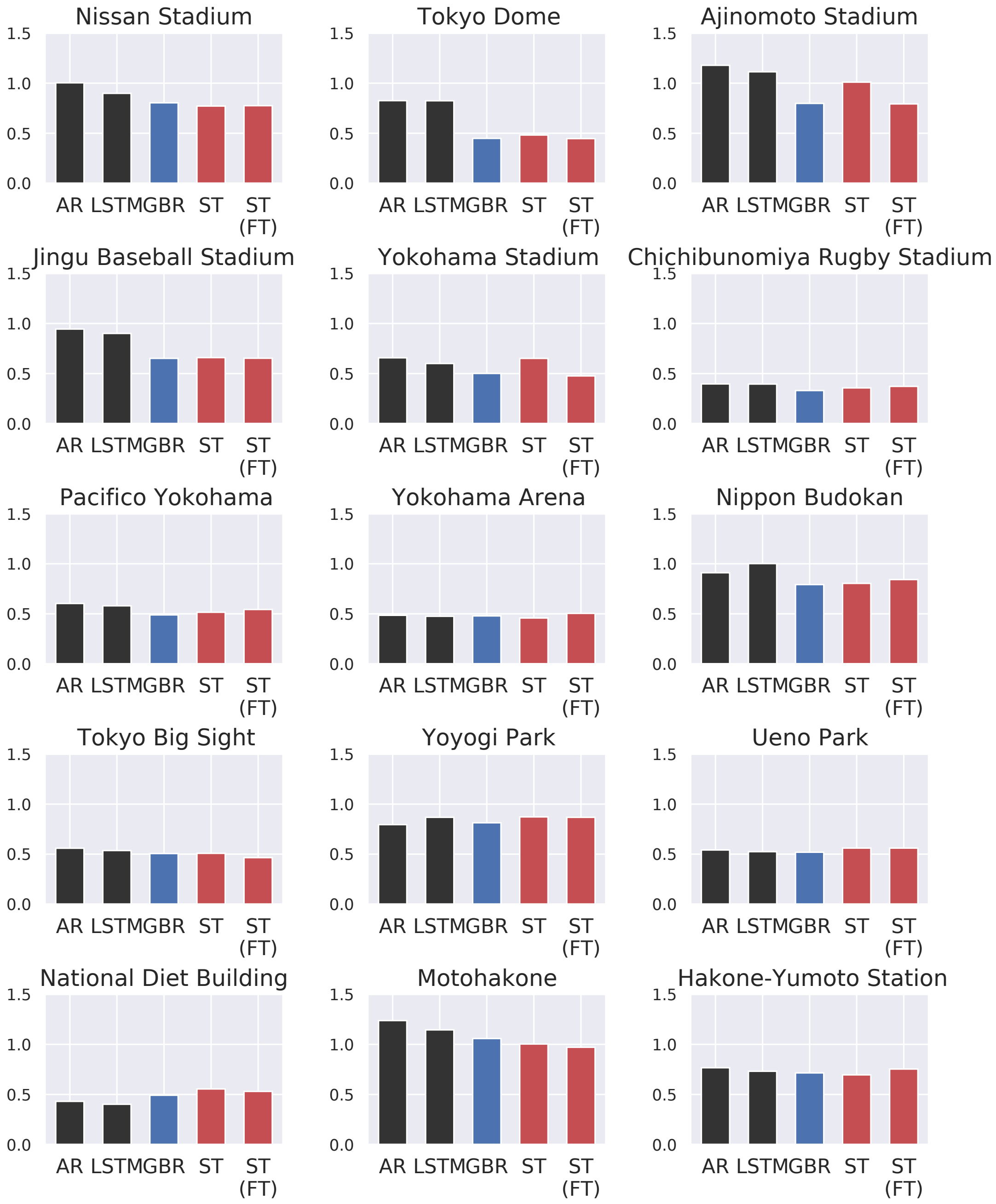


Figure 8 Prediction error (WAPE) on the whole test data.

"Konzatsu-Tokei®" ©ZENRIN DataCom CO., LTD.

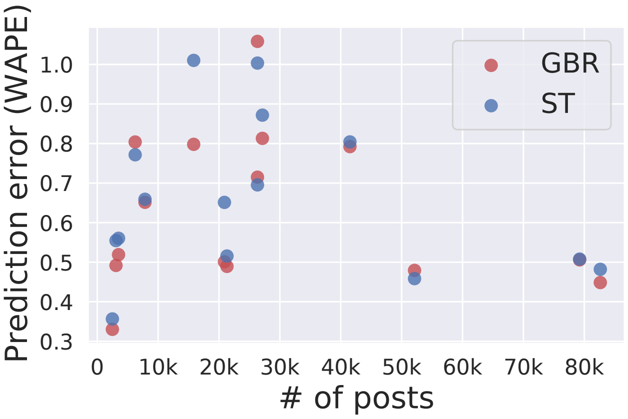


Figure 9 Number of matched posts and prediction error (WAPE) on the whole test data. Note that ST presented here was without fine-tuning.

“Konzatsu-Tokei®” ©ZENRIN DataCom CO., LTD.

were collected, the lower the prediction errors became. An interesting observation here is that our NN-based model showed comparable performance to our tree-based model when more posts were available. Also, it had poorer performance than our tree-based model did when the numbers of posts were relatively small (10,000 – 30,000). This result supports the initial motivation for introducing an NN-based model (Section 5.3), and suggests that it is better to choose an NN-based model for places mentioned in many posts. Similarly, this result also supports the motivation for introducing a tree-based model (Section 5.2) that the better performance is expected for places with small training data.

Comparing the results of ST without fine-tuning with those of ST with fine-tuning, forecast errors at the places with relatively few (10,000 – 30,000) posts (i.e., Ajinomoto Stadium and Yokohama Stadium) were reduced. At these sports venues, the fine-tuned version of our model successfully leveraged the knowledge from the training samples of similar sports games held at the other sports venues.

To further investigate the ability of our models, we conducted the same prediction task under different settings.

6.4.1 Event Day and Non-Event Day

Since most of the test data consisted of non-event days, the evaluation on the whole test data leads to an underestimation of errors. Therefore, we evaluated the prediction performance on event days (Figure 10) and non-event days (Figure 11) separately. Although it was more challenging to predict the population during event days, our models again outperformed the baseline models for all places. The difference was smaller on non-event days. Nevertheless, our method performed the best on all except one place.

