Building Location Embeddings from Physical Trajectories and Textual Representations

Laura Biester, Carmen Banea, Rada Mihalcea
Computer Science & Engineering, University of Michigan, USA
{lbiester,carmennb,mihalcea}@umich.edu

Abstract

Word embedding methods have become the defacto way to represent words, having been successfully applied to a wide array of natural language processing tasks. In this paper, we explore the hypothesis that embedding methods can also be effectively used to represent spatial locations. Using a new dataset consisting of the location trajectories of 729 students over a seven month period and text data related to those locations, we implement several strategies to create location embeddings, which we then use to create embeddings of the sequences of locations a student has visited. To identify the surface level properties captured in the representations, we propose a number of probing tasks such as the presence of a specific location in a sequence or the type of activities that take place at a location. We then leverage the representations we generated and employ them in more complex downstream tasks ranging from predicting a student's area of study to a student's depression level, showing the effectiveness of these location embeddings.

1 Introduction

Due to the rising adoption of smartphones over the past decade, the number of services with full or partial information about people's spatial mobility has skyrocketed. Inspired by the natural language processing (NLP) literature, we investigate various properties of location embeddings. We explore whether valuable information is encoded in individual location embeddings, as well as embeddings that encompass a sequence of locations. We begin by exploring whether they are able to represent aspects such as location presence or location functionality. Ultimately, we test the hypothesis that if enough underlying information is encoded, embedding models should aid in predicting user-centered descriptors, such as area of study, academic status, or mental health.

Location data can be used by university administrators for applications that improve student life. From the frequency and the type of locations accessed in one's daily routine, we may be able to identify someone who is depressed or someone who is overworked. Importantly, opt-in frameworks can be established to supplement existing counseling and advising offices, allowing for early intervention in the case of mental health and academic concerns. With proper privacy safeguards in place, such models could readily be applied on most university campuses, as WiFi connection data (from which we infer location) is likely already available. In addition, universities could use this data in an aggregate form to better understand student life and well-being, and find ways to promote healthy and engaging behaviors on campus. Such aggregate location information can also be used by architectural firms or municipalities to help with the selection of buildings' locations, architecture, and design; with road and pedestrian traffic optimization; or for emergency response.

We also know that such data is already available to large technology companies that track their users, and it is important to spread awareness about the personal information that can be gleaned. Research like ours helps inform users about privacy concerns, and may open up a path to stricter legislation regarding the use of such data in the future. While we envision numerous positive applications of these methods, there are clear privacy drawbacks that the public should be aware of in the current technological environment.

Our work focuses on building an understanding of what information is encoded in location embeddings. In addition to creating embeddings using location trajectories, we propose an alternative method that synthesizes text from online sources to build representations that we hypothesize will better encode certain properties of locations. We

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 425–434
December 4 - 7, 2020.

show that using dense location embeddings that incorporate both movement patterns and text data improves our ability to model downstream tasks. We see that although we are not able to recover as much surface level information from embeddings of location sequences as we are from a simpler representation, the additional semantic information that is encoded allows us to better predict some user attributes.

2 Related Work

Embedding Evaluation and Probing. Word embeddings are now widely used to create word representations using methods such as word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), ELMo (Peters et al., 2018), and BERT (Devlin et al., 2019). BERT and ELMo can be used to create contextualized word embeddings, in which the vector representing an individual word varies depending on the context in which it appears. Previous methods including word2vec and GloVe did not make this distinction; adding context helped BERT achieve state-of-the-art results on many downstream NLP tasks. One traditional benchmark for word embeddings is performance on synthetic tasks, such as word similarity and word analogy tasks (Mikolov et al., 2013; Pennington et al., 2014). However, word embeddings are widely used because of their superior performance on a variety of downstream NLP tasks when compared to other word representations. Performance on downstream tasks has been used to evaluate sentence embeddings, however such approaches cannot gauge the content that is actually captured in the embeddings. To systematically ascertain what information is encoded in sentence vectors, researchers have turned to probing tasks (Shi et al., 2016; Adi et al., 2017; Conneau et al., 2018). These are meant to address the question "what information is encoded in a sentence vector" at a higher level.

