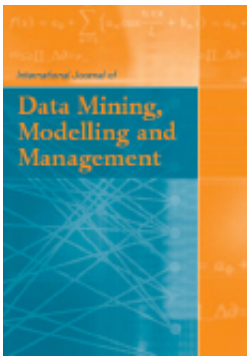


International Journal of Data Mining, Modelling and Management

ISSN online: 1759-1171 - ISSN print: 1759-1163
https://www.inderscience.com/ijdmmm

Adaptable address parser with active learning

You-Xuan Lin

DOI: 10.1504/IJDMMM.2023.10051856

Article History:

Received:	18 November 2021
Last revised:	07 January 2022
Accepted:	28 January 2022
Published online:	04 April 2023

Adaptable address parser with active learning

You-Xuan Lin

National Center for Research on Earthquake Engineering,
No. 200, Sec. 3, Xinhai Rd., Da'an Dist.,
Taipei City 106219, Taiwan
Email: yxlin@narlabs.org.tw

Abstract: Address parsing, decomposing address strings to semantically meaningful components, is a measure to convert unstructured or semi-structured address data to structured one. Flexibility and variability in real-world address formats make parser development a non-trivial task. Even after all the time and effort dedicated to obtaining a capable parser, updating or even re-training is required for out-of-domain data and extra costs will be incurred. To minimise the cost of model building and updating, this study experiments with active learning for model training and adaptation. Models composed of character-level embedding and recurrent neural networks are trained to parse address in Taiwan. Results show that by active learning, 420 additional instances to the training data are sufficient for a model to adapt itself to unfamiliar data while its competence in the original domain is retained. This suggests that active learning is helpful for model adaptation when data labelling is expensive and restricted.

Keywords: address parsing; record linkage; active learning; model adaptation; recurrent neural network; RNN; address in Taiwan.

Reference to this paper should be made as follows: Lin, Y-X. (2023) 'Adaptable address parser with active learning', *Int. J. Data Mining, Modelling and Management*, Vol. 15, No. 1, pp.79–101.

Biographical notes: You-Xuan Lin is currently working at the National Center for Research on Earthquake Engineering as a Research Assistant. Her research interests include natural language processing for real-world applications, discrete event simulations, seismic loss estimation of emergency departments, and seismic disaster preparedness and response measures formulation.

1 Introduction

Address parsing is a process of decomposing an address string to semantically meaningful components. Commonly, the components correspond to administrative divisions, place names, spatial features, and numbering systems, depending on the address system of interest. A wide range of fields and industries have recognised the advantage of applying address parsing. In the industry of postal and delivery services, the potential of address parsing for cost reduction is notable (Sharma et al., 2018). Abid et al. (2018) combined computer vision and a postal address parser to automatically recognise and annotate postal addresses on parcels. Record linkage, a task mapping records in disparate sources, is another prominent motivation that propels the development of

address parsing (Churches et al., 2002). Since address is basic information appearing in documents, records, and datasets, joining data in different sources on shared addresses helps make the best use of information, yielding added value to further applications and practices. A specific example is geocoding. Geocoding is a task of transforming textual addresses to coordinates. It relies on parsing addresses to representations consistent in formats with address indices in a geospatial database from which coordinates of the matched index can be retrieved (Mokhtari et al., 2019). This study was inspired by a task of record linkage and geocoding, particularly linking House Tax data of Taiwan to a geospatial database to obtain a building database of Taipei City in Taiwan. Due to the use of abbreviations and synonyms, flexibility in address structures, inconsistencies in individuals' habitual use, and even simply mistakes, one identical location may be indexed by addresses in various literal representations (Wang et al., 2016). Parsing addresses and standardising the forms can facilitate entity matching (Küçük Matci and Avdan, 2018).