6.4.2 Contribution of Each Data Source

To test if our approach of mixing two heterogeneous data sources was effective in forecasting population, we designed ablation tests for each of the two proposed models. Figure 12 shows the results for our tree-based model. We increased the types of input in three stages: (1) BoW, (2) BoW + Texp, and (3) BoW + Texp + TS. Here, BoW is the bag-of-words representation of posts, Texp is a time-expression vector of posts, and TS is the historical time series of population. For most places, BoW + Texp + TS resulted in the lowest error. This demonstrated that these data sources have a complementary role in reducing forecast errors. Utilizing time expressions was particularly effective for sports venues such as Nissan Stadium. This is probably because there were day and night games at sports venues, for which time expressions suggested important clues. Figure 13 shows the results for our NN-based model. Similarly, we increased the types of input in three stages: (1) Text, (2) Text + Texp, and (3) Text + Texp + TS. Overall, the contribution of each data source was similar to that of tree-based model's results.

6.4.3 Forecast Horizon

To test if our model can predict population for longer forecast horizons, we changed the amount of available information prior to prediction. Due to time constraints, we conducted this experiment only for our tree-based model, which

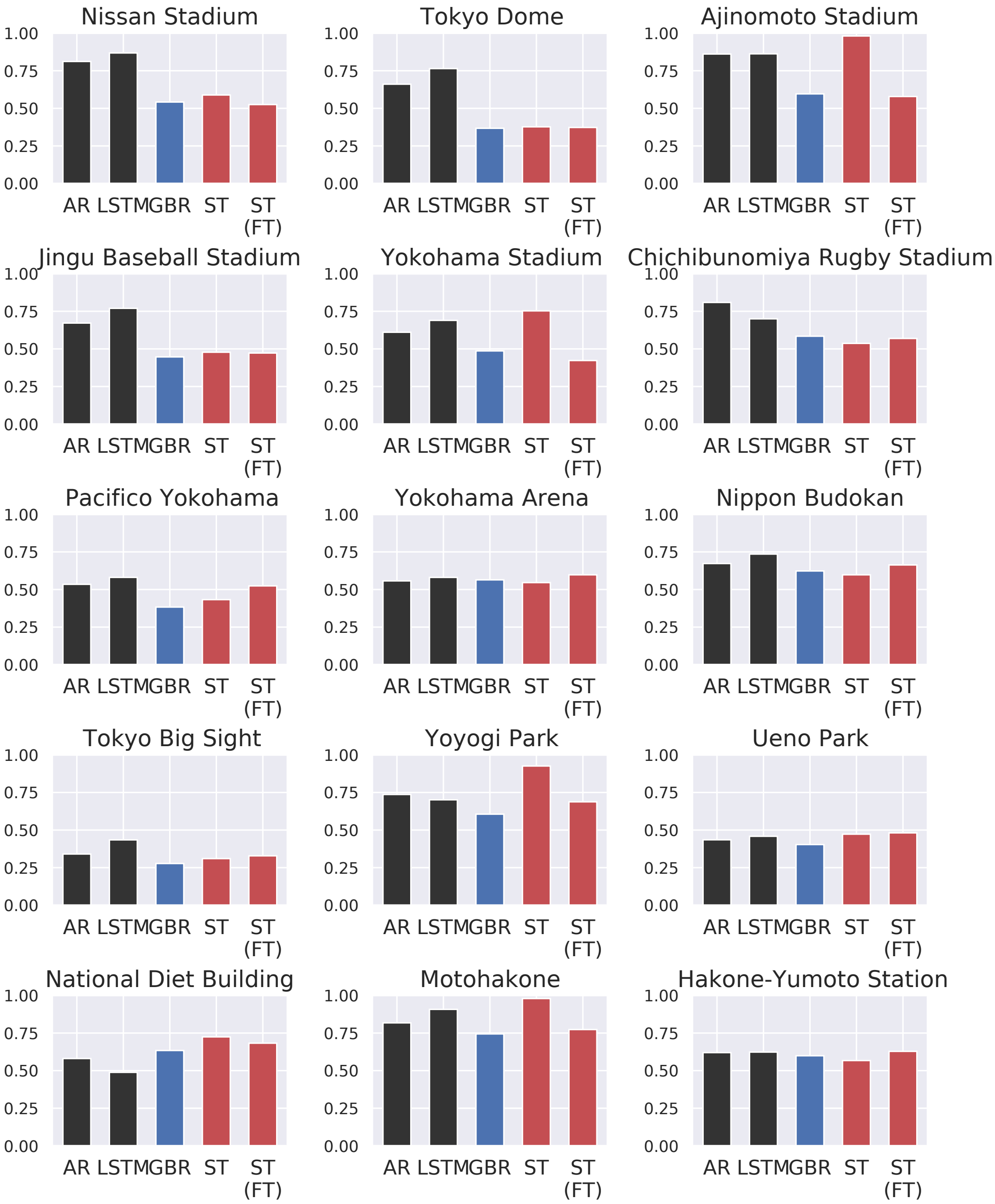


Figure 10 Prediction error (WAPE) during event days on the test data.

“Konzatsu-Tokei®” ©ZENRIN DataCom CO., LTD.