In our work, we find inspiration in the research by Conneau et al. (2018), who propose a formalized evaluation technique for sentence embeddings using a suite of ten classification tasks focusing on: (1) surface information (e.g., length, word content), (2) syntactic information (e.g., bigram shift, tree depth), and (3) semantic information (e.g., tense). The deep learning methods gave the best results overall, but the bag-of-vectors approach was a solid baseline for the word content task, where it outperformed the deep learning models.

Applications of Embeddings for Location Data. Liu et al. (2016) were among the first to use the skip-gram model on location data. They use locations visited before and after a target location as context to create location embeddings. These are then used in a personalized location recommendation system. Feng et al. (2017) similarly create embeddings of check-in data, but use the CBOW model. Their application task is reversed, predicting future visitors for a location instead of predicting locations that a user will visit. Chang et al. (2018) also predict next check-ins for users using a model based on skip-gram. Their work is uniquely related to ours in that they also build prediction of the text content of check-ins into the objective function. Zhu et al. (2019) trained a skip-gram model to build location embeddings, and use them to understand the flow between urban locations. Crivellari and Beinat (2019) explore location embeddings from the perspective of geoinformatics, paving the way for our probing tasks.

The work of Solomon et al. (2018) is most similar to our own. They use GPS data from cell phones as input to create embeddings and use data from a university setting. Our work differs in that we use the skip-gram model and incorporate text-based embeddings. We also propose probing tasks to better understand the embeddings that we create, and predict additional user attributes from our new dataset that go beyond demographic information.

3 Data

3.1 Student and Location Data

Our dataset consists of location data collected from 729 undergraduate university students who agreed to participate in our study in 2018 and 2019 over a period of seven months.[1] Two-thirds of the students participated during the winter semester, and the other third during the fall semester. Dataset statistics are presented in Table 1.

Due to the sensitivity and scope of the data, it is infeasible for our study to include other universities; nonetheless, we believe that similar patterns would hold on other campuses as well. Because of privacy concerns, we are not able to publicly release this dataset.

[1]The data was collected as part of a study that underwent a full board review and was approved by the IRB at the University of Michigan (study number HUM00126298). All participants in the study have signed an informed consent form.

Number of Participants	729
Valid Location Visits After Pre-Processing	478,329
Unique Locations	194
Mean Locations per Participant	656.2
Mean Locations per Day	4.7

Table 1: Statistical summary of the location dataset.

While most similar research uses GPS (Solomon et al., 2018), mobile check-ins (Feng et al., 2017; Liu et al., 2016), or cell phone pings (Zhu et al., 2019) for location tracking, we collect location data from WiFi access logs. WiFi access logs provide a strong and unbiased location signal on campus, as most students carry their smart phones with them at all times; however, a downside is that we do not have location data for large time chunks when students are not connected to the campus WiFi.

The original data consists of 20,766,750 WiFi session updates across all the students. We only consider connections with uninterrupted updates from a single building (without a connection to a network in another building) for at least ten minutes. This ensures that a student's location will not be mapped to multiple points during overlapping time spans, and that locations where a student does not spend a notable amount of time are excluded. After collecting this list of locations, start, and stop times, we perform a merging operation on the data, sorted by start time. If spans for the same location occur consecutively in the series with start and stop times less than 30 minutes apart, those spans are merged together.

After this pre-processing, we are left with 478,329 valid location spans with start and stop times. Since our dataset covers a single campus (194 locations), each location was manually labeled with its functionality, for a total of thirteen functionalities. The five most frequent are: class, study, dorm, lab, and library. While there are 194 locations in the location dataset, we utilize 132 in our analysis because this set of locations appears in all of the text-based datasets (described in Section 3.2); the ones that are left out are not among the most frequently visited.

In addition to location data, we collected a rich dataset containing information about the 729 students, consisting of a series of extensive surveys taken by the students throughout the semester and academic data from the registrar. From the survey data, we use information on class year, gender, depression, and sleep satisfaction. From the academic data, we utilize the GPA and the school where the student is enrolled. These combined data sources are used for our downstream classification tasks. We chose students for the study covering all undergraduate class years, genders, and academic disciplines.

