Noise Prediction for Geocoding Queries using Word Geospatial Embedding and Bidirectional LSTM

Tin Vu*
tin.vu@email.ucr.edu
University of California, Riverside
Riverside, California

Solluna Liu
mil@microsoft.com
Microsoft Corporation
Redmond, Washington

Renzhong Wang
rewan@microsoft.com
Microsoft Corporation
Redmond, Washington

Kumarswamy Valegerepura
kumarvp@microsoft.com
Microsoft Corporation
Redmond, Washington

ABSTRACT

User geocoding queries in map applications often contain noisy tokens such as typos in street, city name, wrong postal code, redundant words due to copy-paste action, etc. This issue becomes worse with the rapid growth of mobile devices, where errors from user input are inevitable. Such noisy tokens may fail the searching process if they are passed as-is to the downstream query processing components. In particular, there might be nothing or irrelevant results returned to the user. Therefore, noisy tokens in geocoding queries should be recognized and handled properly prior to the searching process. In this paper, a deep learning based noise prediction model for geocoding queries is proposed. It combines a novel Word Geospatial Embedding (WGE) and a Bidirectional LSTM based sequence tagging model. The proposed WGE is the first language model that allows geospatial semantics to be encoded into the vector representations. It allows geo-related machine learning/deep learning models making spatial-aware prediction.

CCS CONCEPTS

• **Information systems → Document representation**; • **Computing methodologies → Machine learning approaches**.

KEYWORDS

geocoding, word embeddings, deep learning

ACM Reference Format:
Tin Vu, Solluna Liu, Renzhong Wang, and Kumarswamy Valegerepura. 2020. Noise Prediction for Geocoding Queries using Word Geospatial Embedding and Bidirectional LSTM. In *28th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '20), November 3–6, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397536.3422201

1 INTRODUCTION

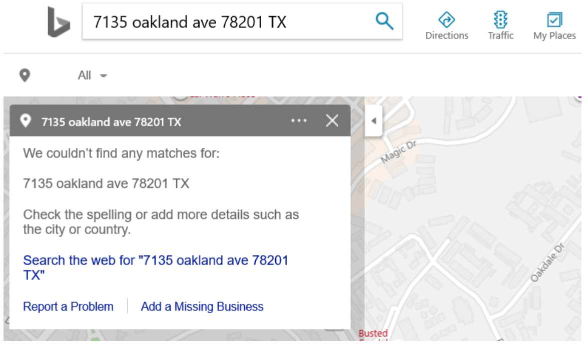
Geocoding is a process which transforms the description of a location to the coordinates of that location on Earth's surface. This is a difficult task which research community has been attacking for a long time. One of the critical problems of geocoding is the noisy tokens inside the queries. In particular, the success of geocoding highly depends on the quality of geocoding tokens. If the address tokens are incomplete, redundant or containing mistakes, it would be very hard or impossible to find the expected location from the input address. Some examples of noisy tokens in geocoding queries are typos in street name or city name, redundant words inside an address due to copy-paste action, incorrect postal code, to name a few. Figure 1 shows an example when a Bing Maps user cannot find the expected result due the a typo in postal code in the address query. A simple analysis on Bing Maps user search logs indicates that there are about 20% of geocoding queries contain noisy tokens. In this paper, we use geocoding queries and address queries interchangeably.

If a query contains noisy tokens, the results from the geocoder might be not reliable or the geocoder even cannot find a result. Therefore, it is very important to recognize the noisy tokens in geocoding queries. Once geocoder could identify these noisy tokens, it can do further actions such as removing them from the input or notifying to user about the noises. This will improve user experience instead of giving them nothing or an unexpected result. In general, given a geocoding query which contains several tokens, we aim to design a classification model so that it can predict whether a token inside the input query is a noisy token or not.

First, based on the query analytics, we found that many noisy tokens in geocoding queries might not hold a spatial relationship with other useful tokens. Therefore, we come up with an idea that if we can take the spatial semantics of address tokens into account in our prediction model, we might have better insights to identify whether a token is a noise or not. In detail, we proposed a new embedding technique, namely Word Geospatial Embedding (WGE), which transforms address tokens into numerical vectors. A vector representation of a word should carry its spatial information. The main purpose of WGE is that if two words has a relationship in terms of spatial semantics, their vector representations should also reflect this relationship. We addressed this problem by proposing

*Work performed while at Microsoft

SIGSPATIAL '20, November 3–6, 2020, Seattle, WA, USA
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8019-5/20/11.
https://doi.org/10.1145/3397536.3422201



7135 oakland ave 78201 TX

7135 oakland ave 78201 TX

We couldn't find any matches for:

7135 oakland ave 78201 TX

Check the spelling or add more details such as the city or country.