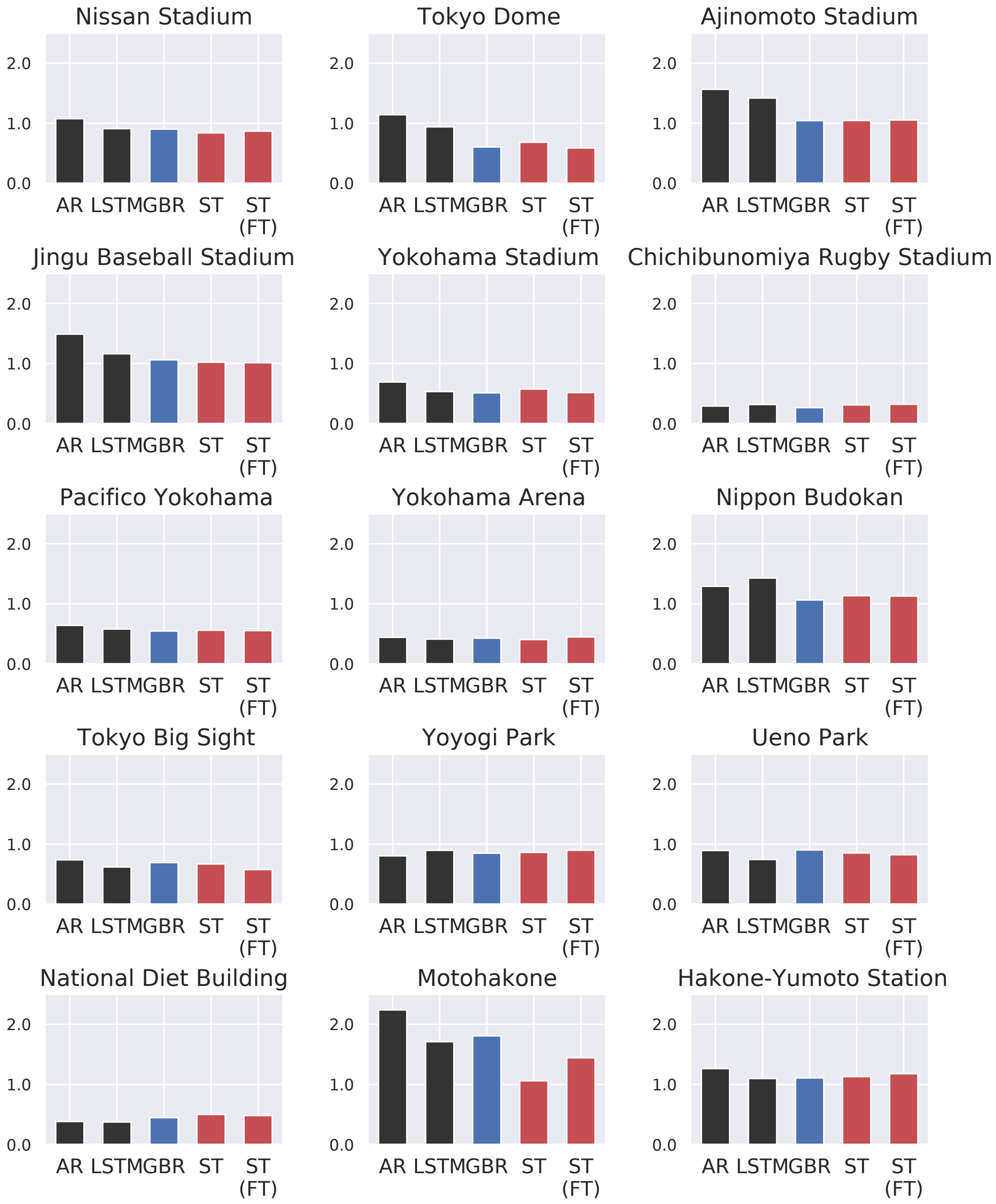


Figure 11 Prediction error (WAPE) during non-event days on the test data.

“Konzatsu-Tokei®” ©ZENRIN DataCom CO., LTD.

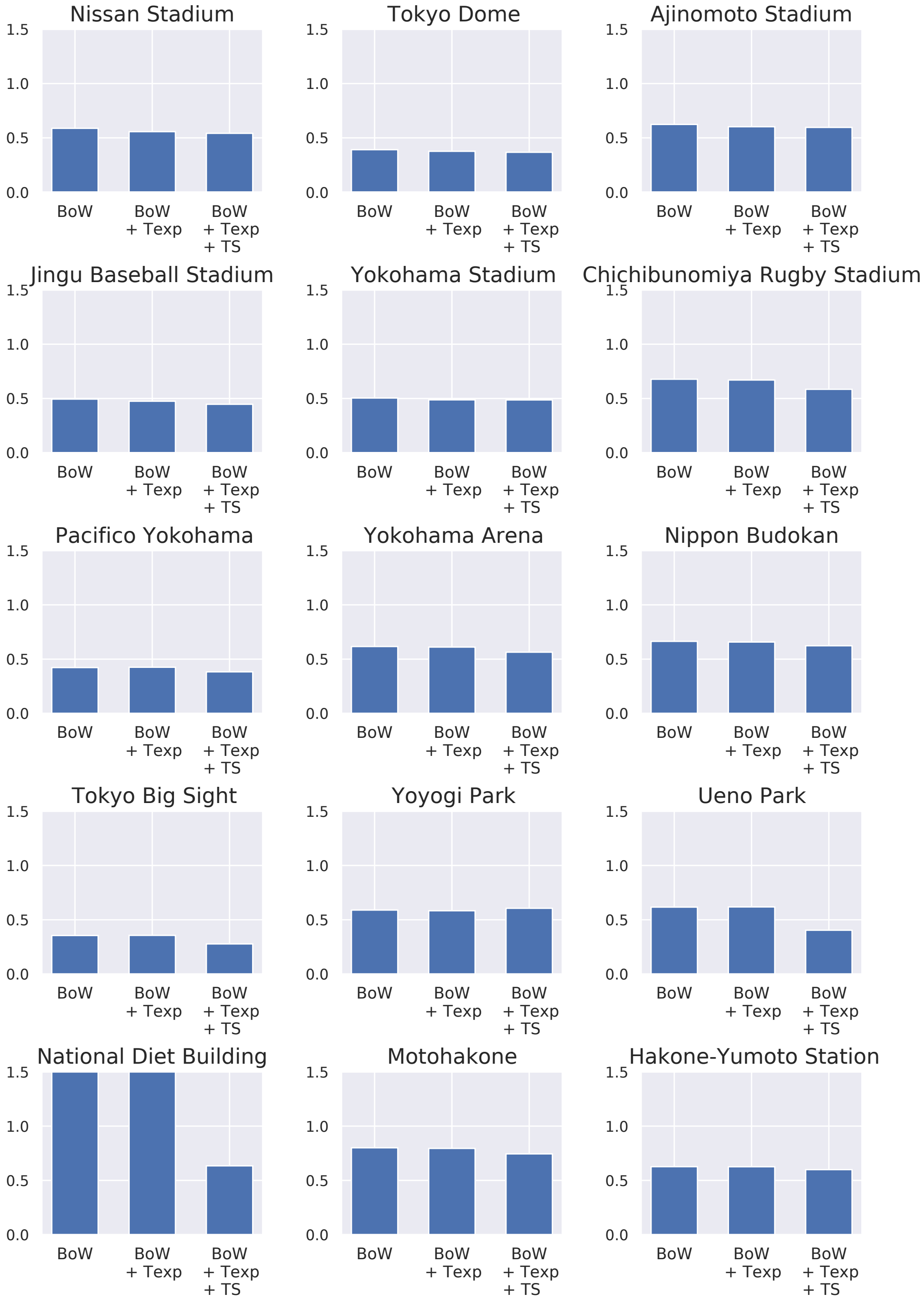


Figure 12 Prediction error (WAPE) on the test data (GBR).