Dataset	Campus Website	Reddit	Twitter
Overall Tokens	581K	882K	655K
Unique Tokens GloVe	9K	11K	18K
Median Instances Per Loc.	3.5	20.0	166.5
Start Date (year-month)	N/A	2011-05	2010-09
End Date (year-month)	2019-05	2019-07	2019-08

Table 2: Statistical information about text datasets.

3.2 Text Data

In addition to location trajectories, we use text data from three sources (campus website, Reddit, Twitter) that illustrate various ways in which text can be used to represent places. Statistics of the text datasets are shown in Table 2.

Campus Website. With this dataset, we capture how people *formally define* locations. The university hosts a building search website that links to pages containing information about campus buildings, including the departments hosted inside. We manually link the locations in our dataset with building pages on this site, then scrape the first Google search result constrained within the university domain for each listed department, and use that text to represent the location. In addition to the departments, some pages directly link to a website (e.g., a gym links to recreational sports), from which we also scrape text.

Reddit. With this dataset, we capture how people *informally discuss* locations. From the university Reddit page, we search for building names. We increase the search term list using OpenStreetMap,[2] which lists alternate names for many buildings. We include text from posts and comments that specifically mention a building.

Twitter. With this dataset, we capture how people *express themselves* in various locations. We collect tweets that have been geotagged with GPS points within 0.05 kilometers of campus buildings.

[2]https://www.openstreetmap.org/

4 Representing Locations

We use location trajectories and text data to create vector representations of locations and, subsequently, embeddings of sequences of locations that are visited by a single person. After pre-processing using the method described in Section 3.1, the location input data consists of a series of sorted, non-overlapping locations for a number of users with start and end times. We discuss multiple methods to create vector representations based on this data.

4.1 Location Trajectory-Based Representations

To create embeddings of locations, we make use of the temporal nature of the location trajectories to create a sequence of names of locations visited by a user over a period of time (e.g., the seven month period of our data collection, see Section 3.1). A skip-gram model is trained to use a location to predict locations around it in a user's schedule, creating location embeddings that we expect will encode semantic information about locations.[3]

We represent each hour during the data collection period as a distinct token in the input trajectories. If a user has visited a single location in one hour, that location will be used in the slot for the hour; if they visited multiple locations, their predominant location will be used. If we do not have any location data for the user during that hour, we use the EXTERNAL token. This approach gives an exact meaning to the distance between locations in a sequence, while a raw sequence would ignore gaps in the data. The approach of using one token per set time interval is also used in Zhu et al. (2019). We refer to the method as Loc2V, and show a visualization in Figure 1.

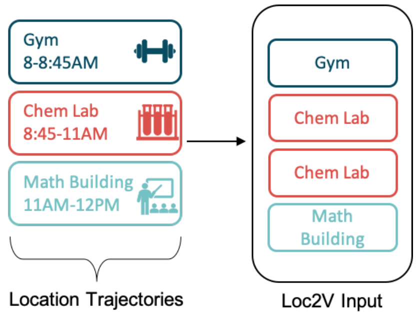

Figure 1: A sample sequence of locations, and the corresponding sequences that are used as Loc2V input.

4.2 Text-Based Representations

In addition to creating location representations from trajectories in the physical world, we explore the idea of using relevant text to define locations. Such text can reveal information about locations that may not be discernible from location trajectories, e.g. that people meet friends in a certain place. Therefore, for the same locations that appear in our trajectories, we collect textual data that enables us to derive text-based representations from three sources as described in Section 3.2.

Using each textual data source, we map a location name to a set of relevant words. We calculate tf.idf (Salton and Buckley, 1988) weights for each word, then use those weights to compute a weighted average of pre-trained word embeddings. Because our datasets are primarily from social media, we use pre-trained GloVe embeddings that were obtained from Twitter data.[4] The resulting vector is used as a location representation.

4.3 Combining Representations

We hypothesize that trajectory based and text-based representations may encode different aspects of locations. Therefore, in addition to representing locations using text and physical trajectories, we experiment with combining the two. Our first method concatenates embedding vectors created from physical trajectories and vectors created from text data. Our second method performs retrofitting on top of text-based vectors. In the context of embeddings, "retrofitting" describes the process of modifying vectors that have already been created to better encode additional criteria. We find inspiration in the method from Faruqui et al. (2015), which retrofits word embeddings to a graph representing a semantic lexicon. In our work, we retrofit text embeddings to the graph that represents the transitions between locations; the nodes are locations, and the edges are weighted by the number of times there was a transition between those two locations in our dataset.