Despite all the benefits address parsing brings, developing an address parser is not an easy task. Traditionally, address is parsed using hand-crafted rules. However, real-world address data have a range of forms, and flexibility in the address components as well as structure adds more complexity to the task. Formulating rules that exhaustively attend to all features is labour-demanding. The whole system may end up being stuffed with a bulk of rules, making it too complex to be maintained and updated. New approaches based on machine learning (ML) algorithms ease human workloads in identifying features for classification and render a certain extent of generalisation. Still, diversity and flexibility in the address forms and structure impose a challenge on the preparation of labelled data for ML. Collecting a large enough dataset with labels properly annotated is expensive. Limited access to labelled data either impedes training of parsers at the first place or restricts applications of the developed parser.

One promising solution is active learning, a training scheme operating counter to conventional passive learning. Active learning works based on an assumption that allowing a developing model to select instances informative for its own learning helps improve effectiveness and efficiency. To put it simply, active learning transforms a learner (i.e., a developing model) from 'passively' receiving labelled training data in bulk to 'actively' selecting instances to be labelled by the 'oracles' (human annotators) (Settles, 2010). The learning process starts with training an initial model on a small labelled dataset. The trained model then scans through all unlabelled data and selects instances to be labelled based on a predefined query selection strategy (e.g., selecting instances that the current model is least certain about). Afterwards, the newly labelled data will be added to the training set, and another round of training, querying, and labelling will continue till the termination criterion is fulfilled. This iterative learning scheme makes the whole training process more economical with less expense in data preparation in bulk while, at the same time, comparable or even improved performance can be achieved as compared to passive learning (Gal et al., 2017; Shen et al., 2017). This training procedure is especially useful when unlabelled data is cheap and abundant, but labelled data is scarce and expensive.

In this study, we compared the effects of passive and active learning procedures on training a competent and general address parser. We constructed a neural network (NN) based model using the bidirectional recurrent neural network (RNN) architecture. RNN is a specialised NN designed for sequential data where each token is dependent on its preceding and/or following tokens. Regarding active learning, we adopted uncertainty

sampling by which instances the model is most uncertain about are selected. Considering costs and effects, we modified the typical pool-based approach where only one instance is selected each time. Instead, by our design, five least confident instances are selected in each iteration. Finally, five models constructed with the same model architecture were obtained following either passive or active learning procedures with different training data. Comparison of model performance was made to figure out two main research questions:

1 Is active learning an economical method to train an address parser?

2 Does active learning help improve model generalisation when a model is to be updated for new data?

Five models were trained for experiments to answer the research questions, using addresses of Taipei City as in-domain data and New Taipei City as out-of-domain data. First of all, we examined whether the typical passive learning that requires a large amount of labelled data is advantageous over the active learning where a remarkably small set of data is labelled for training. Second, we tested the effectiveness of passive learning and active learning when an equal amount of training data was given. We further tested generalisation ability of all models on labelling a new dataset composed of out-of-domain data. Finally, we experimented with a hypothesis that greater improvements can be made in the model's performance on the out-of-domain data after active learning is applied again with a selection of new instances from New Taipei City, as compared to a model trained with random instances of the same size.

The learning curve of active learning suggests that the modified active learning procedure is effective in timely self-adjustment, preventing the model deviating much from the optimal one. However, if the initial model for active learning has achieved high accuracy in the first place, active learning may risk leading to a weaker model at the end due to the unstable learning curve as more and more noisy instances are selected and added to the training data along the iteration. Despite the pitfall, active learning was found particularly effective in tuning a trained model for domain adaptation and, at the same time, retaining or even improving its competence in the old data.

The main contributions of this study include:

1 an example methodology of building an effective parser of Chinese address

2 a demonstration of applying active learning to adapt an existing model at lower cost with less data labelling required.

2 Models of address parsers

The techniques of address parsing have long been studied using various methods. There are roughly two major types, the rule-based approach and ML based approach, which can be further divided into two types, probability models and NN models. In the following, models in literature will be introduced with their pros and cons elaborated. A summary is provided in Table 1.

Rule-based approaches are relatively accessible as compared to ML based approaches. As the name suggests, it leverages hard coded declarative rules derived either from existing regulations or from domain knowledge. Advocates of this approach in

information extraction, of which address parsing can be viewed as a sub-type, endorse its interpretability, ease of maintenance, and error traceability and fixability (Chiticariu et al., 2013).