“Konzatsu-Tokei®” ©ZENRIN DataCom CO., LTD.

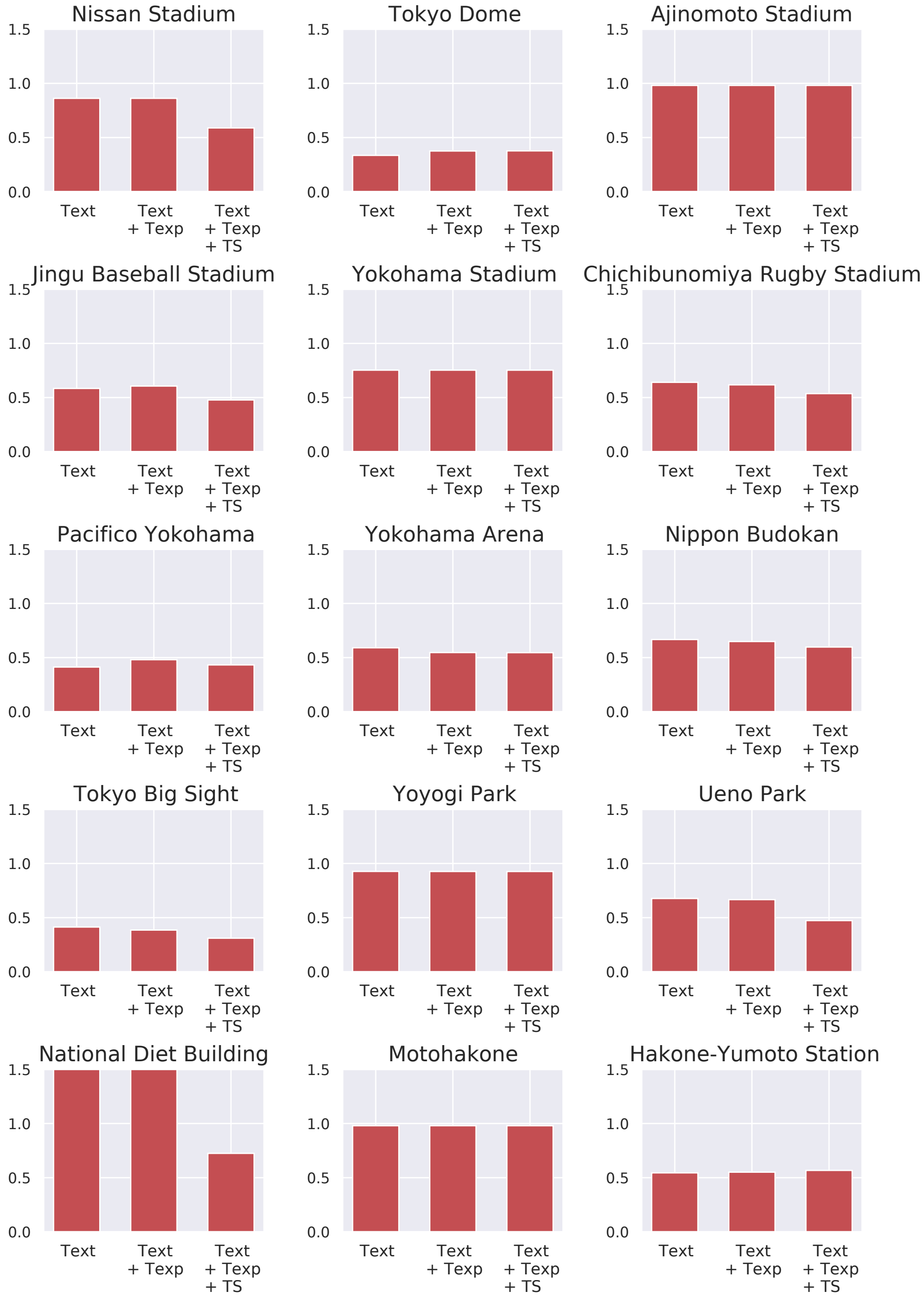


Figure 13 Prediction error (WAPE) on the test data (Set Transformer).

"Konzatsu-Tokei®" ©ZENRIN DataCom CO., LTD.