The retrofitting method takes a matrix $\hat{Q}$, the initial vectors, and updates matrix Q (initialized to $\hat{Q}$) using a location transition graph. The objective

[3]We use the default window size of 5 and generate embeddings with 25 dimensions. While 25 dimensions is fairly small in the context of word embeddings, since our dataset has fewer than two hundred locations that we seek to embed, higher values cannot be considered as leading to a dimensionality reduction. We use a negative sampling value of 20, as is suggested by Mikolov et al. (2013) for small datasets.

[4]https://nlp.stanford.edu/projects/glove/

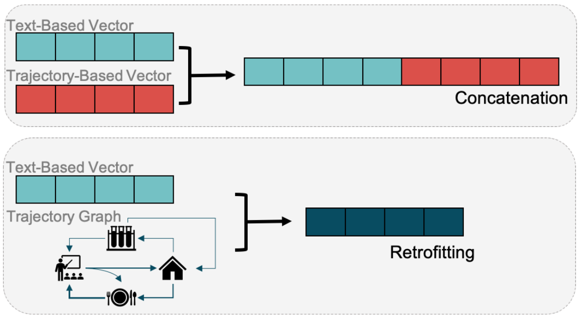

Figure 2: Comparison of the concatenation and retrofitting methods.

function incorporates the set of edges E, bringing vectors that share an edge closer together in the vector space:

$$\Psi(Q) = \sum_{i=1}^{n} \left[\alpha_i \|q_i - \hat{q}_i\|^2 + \sum_{(i,j)\in E} \beta_{ij} \|q_i - q_j\|^2\right]$$

An iterative method is used to update matrix Q:

$$q_i = \frac{\sum_{j:(i,j)\in E} \beta_{ij} q_j + \alpha_i \hat{q}_i}{\sum_{j:(i,j)\in E} \beta_{ij} + \alpha_i}$$

We perform ten iterations, as was done in previous work. The parameters α and β control the relative importance of the two components (initial vectors and location graph). In their implementation, Faruqui et al. set $\alpha_i = 1$ and $\beta_{ij} = degree(i)^{-1}$. As the graph we use is weighted, we introduce a weighted version that incorporates edge weights W, using a weighted inverse degree for β.

The retrofitting method enhances the text-based information by adding the assumption that locations that are visited sequentially are similar (in the sense that a person who visits one would visit the other), bringing them closer in the vector space. This method aims to infuse the text-based representations with information related to the co-occurrence of locations in a student's trajectory; locations that co-occur may be suggestive of, for instance, areas of campus that tend to be visited by engineering students. It is not used on the trajectory-based representations, as these already incorporate location transitions.

Figure 2 compares the concatenation and retrofitting methods. As outlined above, the concatenation method directly combines the two vectors into one with the same content, while the retrofitting method takes information from a graph structure representing trajectories into account to create a modified version of the original vector.

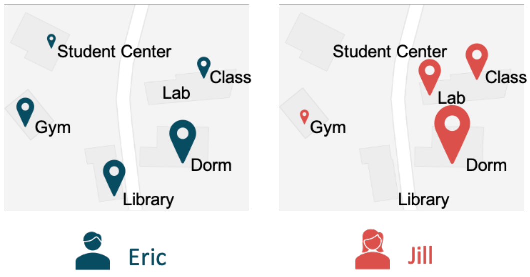

Figure 3: Fictional examples of locations visited by students; a larger pin reflects more time spent at a location.

4.4 Representing a Sequence of Locations

To represent a sequence of locations, we use a vector representing the locations that a person has visited in a month, instead of the individual locations. We settled on this time interval since a shorter time span (such as a day) contains very little predictive information, while a longer span (one semester) groups together distinct time spans that may lead to divergent behaviors, such as exam periods. We create a sequence embedding by taking a weighted average of the location vectors included in the sequence, using the time spent at each location as weights, thus increasing the importance of locations at which the person spent more time.