Table 1 Comparison of address parsing models

Methodology	*Pros*	*Cons*
Rule-based approach	Interpretability; tractability and fixability of errors	Relying on expert knowledge; limited generalisation; complex rule programming
Probability model	Requiring less training data; exploitation of established knowledge (e.g., probabilities of the observed states)	Limited by assumptions (e.g., independence assumption in HMM)
Neural network	State-of-the-art approach; free of assumptions; stronger generalisation; requiring less domain knowledge	Black box; requiring a large amount of labelled training data

The rule-based approach, however, is considered weak in generalisation and requires domain knowledge as well as skillful rule-programming (Abid et al., 2018; Churches et al., 2002). Alternatively, the ML based approach is advantageous in generalising and is more tolerant of variability in input data. Churches et al. (2002) used hidden Markov model (HMM), a probabilistic machine that considers the transition probability of the hidden states (or labels in address parsing) and the emission probability of the hidden states to the observed states (tokens to be labelled), to obtain a chain of labels that maximises the joint probability with the observed sequence. The HMM approach is found on par with rule-based approach.

Conditional random field (CRF) is another competent probabilistic model in sequence labelling, especially in named entity recognition (Waszczuk et al., 2013) and automatic speech recognition (Palaz et al., 2013). Similar to HMM, CRF takes the neighbouring context into account while the objective of CRF is to maximise the conditional probability of a label given an input token. Before the NN-based approach prevail sequence labelling tasks, CRF was the most popular option. Several studies have shown that CRF is more robust than HMM (Comber and Arribas-Bel, 2019; Wang et al., 2016). The advantage of CRF over HMM is that expert knowledge can be exerted in extracting relevant features that facilitate model performance while the latter relies simply on the surface sequential order and the relation between labels and tokens.

NN based approaches are state-of-the-art in sequence labelling and information extraction. For the input, some researchers continue to depend on human knowledge and judgements about features that provide critical information for the model to learn and accomplish a task (Sharma et al., 2018). Since feature extraction is computationally expensive, most studies turn to word representations, which is to convert categorical data (words, characters, or other units depending on the data granularity used) to distributed representations (or vectors) by a pre-trained layer, mostly word2vec (Mikolov et al., 2013) or BERT (Devlin et al., 2019). As for the design of the classifier/parser, long short-term memory (LSTM) (Abid et al., 2018; Craig et al., 2019) and sequence-to-sequence (seq2seq) (Yassine et al., 2020) are popular model architecture. Mokhtari et al. (2019) compared different NN architecture on tagging address queries in map search. Results showed that the performance of the seq2seq surpasses LSTM. Still some other researchers took the probability model and NN model together and made the most of the

advantages of the two (Kong et al., 2016; Lample et al., 2016). Though LSTM and seq2seq are frequently used architecture, in this study we adopted vanilla RNN, the basic architecture designed for sequence data, to build the parser. The basic model architecture requires computational resources less, aligning with our goal to develop an address parser in an efficient and economical way. Results will show that the vanilla RNN architecture is robust enough for our data and training objectives.

ML approaches have made a cornerstone in address parsing, but studies in literature face common problems associated with training data scarcity. To train an ML model that solves real-world problems is to expose the model to a training dataset that represents authentic situations. Since readily available labelled datasets are usually monotonous and restricted in scope, to obtain a dataset representative enough for training, and, at the same time, to contain the costs of labelling, many researchers opted for synthetic data. That is, data is artificially generated following observed patterns to approximate various forms, such as missing values, misspellings, and multiple values (Abid et al., 2018; Lin et al., 2020; Mokhtari et al., 2019). Despite adding diversity to the training data, synthetically generated data is still relatively less complex and lacks variability. A trained model may end up detecting the human-made patterns for data synthesis and learning to label addresses by those patterns instead of by the underlying patterns that we expect it to acquire. Moreover, address systems may vary from place to place. Even for addresses of the same language in the same country, address formats differ because of administrative hierarchy, city planning or urban-rural gaps, and local conventions or colloquial usage. Previous studies pertaining to address processing admitted the limitations of their models when being applied to addresses outside of the realm of their training data (Lin et al., 2020; Xu et al., 2020).