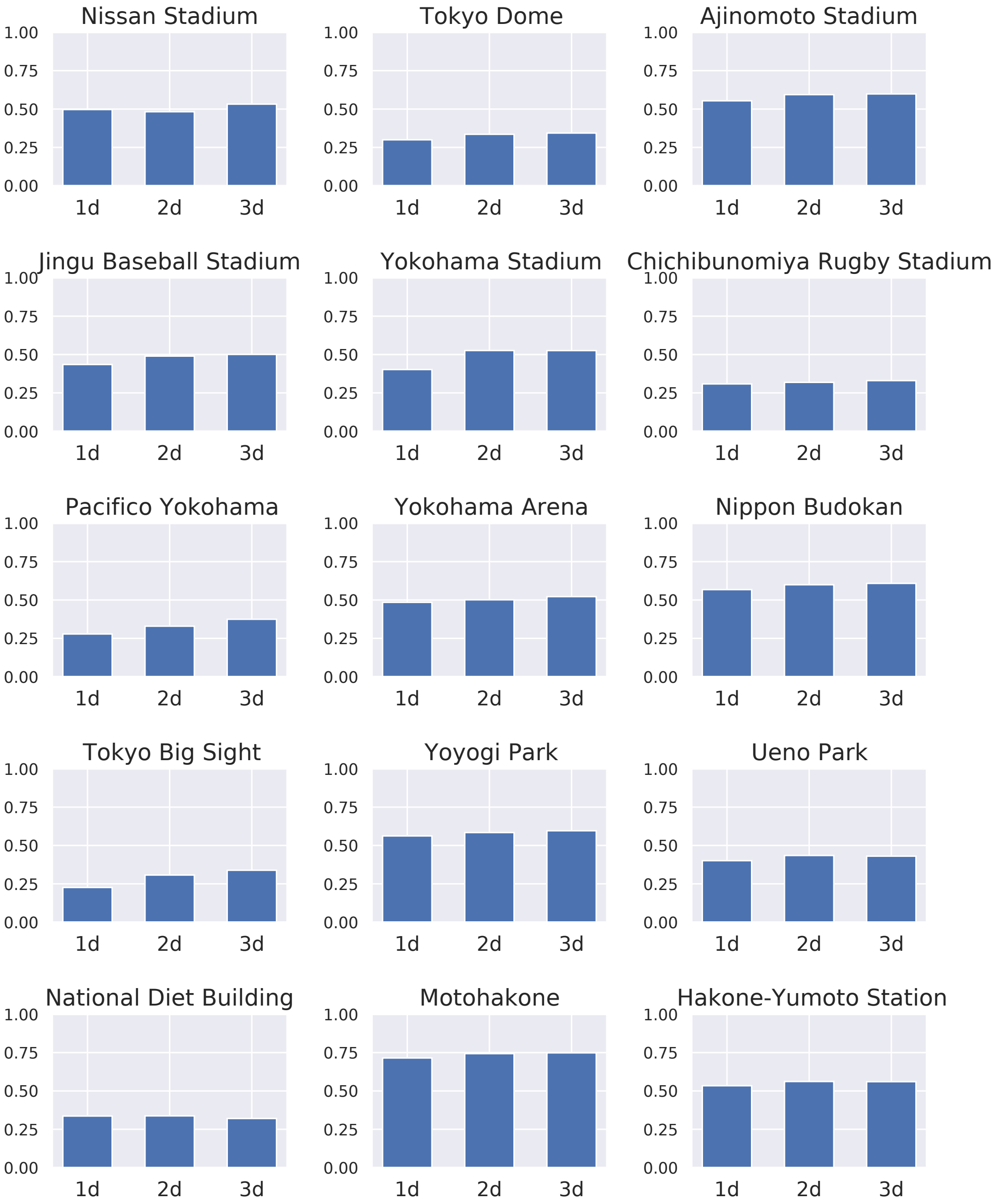


Figure 14 Prediction error (WAPE) during event days on the test data with different forecast horizons.

“Konzatsu-Tokei®” ©ZENRIN DataCom CO., LTD.

was much faster than our NN-based model. Figure 14 shows the results for different forecast horizons. For the d-day ahead prediction, only the posts that were posted d or more days before the target day were used. Similarly, only the historical time series d or more days before the target day was used.

For all places, the longer forecast horizon resulted in a higher error. In particular, the gap between one-day and two-day ahead prediction was the largest. This suggests that the important posts that provide clues to future events tend to be posted on the eve of the events. Another possibility is that the one-day-ahead prediction setting was able to leverage the historical time series to identify the increase in the number of people gathering at the place on the eve of the event while the earlier prediction settings could not do so.

6.4.4 Number of Posts

To check if the number of posts has an impact on prediction performance, we randomly sampled posts in the training and test data at a constant rate and trained the model with the sampled data. Again, we conducted this experiment only for our tree-based model. Figure 15 shows MAE with the reduced number of posts. The experiment where 100% of the posts were used is the same as the experiment shown in Figure 10. When 0% of the posts were used, only the historical time series vector was fed to the model. Conversely, “Only posts” indicates when only the bag-of-words vector was fed to the model.

It can be seen from the table that the more posts there were, the lower the errors became. The errors were highest when the posts were not used at all, suggesting that the microblog posts contain information that is useful for population prediction. Another interesting observation is that the “100%” condition performed better than the “Only posts” condition for most places. The results presented here demonstrate that historical population information and event-related posts have complementary roles in the future population prediction. Therefore, our idea of using these two types of information is promising.