5 Probing Location Representations

While some of the methods we use (i.e., skip-gram) have been used in the past to represent locations for certain tasks, there has been less work studying them intrinsically. We propose surface level tasks to probe the properties encoded in location embeddings, which are important to gain a deeper understanding of the type of information they capture. We split surface level tasks into two categories: those that focus on individual locations and those that focus on location sequences. In addition to these surface level tasks, we propose a set of downstream prediction tasks to validate the utility of such embeddings.

5.1 Surface Level Location Tasks

With these tasks, we examine two properties that should be encoded in location representations: location functionality and physical proximity. To directly compare how well each method encodes these semantic properties, we propose a metric to measure each property. We are inspired by Ye and Skiena (2019), who use similar methods to analyze properties of name embeddings (representations of people's names). We borrow their method of analysis, measuring overlaps in the N nearest neighbors for various values of N, but they analyze a different property, namely the gender associated with the name.

Functionality Overlap. Each location in our dataset is annotated with its functionality, including two functionalities for mixed-use buildings, e.g., a class building that also contains labs. For each location, we calculate the percentage of its nearest neighbors in the vector space that share at least one functionality; a higher value indicates that the embeddings more distinctly capture functionality. We compute nearest neighbors using cosine similarity.

Physical Distance. We compute the distance in kilometers between a location and its nearest neighbors, and average the distances. This allows us to measure exactly how far a location is from its nearest neighbors; a lower number for this metric correlates with an increased physical proximity.

5.2 Surface Level Sequence Tasks

Our surface level sequence tasks are inspired by the methodology proposed by Conneau et al. (2018) to probe sentence embeddings. Many of those tasks focus on syntax, which is not relevant for our use case, but we adapt their task for location-presence and propose probing for functionality-presence.

Location Presence. We propose a binary location-presence classification task (LocPres). We create classifiers for each location, predicting if the location appears in a sequence. We average the results across all locations with at least one hundred positive and negative examples (resulting in being able to assess 83 locations out of 132).

Functionality Presence. We also propose a functionality-presence task (FuncPres). Given a sequence embedding, we predict if it includes locations of a certain functionality. We use a binary classification setup that mirrors the one used for the location-presence task. We treat the classification of either the primary or secondary functionalities assigned to locations as correct. As with the location-presence task, we average results over all functionalities with at least one hundred training instances from each class (accounting for 11 functionalities out of 13).

5.3 Downstream Application-Based Tasks

In addition to surface level tasks, we want to understand what other human-centric information is encoded in location sequence embeddings. Our hypothesis is that the way in which students spend their time may be indicative of certain information about them; an example of students' diverse behavior on campus is shown in Figure 3. Using the dataset described in Section 3.1, we propose seven classification tasks: five tasks with two classes (major depression, all depression, gender, sleep satisfaction, and GPA), one task with three classes (to predict which school a student is enrolled in, e.g. business or engineering), and one task with four classes (to predict class year).

Sleep satisfaction is reported in a survey (Section 3.1) on a five-point Likert scale; the top three responses are mapped to a positive class, and the bottom two to a negative class. As semester GPA is continuous, we formulate the binary classification as less than or greater than 3.5 (between A- and B+). Depression is measured using the standard PHQ-8 survey; using a clinically validated algorithm (Kroenke et al., 2001), we classify major depression (binary), along with major and other depression (a weaker diagnosis); we label the former as "major depression" and the latter as "all depression." For the other tasks, we filter out underpopulated classes, going from 18 to three classes for school, from five to two for gender, and from five to four for class year. We use a classification approach over regression because we hope that this work can be used to identify at-risk students.

6 Experimental Setup

We perform 10-fold cross validation on 729 instances, where each instance represents a student. Preliminary classification experiments were conducted on a small subset using SVM with linear and RBF kernels, random forests, decision trees, and Naïve Bayes, yet linear SVM had the most robust performance. Accordingly, our experiments consist of classification tasks using linear SVM. As many of the classes are unbalanced, we more heavily weight updates for the minority class(es) by modifying the loss function to use a weight that is inversely proportional to the class's prevalence.