To exploit the powerful ML algorithms in address parsing, the problem pertaining to the scarcity of training data should be addressed to ensure the capability and generalisation of the trained model. In this paper, we will show how active learning can serve as an effective solution.

3 Features of address

Address in Taiwan is the target of this study. In this section, features of address in Taiwan and some atypical features that characterise addresses in House Tax data will be elaborated. Methods catering to those features will also be stated.

3.1 General features of address systems and address in Taiwan

Features of an address system are mainly attributed to:

1 intrinsic features of the language

2 the standard address structure.

To start with, unlike English where a space sits between words, there is no natural word separator in Chinese. A Chinese address is a consecutive sequence of Chinese characters together with digits, symbols, or alphabets. If the granularity of input is set to be word, data pre-processing for word segmentation is required with word segmentation tools, like CkipTagger (Li and Ma, 2019) and Jieba (Sun, 2020). Figure 1 displays an example of

address in Taiwan and the form after being segmented into words (with English translation of each segment provided below). However, it should be noted that segmentation tools are usually developed based on corpora of general language use; highly specialised language for location indexing seems on the margin of its usage domain (Lin, 2021). Previous researchers have noticed that the quality of segmentation imposes a remarkable impact on the model's learning outcome (Chang et al., 2016). One solution catering to the potential segmentation failure is to add gazetteers to the dictionary (Lin et al., 2020) to encourage or force words listed in the dictionary to be returned. The problem is that a gazetteer with an exhaustive list of place names is not always available.

Figure 1 An example address in Taiwan and word chunks (see online version for colours)

Original	臺北市大安區辛亥路三段200號				
Segmented by word	臺北市	大安區	辛亥路	三段	200號
	Taipei City	*Da'an Dist.*	*Xinhai Rd.*	*Sec. 3*	*No. 200*

To evade the potential noises caused by segmentation errors, in this study, the level of granularity was set to be character. While word is commonly agreed to be the fundamental semantic units, in Chinese address, a semantically meaningful chunk is mostly marked by suffixes at the character level (e.g., the character 路 for the road name, and 號 for the house number). As long as a model can learn the significance of these suffix characters, we assume that it is able to identify the semantic meanings of chunks separated by suffixes and assign corresponding labels.

Another key factor that dominates addresses is the structure that determines constituents and their orders. In Chinese, address items are ordered from general items to specific items, whereas in English, the order is reversed, i.e., from specific to general. Therefore, a successful address parser is supposed to be able to recognise address structures. The importance of knowledge acquisition about address structures is also observed by Yassine et al. (2020) who proposed a model for multinational address parsing and found that this model could yield acceptable performance on addresses in an unfamiliar language with its structure similar to that of the training data; however, if the condition was reversed, in familiar languages but with alien structures, the performance dropped sharply. According to the level of rigor and the use of spatial relations for location references, address systems can be categorised to three types, structured, semi-structured, and unstructured. The degrees of difficulty for automatic address parsing increase as the address type moves toward the unstructured end since more relevant background knowledge would be required (Javidaneh et al., 2020).

The target of this study inclines toward the semi-structured type since flexibility holds to some extent. Table 2 summarises the address components and structure in Taiwan. The field names are arranged from top to bottom following their standard sequence in address. Sub-components are subordinate to the main field as further division, if any. Some components may be absent in an address for different reasons. The first reason is that a division may not be applicable. For example, for some roads not further divided into sections, section would then be absent in the field of Road. Secondly, some address fields can be omitted without causing ambiguity or a failure of identification. Tract and Neighbourhood are two fields frequently omitted.