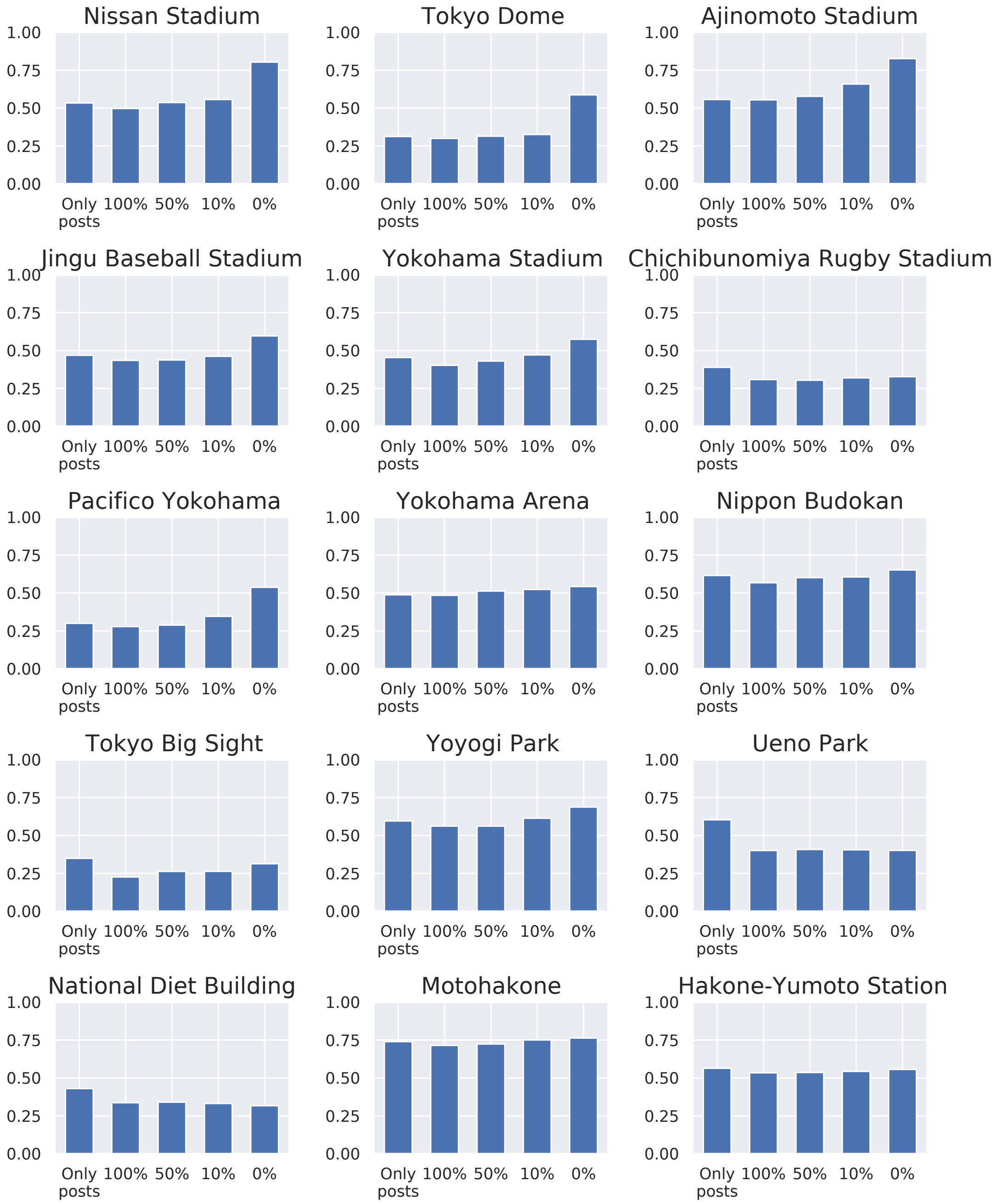


Figure 15 Prediction error (WAPE) during event days on the test data with a reduced number of posts.

“Konzatsu-Tokei®” ©ZENRIN DataCom CO., LTD.

6.5 Q3: Feature Importance / Attention Weights

Finally, we report features useful for predicting future population around gathering places.

6.5.1 GBR

Our model (GBR) is based on a decision tree, and thus, the importance of each feature can be computed [31]. Thus, for each hour, we analyzed words in the posts or slots of the historical time series that were significant for predicting the population at that hour. Table 7 and Table 8 show the ten most important features for five prediction time slots at Tokyo Dome and Nippon Budokan, respectively.

At Tokyo Dome, the word "Giants" (a professional baseball team whose home stadium is Tokyo Dome) ranks high for most hours. This word characterizes the baseball games, which compose the majority of events held there. The words "14" and "18" are important for hours from 12 p.m. to 6 p.m. These words represent the start time (e.g., "The game starts at 18:00...") of baseball games or concerts.

At Nippon Budokan, the sports-related phrases "All Japan Championship" and "tournament" rank high for the early hours. These sports tournaments usually start in the morning. Thus, our model focuses on these words to identify the event type and the start time. For the late hours, concert-related words, "concert", "ticket", and "seat", rank high. These words are used to identify concerts. However, specific performer names (in analogy with the word "Giants" at Tokyo Dome) do not rank high for the late hours, which means our model did not focus on such words for concert halls. This is because there are numerous performers and it is rare that a performer repeatedly holds concerts at the same place.

This analysis partly explains why the prediction errors at concert halls were higher than those at baseball stadiums.

Table 7 Ten most important features for five prediction time slots at Tokyo Dome ([h @ d day(s) ago] represents the population of the place d day(s) ago at h). Note that these features are translated.

12 PM	2 PM	4 PM	6 PM	8 PM
14	14	14	Tokyo Dome	18
[10 AM @ 1 day ago]	kyojin	18	18	Giants
[3 PM @ 1 day ago]	18	probable pitcher	Giants	Tokyo Dome
[11 AM @ 1 day ago]	Giants	Giants	winning	14
day	probable pitcher	kyojin	Paul McCartney	ticket
hometown	[11 AM @ 1 day ago]	[4 PM @ 1 day ago]	seat	Paul McCartney
[8 PM @ 1 day ago]	[4 PM @ 1 day ago]	Tokyo Dome	tomorrow	Pasela
kyojin	[10 AM @ 1 day ago]	[10 AM @ 1 day ago]	14	[8 PM @ 2 days ago]
[11 AM @ 2 days ago]	[8 PM @ 1 day ago]	tomorrow	exchange	winning
schedule	[2 PM @ 1 day ago]	[8 PM @ 1 day ago]	kinki	Southern All Stars

"Konzatsu-Tokei®" ©ZENRIN DataCom CO., LTD.

Table 8 Ten most important features for five prediction time slots at Nippon Budokan ([h @ d day(s) ago] represents the population of the place d day(s) ago at h). Note that these features are translated.

12 PM	2 PM	4 PM	6 PM	8 PM
[12 PM @ 1 day ago]	JSDF Marching Festival	Seiko Matsuda	ticket	[8 PM @ 1 day ago]
August 15	[1 PM @ 1 day ago]	[8 PM @ 3 days ago]	those who	ticket
[1 PM @ 1 day ago]	[12 PM @ 1 day ago]	[5 PM @ 1 day ago]	seat	[7 PM @ 6 days ago]
akb48	full of	Wakayama	[8 PM @ 1 day ago]	piece
All Japan Championship	tournament	[6 PM @ 1 day ago]	gate	[8 PM @ 6 days ago]
[10 AM @ 1 day ago]	Taemin	those who	[8 PM @ 6 days ago]	seat
[7 AM @ 3 days ago]	[12 PM @ 3 days ago]	very	alfee	eric
tournament	stage	hall	July 9	book
[11 AM @ 2 days ago]	Seiko Matsuda	[7 AM @ 4 days ago]	concert	[8 AM @ 4 days ago]
[11 AM @ 1 day ago]	All Japan Championship	concert	[7 PM @ 1 day ago]	adjustment

"Konzatsu-Tokei®" ©ZENRIN DataCom CO., LTD.

6.5.2 Set Transformer

Our NN-based model performs self-attention between words in an input post. We can visualize attention weights for each word to help us understand the behavior of the model. In the rest of this section, we present the analysis on the weighted words for several types of places.

For sports venues, where sports games compose the majority of events held there, we observed that the start time of the games was weighted for most games. The below is a post from an actual input for the day when a baseball match was held at Tokyo Dome. The underlined words were weighted in the self-attention mechanism.