To predict a student attribute, we create one vector for each month of data collection pertaining to each student, using the process described in Section 4.4. Our training framework is illustrated in Figure 4. We start by feeding the sequence vectors through a SVM classifier, which predicts month-level labels. These are then concatenated to form a

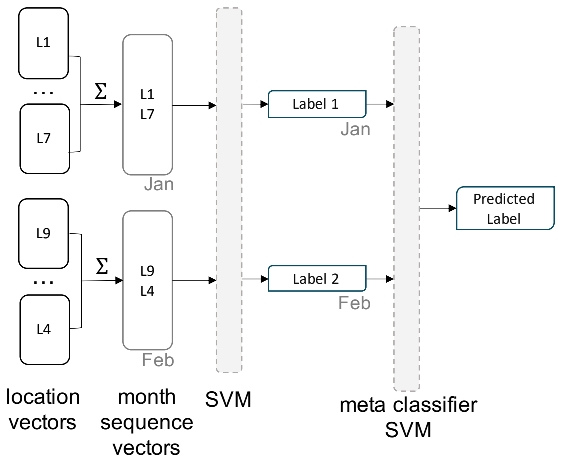

Figure 4: The framework for downstream prediction tasks.

student instance and are passed to a meta-classifier that decides the final class label for that student. We use the meta-classification approach to allow the first classifier more data to learn from; without this approach, the number of input samples is relatively small (729). The process for surface level sequence tasks is similar, but no meta-classifier is used, as the gold standard labels have a month-level granularity.

7 Results and Discussion

Figure 5 and Tables 3 and 5 show the results obtained for the probing tasks. In addition to the loc2vec trajectory and text-based models, we run our experiments with two combination models, using the methods discussed in Section 4.3. We employ the Reddit variation for these combination models due to its strong performance on downstream tasks; we incorporate one model using concatenation and a model using retrofitting. We refer to these models as "Loc2V-Reddit," and "Reddit-Retrofit," respectively.

We compare our classification performance against a random baseline. In order to introduce a stronger supervised baseline for our methods, we employ simpler location representations, in the form of one-hot vectors, which are passed as input in our supervised evaluation framework (Figure 4). We take the mean of those one-hot vectors to create month sequence vectors as we do for the embeddings.

7.1 Surface Level Location Tasks

For these tasks, we include an overall average baseline, where we compute the metric for all locations. The results, shown in Figure 5, lead to two unsurprising findings: text-based methods are better at encoding functionality, and the methods rooted in physical location are better at encoding distance. The results are somewhat skewed for the text-based representations such as "Campus-Website," as some locations share a single page; however, this effect alone does not entirely explain the performance of that model on the functionality overlap task, as it is outperformed on the physical distance task.

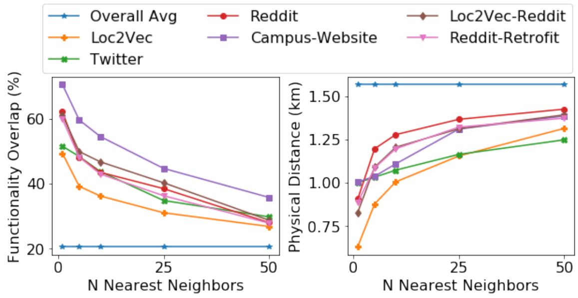

Figure 5: Results on surface level location tasks.

One fascinating result is that the Twitter embeddings offer the best performance on the physical distance task by a method that does not utilize physical trajectories, which may be because this data is collected using geotags. People may tweet as they move between buildings, blurring the line between tweets in adjacent locations. We also observe that the methods that account for physical trajectories and text data can outperform those that use only text data; this is especially clear from the results for Loc2V-Reddit, which show stronger performance than Loc2Vec *and* Reddit individually for functionality overlap, and slightly stronger performance than Reddit for physical distance. This demonstrates one way in which we can create more robust representations of locations.