Table 2 Address components and examples

Field name	*Sub-components*	*Suffixes/indicators*	*Example*
City		縣, 市	臺北市
Town		市, 鄉, 鎮, 區	大安區
Tract		村, 里	學府里
Neighbourhood		鄰	5鄰
Road		路, 街, 大道	辛亥路
	Section	段	辛亥路一段
Lane		巷	15巷
Alley		弄	15弄
House number		號	58號
	Sub-number	之, -	58-1號
	Qualifier	臨, 特	臨58號
Ground floor		樓	5樓
	Unit	之, -	5樓之1
Basement		地下	地下室
	Floor	樓, 層	地下5層
	Unit	之, -	地下5樓之1

Aside from the above-mentioned flexibility in address fields, values of the fields are allowed to deviate from the standardised form as long as comprehensibility is maintained. Elements reaching to the tail of an address are especially less restricted. particularly, the house number and floor (ground floor/basement) have values in quite diverse patterns. Indicators of ground floor and basement are not limited to those presented in Table 2. To name a few, for number, *58-1號* (No. 58-1) is equivalent to *58號之*1 as well as *58之1號*, for ground floor, *5樓* (5th floor) is equivalent to *5層* as well as *5層樓*. Additionally, abbreviations (e.g., *北市* for *台北市*) and interchangeable characters (e.g., *台* and *臺*; and digits and Chinese numbers) further contribute to variability in address components. These features together increase complexity of address parsing.

Due to the high variability, the training data size is supposed to be large so that instances of various patterns can be included as much as possible. However, labelling a tremendous dataset is expensive and time consuming. Active learning is supposed to be a feasible solution to this challenge. Query selection algorithms of active learning ensure that the selected and labelled instances are informative for model development so that the invested time and labour are not wasted on types of instances that the model has already learned how to label.

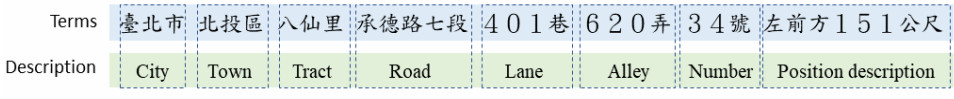
3.2 The data source and features of the data

The address data used in this study were extracted from the location description of tax objects recorded in the House Tax data of Taiwan. The House Tax data record every single tax object, particularly buildings/constructions, with attributes documented, e.g.,

floor area, the structure type, and the location description. The location description is primarily structured address for houses that have a formal house plate. However, since tax objects are constructions of various types, other than a house, such as an apartment, a parking lot, and other infrastructures, for constructions with no house plate or more than one house plates, the field of location description contains unstructured description that is prone to ambiguation (Abrol et al., 2013). Data of this type have the following features making them distinct from common postal addresses:

1 For places that have no house plate or have not yet registered for one at the time when taxation initiated, their locations may be described by their positions relative to nearby reference points, which may be a house with a formal address, road intersections, or landmarks. Figure 2 exemplifies an address entry containing position descriptions that tell the intended object is located on the left front 151 meters from the reference address. In other cases, supplementary information may be added to specify the objects, e.g., the community name or the building name.

Figure 2 Example of an address with position descriptions (see online version for colours)



2 A tax object may be a building that is shared by several households each of which has their own house plates. Therefore, the location description may be a string with multiple addresses concatenated. Figure 3 is an example of one single address entry with multiple house numbers concatenated. An entry may also contain two road names and multiple house numbers (see Figure 4). Entries of this type denote one single building located at the intersection of two roads; therefore, households in the same building facing different roads have different road names in their addresses.

Figure 3 Example of an address entry with multiple numbers (see online version for colours)

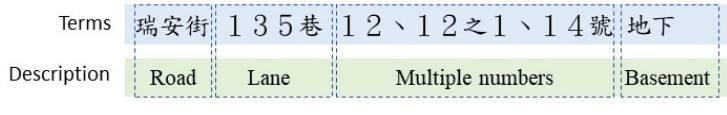
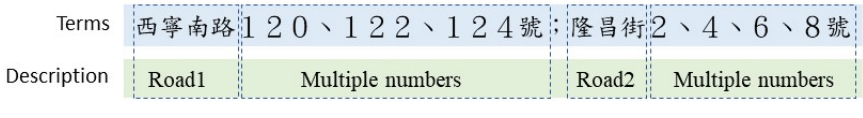


Figure 4 Example of an address entry with multiple road names and numbers (see online version for colours)



Since we intended to build a building database containing building attributes and geospatial information by relating the House Tax data to a geospatial database, for address entries with multiple addresses embedded, only one would be extracted as the representative of the building. As for instances with position description, the proximity of the intended location to the reference point was assumed. Thus, the location of the reference point would be used as a substitute for the real location.