<CLS> <TARGET_DATE> 木 18 時 巨人 対 阪神 <TARGET_PLACE> 指定 席 2 階 3 塁 2400 円 2 連 kyojin hanshin 巨人 阪神 阪神 タイガース giants tigers タイガース 8 <UNK> 発売 中

Here, the start time of the baseball match was weighted. This result coincided with the feature importance analysis of our tree-based model. Another interesting observation is that the team name of visitors was weighted. This suggests that the attendance of a game partly depended on the visitor teams.

For concert halls, where many noisy posts (e.g., the advertisement of concert DVDs) were collected, our model weighted the words that suggested the post containing them was not related to the actual population. The below is a post from an actual input for the day when no event was held at Nippon Budokan.

<CLS> ... KING SUPER LIVE 2017 <UNK> BD <TARGET_DATE> 発売 3 5 <TARGET_PLACE> 開催 れた ライブ イベント 映像 化

Here, the words that announced the release of a concert's Blu-ray disc were weighted. This result supports the hypothesis that our NN-based model distinguished useful posts from noisy posts.

For parks, where non-scheduled public gatherings sometimes happen, the words related to them were weighted. The below is a post from an actual input for the day of *hanami* (public gatherings of cherry blossom viewing) season at Ueno Park.

<CLS> <UNK> 予想 東京 埼玉 桜 開花 <UNK> 迎え ました <TARGET_PLACE> 来週 後半 <UNK> <UNK> 堂 桜 <UNK> <TARGET_DATE> 頃 花見 楽しめ そう

Here, the words related to cherry blossoms were clearly weighted. This shows that our model was powerful enough to automatically detect future non-scheduled events in microblog streams.

Chapter 7

Conclusion

In this study, we tackled the problem of forecasting population at gathering places.

Firstly, we conducted a preliminary experiment, where the traditional autoregressive model was trained and tested to forecast the city-wide population at Tokyo. Its result showed that historical population information alone is insufficient for forecasting, as it cannot capture the surge of people caused by non-recurrent events held at gathering places.

Secondly, to automatically find clues about future events at general gathering places, we leveraged microblog posts mentioning the target places and the target days as additional features along with historical population information for training a population forecasting model. Since our proposed approach essentially includes a set-input problem, which is quite different from typical instance-based problems, we introduced aggregation mechanisms of microblog posts to our models. Using the spatio-temporal population data of Tokyo and Kanagawa, Japan and our large-scale Twitter archive, we empirically demonstrated that our approach was effective in reducing forecast errors of the baseline models, which only utilized historical population information. Our ablation study on these data sources provided evidence that they have a complementary role in improving population forecasting at general gathering places.

Finally, our analysis on feature importance and attention weights suggested that our models focused on the understandable textual features of both event and non-event days to predict the time and scale of the incoming population surge. Also, the analysis showed that our models successfully detected the future non-scheduled public gatherings in the microblog streams and forecasted the population surge with them. This wide range of forecasting targets is clearly different from existing approaches of population forecasting at event venues for which event schedules are available.

Chapter 8

Future Work

Since our proposed approach does not require event schedules, we can easily extend it to many gathering places, without any additional labor-intensive process. The experiments showed that, however, our models had lower performance when few posts referred to the future events. Thus, in the future, we will consider using knowledge acquired from the places or events mentioned in many posts to improve the prediction of population at gathering places mentioned in few posts. There is still room for improvement of the matching conditions of posts (Section 5.1). Some of the posts that meet these conditions may be noisy. For example, a post that says "The DVD of our Tokyo Dome concert will be released tomorrow!" meets these conditions, but it does not suggest any information about the population at Tokyo Dome tomorrow. On the other hand, posts that do not meet these conditions can also be useful. There are many event-related posts that do not contain both dates and place names. We thus need to devise a sophisticated method of data fusion to pick up important information from such noisy but valuable posts. We also plan to make the forecasting results explainable. For example, if our forecasting model can present an actual post that suggests a population surge at a gathering place with the forecasted time series there, users can choose whether or not they get into the congestion in a more confident way.

Acknowledgements

I would like to express my deepest appreciation to my supervisor, Professor Masashi Toyoda, for his continued and patient support. Whenever I got stuck, he gave me valuable advice and insightful suggestion. Thanks to him, my vague interest in social and traffic big data was led to complete this research project. I would also like to thank Assistant Professor Kazutoshi Umemoto. He always gave me a lot of essential comments and pointed out what I had totally missed. I am also grateful to Associate Professor Naoki Yoshinaga. With his enthusiastic help, my conference papers and presentation slides became highly sophisticated from both perspectives of their contents and styles.

I would like to express my gratitude to my conference papers' co-authors. Mr. Haosen Zhan has supported me many times with his NLP experience. Dr. Shonosuke Ishiwatari has helped me through constructive discussions whenever I needed, even after he left our laboratory. Specially Appointed Assistant Professor Haichuan Shang gave me useful comments on my papers, which guided me to revise them.

I would like to show my greatest appreciation to Professor Masaru Kitsuregawa and all the researchers in the Kitsuregawa-Toyoda-Nemoto-Yoshinaga-Goda laboratory. They have provided me with comfortable research environment including powerful computation resources, valuable datasets, and high-availability servers. Without them, I could not have continued this research project in this COVID-19 spreading situation. I owe a very important debt to the secretaries. I have always received quick and warm supports from them. I would also like to thank the students in our laboratory. They are always eager to give me helpful comments from their various background. Mr. Shoetsu Sato has continuously supported me, a beginner of neural networks, even just before his doctoral defense. Mr. Hiroki Shimizu has organized a reading group for mathematical books many times.

Finally, I would like to thank my parents for their unconditional support.

References

[1] CNN. ‘I failed to protect you’ – Details emerge of victims in deadly Shanghai stampede. https://edition.cnn.com/2015/01/02/world/asia/china-shanghai-new-years-stampede/, 2015.

[2] Zipei Fan, Xuan Song, Ryosuke Shibasaki, and Ryutaro Adachi. CityMomentum: An online approach for crowd behavior prediction at a citywide level. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 559–569. ACM, 2015.