Search the web for "7135 oakland ave 78201 TX"

Figure 1: How the noisy tokens badly affect user experience

a new word embedding technique which is tailored for geospatial word space.

Second, this noise prediction problem is similar to a word classification problem with two labels: noisy or useful token. Thus, we could utilize the current NLP approaches to solve this problem. In addition, the recent success of deep learning in NLP problems motivated us to build a noise prediction model using deep learning. We used a Bidirectional LSTM [4] model as the main component of our prediction model. The reason is that an address query should be examined in both directions: forward and backward in order to have a reliable conclusion about the labels of its tokens.

In summary, we proposed a noise prediction model for geocoding queries using deep learning based techniques. In this paper, the proposed system focuses on address queries but it can also be applied to other geocoding queries such as business name, point of interest, etc. In addition, we enhance the representation of address token by introducing a novel embedding technique, called WGE. The experimental results validated that our approach can efficiently predict the noisy token in address queries with a very high accuracy.

2 RELATED WORKS

Word Embedding Techniques: In summary, word embedding techniques could be classified into two categories: prediction based and co-occurrence based. The prediction based methods (e.g. word2vec) take the bi-gram relationships of words in in vectorization process. They are useful to represent word semantics. However, most of the words in address word space carry spatial meaning rather than word semantics. For example, a street name could be a name of a person, a number, or a word without any meaning. This characteristic limits the power of prediction based word embedding techniques. Thus, the co-occurrence based methods such as GloVe might be more suitable to build a word embedding component for spatial textual data. Moreover, a general corpus might not be able to reflect well the spatial relationship between query tokens. Therefore, we propose a new embedding method, which takes the spatial semantic of words into account. For instance, if two words are spatially close to each other, their embedded vectors should be also close in terms of Euclid distance.

Address Parsing: is a class of algorithms and softwares that classify a token in a geocoding query into several labels such as house number, street name, street type, city, state, postal code, etc. In other words, an address parser turns unstructured address data into neat and tidy columns of address fields. In general, address parser could be considered as a specific case of Part of Speech tagging problem [9]. An example of address parser is *libpostal*, which is an open source statistical NLP using open geographic datasets. *libpostal* utilizes the Conditional Random Fields [10] algorithm to train its address parser from billions address from Open Street Map, OpenAddresses and GeoPlanet postal codes. Address parser is applied in geocoding system of several map services such as Bing Maps or Google Maps. In a geocoding system, address parser enhance the performance of query processing since it suggests reasonable search strategies for a given address query. However, a normal address parser is not able to identify the geospatially noisy tokens in a query. The reason is that an address parser usually is unaware of spatial context of the input tokens. For example, if an address query contains a wrong street name, address parser probably still classifies this is a street and passes this token as-is to the geocoder. As the result, the geocoder still cannot resolve the query. In this paper, the proposed noise prediction model aims to resolve this challenge so that it can help the geocoding process to produce a more reasonable result.

3 PROBLEM DEFINITION

This section formally defines the noise prediction problem and the problem scope that we are trying to solve. In this paper, we define the query's tokens that cause the failure of query processing in geocoder are noises. A noise could be a typos from user's query, or some information that our address database does not store such as suite, apartment number or business name. We formulated noise and noise prediction problem as following.

Definition 3.1 (Noise token). A token in a geocoding query is considered as a noise if it fails the address entity retrieval process in a geocoding system.

Noise prediction problem: *Given a query which is represented by an array of n tokens* $Q = [q_1, q_2, ..., q_n]$*, determine a noise prediction vector* $y = [y_1, y_2, ..., y_n]$ *such that* $y_i = 1$ *if* q_i *is a noise, otherwise* $y_i = 0$.

Overall, the problem can be formulated as a sequence tagging problem as whether a token is a noise not only depending whether it exists in the indexed data of the geocoder but also on the role it plays in a geocoding query or its relationship with other tokens in the query. While sequence tagging in natural language processing learns semantic and syntax of natural language sentences, geocoding queries have additional geospatial information to exploit. As the result, the existing solutions for sequence tagging are not applicable for this noise prediction problem. In the next section, a novel system that utilize spatial information to solve noise prediction is presented.

In sequence tagging models, the simple per-token accuracy is the most intuitive metric to evaluate the performance of a prediction model. In this paper, we also focus on optimizing our proposed model based on this metric. However, in some cases, the full query accuracy is important as well. For example, given an address query *"570 W Blaine st, Riverside,* ***CAA*** *92507"*. If the noise prediction model incorrectly identifies *CAA* is not a noise, a full-text search engine like ElasticSearch might still be able to find the correct result.