4 Methodology

4.1 Data pre-processing

House Tax data of Taipei City and New Taipei City were used for training and experiments. To be computationally economical, data pre-processing was reduced to as simple as three steps. First, half-width spaces emerging among an address string were removed. Half-width spaces in the original data serve as a separator, separating the address from the entry index; thus, spaces amid a string should be removed for the sake of future application. Second, digits and alphabets were replaced with special characters *D* and *A*. Digits and alphabets are critical reference information for location indexing, but for address parsing, so long as the parser can identify their status as digits and alphabets, the value is not important. Finally, special characters *s* and *e* were added at the start and end of each address as a cue for the parser to recognise positions of input tokens relative to the left or to the right end of a string. One pre-processed example is provided in Figure 5.

Figure 5 An example of the pre-processed address (see online version for colours)

Original	臺北市大安區辛亥路三段２００號
Preprocessed	s臺北市大安區辛亥路三段DDD號e

4.2 Tagging schemes

Each character of an address was assigned a tag after being parsed. The tags are basically address fields as those presented in Table 2. In addition to address fields, there are labels *S* and *E* for special characters marking the start and the end of a string. As for characters that constitutes descriptive spatial relation and inessential information, a special label, *Redundant*, would be assigned so that characters labelled as *Redundant* could be ignored for data linkage. In total, there are 13 candidate tags.

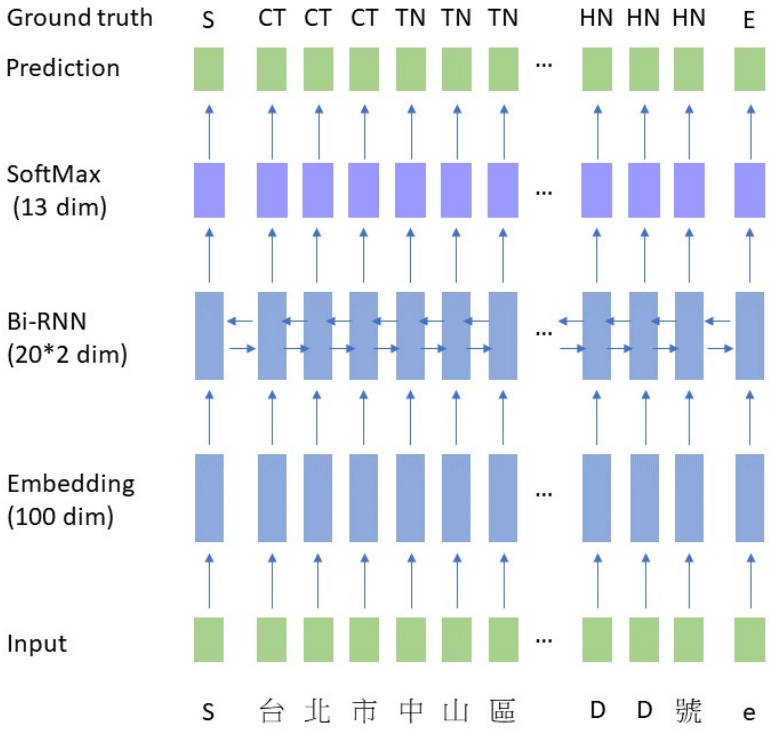
4.3 Model architecture and training details

The model is composed of three parts, one embedding layer, one bi-directional RNN layer, and one fully connected layer. Figure 6 displays the overall architecture. Functions and hyperparameter settings of each layer are as follows.