7.2 Surface Level Sequence Tasks

Overall, we note that all of our methods are easily able to surpass the random baseline. However, when it comes to the supervised one-hot vectorial representation, we see that traditional ways of representing text are able to best encode surface level information. This is because the sparse one-hot representation explicitly encodes information necessary for solving each task; location-presence is denoted by a value greater than one for the particular dimension, and functionality-presence is denoted by a value greater than one for various

	Loc Pres	Func Pres
Random Baseline	41.0	45.0
One-Hot Avg	**61.4**	**62.6**
Loc2V	54.8	55.6
Twitter	56.9	57.8
Reddit	56.9	58.3
Campus-Website	55.8	57.8
Loc2V-Reddit	57.9	59.7
Reddit-Retrofit	55.2	56.5

Table 3: Macro F1 scores (%) on surface level sequence tasks.

Task	# Cls	Inst	% in minority class
Class Year	4	721	22.33
Gender	2	714	49.44
School	3	522	9.77
Sleep	2	729	41.02
GPA	2	729	38.13
All Depression	2	729	18.93
Major Depression	2	729	11.66

Table 4: Class balance for downstream tasks. Instances are reported after filtering small classes.

dimensions.

We find that the text-based methods lead to stronger performance, as compared to their location-trajectory-based counterpart. This confirms that the superior encoding of functionality discussed in Section 7.1 is still discernible with aggregated sequence vectors.

Among all of our proposed methods, the concatenation of trajectory-embeddings and text-based embeddings (Loc2V-Reddit) leads to the strongest results on these tasks. The results on both tasks are completely unmatched by the other methods, indicating that the additional semantic information from concatenation leads to stronger representations.

7.3 Downstream Tasks

We evaluate our embedding methods on the seven downstream tasks introduced in Section 5.3: class year, gender, school enrollment, sleep satisfaction, GPA, all depression, and major depression. These tasks were designed to demonstrate the utility of various location representations in predicting a diverse set of attributes. The overall results for each model are listed in Table 5; we use macro F1 score as our metric. Table 4 shows the size of the minority class for each task. This imbalance and our relatively small data size made it challenging to achieve strong results on some tasks, although we generally were able to improve upon the baselines. Across all the tasks, predicting depression has the most potential for real-world impact, but also showcases the most imbalanced data distribution. With more data, we believe that patterns could be learned in a more robust way.

For the task of school prediction, we greatly improve upon the random baseline even though the data is very imbalanced; this could be because this attribute is clearly linked to where people go on campus, as is class year. For example, freshmen typically live in dorms and eat in dining halls, while seniors often live off campus; computer science students attend classes in different places than English students. The strong performance on the gender prediction task may be explained by the real-world bias entailed in the school of enrollment; e.g., fewer women are enrolled in engineering, so they are less likely to visit engineering buildings. The strong performance on predicting class year with one-hot encodings can be directly linked to the surface level task improvement: freshmen are more likely to visit certain types of locations like dorms (functionality-presence); performance is best among freshmen.

Among text-based methods, we see that the Reddit embeddings enable the best performance on most downstream tasks. Reddit contains the most expressive language compared to the other venues, because its users are able to write at length without a strict character limit or other formalities imposed by media such as Twitter. Furthermore, from manually examining a sample of the posts, the community seems to primarily encompass current and former undergraduate students, therefore establishing a community that is above all else a place for students to share and discuss their daily lives. Meanwhile, the tweets that we link to locations may encompass musings from faculty or visiting scholars, and brief statements that are unrelated to campus life. The campus website data is the furthest from the student experience, as it is devoid of any dynamic content, written in the dry format of informational style. As a result, it seems intuitive that Reddit, in addition to providing *definitional* information about locations (e.g., there are many posts comparing and discussing dormitories), also provides student's *emotional* perspectives on them. We hypothesize that this closeness to student thoughts and feelings is what yields bet-

	Class Year	Gender	School	Sleep	GPA	Depression All	Major
Random Baseline	25.0	50.0	30.0	50.0	49.0	45.0	41.0
One-Hot Avg	**52.1**	56.8	61.8	49.4	51.8	48.2	46.6
Loc2V	50.8	**61.0**	62.0	52.9	51.9	49.6	43.6
Twitter	49.4	57.4	65.4	49.3	51.9	48.5	44.8
Reddit	50.2	59.8	**66.3**	52.7	49.1	50.5	**47.7**
Campus-Website	48.8	58.1	60.1	46.4	51.9	49.4	42.9
Loc2V-Reddit	50.3	59.4	64.5	**53.7**	**52.7**	**50.8**	44.7
Reddit-Retrofit	50.2	60.8	66.0	52.6	47.7	48.7	39.6

Table 5: Macro F1 scores (%) on downstream tasks.

ter performance when predicting student attributes, compared to the other text-based methods.