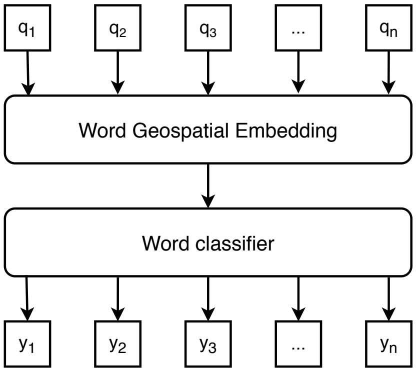


Figure 2: Noise prediction system architecture

However, an inverted index with simple lookup algorithms would probably not return a correct result for this query. In short, we should increase the per-token accuracy as high as possible in order to increase the full query accuracy as well.

4 NOISE PREDICTION MODEL FOR GEOCODING QUERIES

In this section, we describe the complete design of our noise prediction system for geocoding queries. Figure 2 shows the architecture of the proposed system. Suppose that the input query is $Q = [q_1, q_2, ..., q_n]$, the final output of our system is the prediction vector $y = [y_1, y_2, ..., y_n]$ as we described in Section 3. In order to fulfill this goal, our system includes the following components:

Key component 1 - Word Geospatial Embedding (WGE): In general, most of machine learning and deep learning algorithms only works with numerical inputs. Hence the textual input need to be converted to a list of numerical vectors before any other processing components. Word embedding is the most popular approach for this task, which map words or phrases to vectors of real numbers with linguistic contexts of words. Over the years, the quite a lot word embedding approaches were proposed, from statistic based SVD to neural net work models such as the popular word2vec[6] and GloVe[7], to context-aware approaches as ELMo [8] and BERT [2], etc. However, these techniques are not applicable to build language models for textual geocoding queries. Because geocoding queries have unique features, i.e., each useful word in a query can a associated with some geospatial locations. e.g. a couple of tiles. Therefore, a novel word geospatial embedding based on GloVe is proposed here. Glove is chosen because token's geospatial co-occurrence is more relevant for noise prediction task.

Key component 2 - word classifier: As we discussed in the Section 3, the noise prediction problem is formulated as sequence tagging problem. Therefore, we can utilize the existing works which have been trying to attack this problem in a long time such as Hidden Markov Model [1], Conditional Random Field [10], Structured Perceptron/SVM [5], or deep learning based approaches in recent years. The proposed model in Figure 2 has a flexible word classifier component then it could use any mentioned techniques. Based on the surprised efficiency of deep learning based approaches to solve NLP problem, the Bidirectioanal LSTM [4] architecture is adopted in the proposed noise prediction model due to its accuracy and low memory usage.

ID	Datasets	# of records	Noise token ratio
D1	Well-formatted queries	56k	20.1%
D2	User log queries	39k	22.4%
D3	Yelp queries	12k	20%

Table 1: List of address queries datasets

5 PRELIMINARY RESULTS

In this experiments section, we show some preliminary results to validate the advantages of the proposed noise prediction model. First, we evaluate performance of the model in terms of token-level. Second, we evaluate the query-level performance of the proposed model. The baseline techniques include a naive classifier, libpostal address parser and word2vec based sequence tagging model.

5.1 Experimental set up

Data sources for Address Corpus: To build the address corpus for the proposed word geospatial embedding, we use the addresses which are collected from Open Street Map [3] data. In the future, we can also add other data sources such as OpenAddresses or in-house address data.

Address queries: Table 1 shows the list of address queries datasets we are going to use in our experiments. (1) The first dataset is synthesized from a list of well-formatted queries. These queries are extracted from an in-house address database which is used for Microsoft Bing Maps. All the queries contain a complete address fields such as house number, street name, city name, state abbreviation, postal code in sequential order. Due to its well-formatted form, this dataset could be considered as the baseline dataset that promises a high prediction accuracy for all prediction models. (2) The second dataset is a collection of addresses queries which is extracted from Microsoft Bing Maps user query logs. We only collect the user queries which succesfully got the expected answer from the geocoder. Since all the queries are unstructured textual data, they might introduce more difficulties for prediction models. (3) The third dataset is the user queries from public Yelp dataset. Since these queries are publicly available, people can reproduce the results of this paper using this dataset. Since all the queries does not contain any noisy token at the beginning, we make these queries being invalid by injecting new noise token to these queries by a synthetic algorithm. We also show the proportion of noise tokens in these queries in Table 1.

Metrics: We define 2 important metrics that reflect the efficiency of our proposed noise prediction model as follows:

- **Token-level accuracy:** measure how good the model is when it predicts whether or not an input token is a noise.
- **Query-level accuracy:** Given an input query with a sequence of tokens. We consider a query prediction is correct if and only if all of its tokens are predicted correctly. Query-level accuracy is computed as total correct query predictions over the total number of queries.