[3] Binbing Liao, Jingqing Zhang, Chao Wu, Douglas McIlwraith, Tong Chen, Shengwen Yang, Yike Guo, and Fei Wu. Deep sequence learning with auxiliary information for traffic prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 537–546. ACM, 2018.

[4] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pp. 1655–1661. AAAI Press, 2017.

[5] Tatsuya Konishi, Mikiya Maruyama, Kota Tsubouchi, and Masamichi Shimosaka. CityProphet: City-scale irregularity prediction using transit app logs. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 752–757. ACM, 2016.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, Vol. 30, pp. 5998–6008. Curran Associates, 2017.

[7] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set Transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 3744–3753. PMLR, 2019.

[8] Diya Li and Mohammed J Zaki. RECIPTOR: An effective pretrained model for recipe representation learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1719–1727. ACM, 2020.

[9] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in Neural Information Processing Systems*, Vol. 30, pp. 3391–3401. Curran Associates, 2017.

[10] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[11] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, and Xiuwen Yi. DNN-based prediction model for spatio-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 1–4. ACM, 2016.

[12] Filipe Rodrigues, Ioulia Markou, and Francisco C. Pereira. Combining time-series and textual data for taxi demand prediction in event areas: A deep learning approach. *Information Fusion*, Vol. 49, pp. 120–129, 2019.

[13] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes Twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, pp. 851–860. IW3C2, 2010.

[14] Chao Zhang, Guangyu Zhou, Quan Yuan, Honglei Zhuang, Yu Zheng, Lance

Kaplan, Shaowen Wang, and Jiawei Han. GeoBurst: Real-time local event detection in geo-tagged tweet streams. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 513–522. ACM, 2016.

[15] Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. Open domain event extraction from Twitter. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1104–1112. ACM, 2012.

[16] Wataru Yamada, Daisuke Torii, Haruka Kikuchi, Hiroshi Inamura, Keiichi Ochiai, and Ken Ohta. Extracting local event information from micro-blogs for trip planning. In *Proceedings of the 8th International Conference on Mobile Computing and Ubiquitous Networking*, pp. 7–12. IEEE Computer Society, 2015.

[17] Adam Jatowt, Émilien Antoine, Yukiko Kawai, and Toyokazu Akiyama. Mapping temporal horizons: Analysis of collective future and past related attention in Twitter. In *Proceedings of the 24th International Conference on World Wide Web*, pp. 484–494. IW3C2, 2015.

[18] Masaki Onishi and Shinnosuke Nakashima. Mutual interaction model between the number of people in real space and the number of tweets in virtual space. In *Proceedings of the 23rd International Conference on Pattern Recognition*, pp. 2073–2078. IEEE Computer Society, 2016.

[19] Satoshi Miyazawa, Xuan Song, Tianqi Xia, Ryosuke Shibasaki, and Hodaka Kaneda. Integrating GPS trajectory and topics from Twitter stream for human mobility estimation. *Frontiers of Computer Science*, Vol. 13, No. 3, pp. 460–470, 2019.

[20] Jingrui He, Wei Shen, Phani Divakaruni, Laura Wynter, and Rick Lawrence.

Improving traffic prediction with tweet semantics. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pp. 1387–1393. AAAI Press, 2013.

[21] Fernando Terroso-Sáenz, Jesús Cuenca-Jara, Aurora González-Vidal, and Antonio F Skarmeta. Human mobility prediction based on social media with complex event processing. *International Journal of Distributed Sensor Networks*, Vol. 12, No. 9, p. 5836392, 2016.

[22] Ming Ni, Qing He, and Jing Gao. Forecasting the subway passenger flow under event occurrences with social media. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 18, No. 6, pp. 1623–1632, 2016.

[23] Minh-Son Dao, Ngoc-Thanh Nguyen, R Uday Kiran, and Koji Zettsu. Fusion-3DCNN-max3P: A dynamic system for discovering patterns of predicted congestion. In *Proceedings of the 2020 IEEE International Conference on Big Data*, pp. 910–915. IEEE Computer Society, 2020.

[24] Balsam Alkouz and Zaher Al Aghbari. SNSJam: Road traffic analysis and prediction by fusing data from multiple social networks. *Information Processing and Management*, Vol. 57, No. 1, p. 102139, 2020.

[25] Aniekan Essien, Ilias Petrounias, Pedro Sampaio, and Sandra Sampaio. A deep-learning model for urban traffic flow prediction with traffic events mined from twitter. *World Wide Web*, 2020.

[26] Administrative Management Agency. Standard grid square and grid square code used for the statistics. `https://www.stat.go.jp/english/data/mesh/02.html`, 1973.

[27] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, Vol. 29, No. 5, pp. 1189–1232, 2001.

[28] Zhaohua Wu, Norden E. Huang, Steven R. Long, and Chung-Kang Peng. On the trend, detrending, and variability of nonlinear and nonstationary time series. *Proceedings of the National Academy of Sciences*, Vol. 104, No. 38, pp. 14889–14894, 2007.

[29] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with LSTM. In *Proceedings of the 9th International Conference on Artificial Neural Networks*, Vol. 2, pp. 850–855. IEE, 1999.

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, Vol. 32, pp. 8026–8037. Curran Associates, 2019.

[31] Leo Breiman, Jerome Friedman, Charles J. Stone, and Richard A. Olshen. *Classification and regression trees.* CRC Press, 1984.

Publications

International conferences and workshops (refereed)

1. Ryotaro Tsukada, Haosen Zhan, Shonosuke Ishiwatari, Masashi Toyoda, Kazutoshi Umemoto, Haichuan Shang, and Koji Zettsu. Crowd Forecasting at Venues with Microblog Posts Referring to Future Events. In *Proceedings of the 5th IEEE International Workshop on Big Spatial Data*, 2020.

Domestic conferences (non-refereed)

1. 塚田 涼太郎，詹 浩森，石渡 祥之佑，豊田 正史．マイクロブログおよび携帯電話人口統計を用いた大規模イベント会場における人口変化の長期予測．第 12 回データ工学と情報マネジメントに関するフォーラム，2020.

2. 塚田 涼太郎，詹 浩森，石渡 祥之佑，豊田 正史，梅本 和俊，商 海川，是津 耕司．未来のイベントに言及するマイクロブログ投稿を用いた人口変化の予測．第 13 回データ工学と情報マネジメントに関するフォーラム，2021（発表予定）.