Overall, while results vary between different tasks, we find that a method that accounts for both physical location trajectories and text data describing locations (Loc2V-Reddit) has a strong overall performance. Notably, it is the best performing model on three tasks and achieves large improvements over the supervised baseline on two additional tasks. Such a model should be considered in future work on location embeddings because of its robustness on varied tasks.

8 Conclusions

In this paper, we addressed the task of building and probing location embeddings. We investigated several strategies to construct them, as well as a suite of probing tasks to understand the type of information encoded within. First, we showed that while all embedding methods encode both physical distance and functionality, methods using trajectories yield better spatial representations and methods using text data better encode location functionality. We showed that, like in the case of sentence embeddings from natural language, sequence embeddings of location data are able to encode surface level information (location-presence, and functionality-presence), as well as information that can be effectively used in downstream tasks. Overall, we found that an embedding model that accounts for both location trajectories and text related to locations (Loc2V-Reddit) gives the best performance over a diverse range of downstream tasks, from prediction of depression or sleep to prediction of academic area of study.

Importantly, we also found that embeddings of locations tend to underperform more traditional one-hot encodings on surface-level tasks, yet they generally outperform these representations on downstream tasks. This suggests that while such embeddings do not explicitly record distinct locations that people visit (thus being more privacy preserving and counteracting negative actions like stalking), they may be more effective for downstream applications that can yield positive outcomes, such as population-level mental health tracking or opt-in tracking for individuals who are in therapy.

Our code is publicly available at `http://lit.eecs.umich.edu/downloads.html`.

Acknowledgment

We are grateful to Shiyu Qu, Jiaxin Ye, and Xinyi Zheng for assisting us with this study. This material is based in part upon work supported by the Precision Health initiative at the University of Michigan, by the Michigan Institute for Data Science (MIDAS), by the National Science Foundation (grant #1815291), and by the John Templeton Foundation (grant #61156). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Precision Health initiative, MIDAS, the National Science Foundation, or John Templeton Foundation.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *International Conference on Learning Representations*.

Buru Chang, Yonggyu Park, Donghyeon Park, Seongsoon Kim, and Jaewoo Kang. 2018. Content-aware hierarchical point-of-interest embedding model for successive poi recommendation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3301–3307. International Joint Conferences on Artificial Intelligence Organization.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Alessandro Crivellari and Euro Beinat. 2019. From motion activity to geo-embeddings: Generating and exploring vector representations of locations, traces and visitors through large-scale mobility data. *ISPRS International Journal of Geo-Information*, 8(3):134.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado. Association for Computational Linguistics.

Shanshan Feng, Gao Cong, Bo An, and Yeow Meng Chee. 2017. POI2Vec: Geographical latent representation for predicting future visitors. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 102–108.

Kurt Kroenke, Robert L Spitzer, and Janet BW Williams. 2001. The PHQ-9: validity of a brief depression severity measure. *Journal of General Internal Medicine*, 16(9):606–613.

Xin Liu, Yong Liu, and Xiaoli Li. 2016. Exploring the context of locations for personalized location recommendations. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 1188–1194. AAAI Press.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas. Association for Computational Linguistics.

Adir Solomon, Ariel Bar, Chen Yanai, Bracha Shapira, and Lior Rokach. 2018. Predict demographic information using word2vec on spatial trajectories. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, UMAP '18, page 331339, New York, NY, USA. Association for Computing Machinery.

Junting Ye and Steven Skiena. 2019. The secret lives of names?: Name embeddings from social media. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 3000–3008, New York, NY, USA. Association for Computing Machinery.

M. Zhu, W. Chen, J. Xia, Y. Ma, Y. Zhang, Y. Luo, Z. Huang, and L. Liu. 2019. Location2vec: A situation-aware representation for visual exploration of urban locations. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3981–3990.