No.	Dataset ID	Naive	libpostal	word2vec	**WGE**
1	D1	79.62%	87.16%	95.70%	**98.17%**
2	D2	79.69%	86.18%	94.17%	**97.43%**
3	D3	80.81%	87.10%	96.68%	**98.94%**

Table 2: Comparison with baseline techniques in terms of token-level accuracy

No.	Dataset ID	Naive	libpostal	word2vec	**WGE**
1	D1	24.68%	26.75%	76.77%	**88.8%**
2	D2	26.69%	28.42%	70.45%	**85.27%**
3	D3	24.78%	25.2%	81.77%	**93.7%**

Table 3: Comparison with baseline techniques in terms of query-level accuracy

5.2 Model evaluation in token-level

In this experiment, we will evaluate the efficiency of the proposed noise prediction model in different datasets. We train and test our model with the number of units per LSTM model as 64, vector size of 100 dimensions and windows size of 10 tokens. In order to highlight the advantages of the proposed model, we compare its performance with several approaches as the followings:

- **Naive algorithm:** This algorithm simply considers a query token as a noise if that token does not appear in the address corpus.
- **libpostal:** This is a state-of-the-art address parser using statistical NLP and open data. We use libpostal to label input tokens and classify them as noise or non-noise based on our definition of noise types.
- **Pretrained word2vec:** To highlight that the existing word embedding techniques do not have a spatial-awereness, we also evaluate the performance of a model that uses a pretrained word2vec with the same Bidirectional LSTM layer in the proposed model.

Table 2 shows the token-level accuracy of different models in the test datasets as described in Section 5.1. It turned out that the proposed noise prediction model are doing very well in all datasets. In particular, the token-level accuracy the the proposed model could achieve is as high as 98.94% for Yelp address queries. This validates that the proposed noise prediction model successfully integrated spatial semantics of words to identify noisy tokens. We can also observe an upward trend of model accuracy in the order of algorithm's complexity. First, the *naive algorithm* just simply considers the token which is not contained in the address corpus as noise, which might miss many other type of noises. For example, there could be a wrong postal code of address but the naive would not consider it as a noise if that postal code appears in the address corpus. The *libpostal* accuracy are significantly improved when compared to the naive algorithm. However, since it is designed as an address parser, it still cannot detect many noise type such as a wrong street or city name, due to the lack of spatial-awareness. The next baseline model we compare is the *pretrained word2vec* from GoogleNews data. This word2vec model outperformed the prediction accuracy of naive and libpostal model, since it takes the context of words into account. However, since pretrained word2vec does not organize the address corpus in a way the spatial relationship of words are reflected in the embeddings, it is still not good as the proposed noise prediction with word geospatial embedding.

5.3 Model evaluation in query-level

Notice that even the proposed model is only few percents better than word2vec based model in terms of token-level accuracy, there will be larger gaps in other metrics such as precision, recall and false positive rate. This will make a big difference in terms of query-level accuracy as shown in Table 3, since one query prediction is only correct if all of its tokens are predicted correctly. This is an important metric for simple geocoders, where even a single noisy token can affect the entire information retrieval process. The high query-level accuracy of the proposed noise prediction model indicates that it is a potential candidate for existing geocoding systems if they want to improve their query processing performance and user experience.

6 CONCLUSIONS AND FUTURE WORKS

In this paper, we proposed a deep learning based noise prediction model for geocoding queries, which use a novel Word Geospatial Embedding (WGE) technique. Based on its spatial-awareness, WGE might also be very useful for many other geo-related machine learning problems. The experimental results showed that the proposed model could achieve a very high accuracy in terms of both token-level and query-level metric. The proposed model could be integrated into existing geocoding systems with least effort. In the future, we will try to build the model that support multiple languages, which is a feasible demand for the geocoding queries.

ACKNOWLEDGMENTS

This work is done under the support of Microsoft Corporation.

REFERENCES

[1] Matthew J Beal, Zoubin Ghahramani, and Carl E Rasmussen. 2002. The infinite hidden Markov model. In *Advances in neural information processing systems*. 577–584.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[3] Mordechai Haklay and Patrick Weber. 2008. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing* 7, 4 (2008), 12–18.

[4] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).

[5] Ryan McDonald, Keith Hall, and Gideon Mann. 2010. Distributed training strategies for the structured perceptron. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*. 456–464.

[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[7] 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)*. 1532–1543.

[8] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* (2018).

[9] Helmut Schmid. 1994. Part-of-speech tagging with neural networks. In *Proceedings of the 15th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 172–176.

[10] Charles Sutton, Andrew McCallum, et al. 2012. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning* 4, 4 (2012), 267–373.