The first hidden layer, the embedding layer, is a trainable transformation layer that each time receives one input token and outputs a vector. Instead of using a pre-trained layer, like word2vec or BERT that has been trained on an enormous dataset to capture syntactic and semantic features of language, we assumed that a task-specific embedding layer is enough for address parsing since addresses are not as syntactically and semantically complex as everyday language use. Suppose that characters conveying pivotal clues for labelling are from a small group of characters serving as suffixes/indicators that occur repeatedly, in order to reduce the number of trainable parameters for the embedding layer, characters appearing less than 3 times in the addresses of Taipei City would not be embedded distinctively in this model. In total, this layer would learn to embed 811 distinctive input tokens, including Chinese characters,

symbols, special characters (*D*, *A*, *s*, and *e*), and a representative index for all other characters occurring less than 3 times. The output dimension is set to be 100. That is, through the embedding layer, each character would be represented by a vector of 100 dimensions.

Figure 6 Bi-directional RNN based model architecture (see online version for colours)



The second layer is an RNN layer. A unidirectional RNN takes the current input as well as the hidden state from the previous time step to produce the current output, as described below:

$$\overrightarrow{RNN}(x) = \tanh\left(W_x^T x + W_h^T h + b\right) \tag{1}$$

where tanh(·) refers to the hyperbolic tangent function, x is the input vector of the current time step that has n features (n = 100), h is the hidden state vector from the previous time step with a length of m, W_x is an $n \times m$ parameter matrix for the input, W_h is an $m \times m$ parameter matrix of the hidden state, and b is the bias term. The bi-directional RNN is the same as the unidirectional one except that the recurrent computation operates in both forward and backward sequence. The output of the current time step is a concatenated vector of the two output vectors respectively from the forward and backward RNN; that is, the output is $\overleftrightarrow{RNN} = [\overrightarrow{RNN}, \overleftarrow{RNN}]$. In this study, the size of RNN hidden states is set to be 20. Hence, the output of each time step is a 20-dimensional vector for unidirectional RNN and a 40-dimensional vector for bi-directional RNN.

The last layer is a fully connected layer with Softmax function. This layer transforms the output vector from the RNN layer to a vector composed of prediction scores s_j for

each candidate tag, which will be passed through a Softmax function. A Softmax function [equation (2)] computes the prediction probability for each category k and a vector whose dimension equals to the number of all possible tags K, i.e., 13 in this study, will be returned. Each value of the vector indicates the probability of the input token being assigned a particular tag $P_k(x)$.

$$P_k(x) = \frac{\exp\left(s_k(x)\right)}{\sum_j^K \exp\left(s_j(x)\right)} \tag{2}$$

Model training iterated over 10 epochs with Adam as the optimiser (learning rate = 0.001). The batch size is 128 for the training over 900 thousand instances and 30 for smaller training datasets (below 10 thousand). Early stopping was implemented with patience for 4 epochs.

4.4 Active learning strategy

An initial set of 30,066 instances were randomly sampled from the dataset of Taipei City consisting of 1,192,066 entries and were manually labelled. Among the set of labelled data, an initial model was developed on 10,030 random instances, among which 1,500 (about 15%) was reserved as the validation data and the rest 8,530 as the training data, and the remaining 20,036 instances formed the testing set, which will be referred to as test B in the later sections. The left unlabelled instances were randomised and divided into 83 subsets. In every iteration of active learning, one subset was randomly picked, and 5 unlabelled instances from the random subset would be selected by the query selection algorithm. After being labelled, they were added to the training set which would then be used to train a new model. In this study, we adopted Least Confidence as the query selection strategy (Culotta and McCallum, 2004, 2005; Settles and Craven, 2008), which computes the overall uncertainty of labelling a whole entry as described below:

$$x_{uncertainty} = 1 - \prod_{i=1}^{n} p_i \tag{3}$$

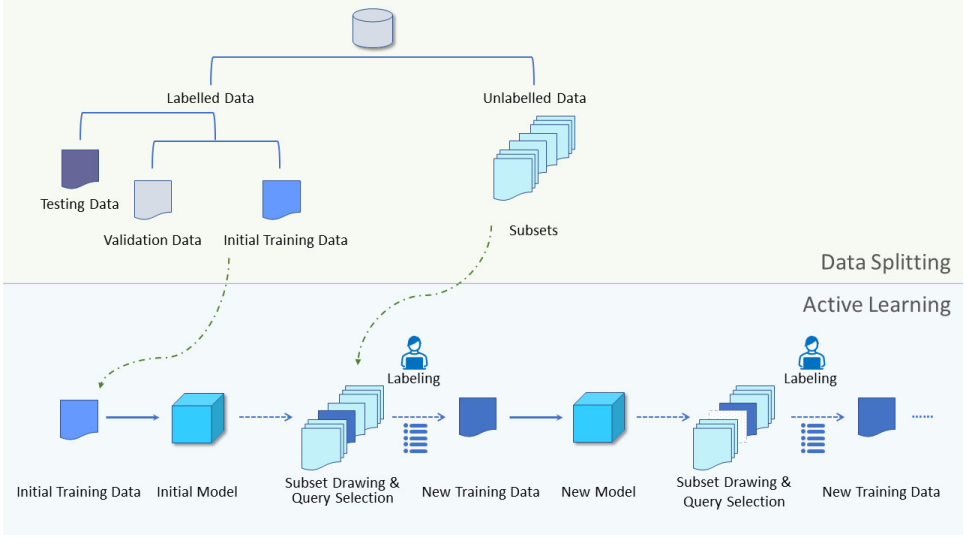
where n is the length of an input sequence x, and p is the probability of a tag that the model considers most likely for a given token i. Instances would be ranked according to the uncertainty level in descending order, and the top five instances were selected in each iteration. The iteration would continue until all subsets were iterated. In short, after the active learning process finished, 415 additional instances would have been added to the training set and 84 models in total, of which the first 83 were interim models, would be obtained.

The active learning procedure used in this study is a modified pool-based procedure. There are two main procedures of active learning, pool-based and batch-based. Conventionally, the pool-based procedure selects one instance at a time to be labelled until the termination condition meets. As for its counterpart, instead of selecting one instance at a time, the batch-based procedure selects a batch of instances of a designated number. There are trade-offs between these two procedures (Settles, 2011). The pro of the pool-based procedure is less overlap in types of selected instances whereas this procedure is time consuming. Since only one instance is selected each time, human oracles are held pending most of the time, waiting for model training and query selection.

On the other hand, the pro of batch-based procedure is that it is more efficient in time. The machine selects a batch for labelling at a time and the labelling tasks can be shared by several oracles. However, since batch of instances are selected by one model, redundancy in the selected instances may, consequently, restrict the learning effectiveness. Algorithms that integrate criterion of certainty, representativeness, and diversity of a batch have been proposed in the literature (Li et al., 2021).

The modified active learning procedure proposed here can avoid these problems. First, we randomly divided all unlabelled data to subsets and each subset would be scanned through at each iteration for querying. The smaller dataset shortens the time spent on query selection, and the randomisation in subset division ensures that each subset is unbiased and approximates the population. Moreover, instead of drawing one instance at a time, having the model select five instances per iteration for labelling can not only increase the number of labelled instances in a faster fashion but also improve the learning effectiveness of the model for acquiring knowledge of a certain type given more relevant instances added. Last but not least, the iteration-based procedure enables the model to evolve over time without ending up biased and limited in performance by a certain shot of queries. Active learning procedure is illustrated in Figure 7.

Figure 7 Active learning procedure



5 Experiments

Experiments were conducted to test if active learning is an effective solution to the obstacles of developing an automatic address parser, especially costly data labelling and limited generalisation. We trained five models either by active learning or by one-shot passive learning. The five models would then be evaluated by three sets of testing data, which are of different distribution and scope. In the following, the five models and the three sets of training data will be introduced. The first three models were trained solely

on addresses of Taipei City and the rest two models were trained on a mixture of addresses from Taipei City and New Taipei City.

5.1 Address data for experiments

Model training and evaluation were performed with address of Taipei City and New Taipei City. The total amount of data utilised is as elucidated in 4.4 above and the two sections below. While having similar names, Taipei City and New Taipei City are two different administrative regions of Taiwan neighbouring to each other and have different address styles. As the capital city of Taiwan, road networks and city planning in Taipei City are relatively more systematic. Address in Taipei City is, therefore, more regular and consistent. As for New Taipei City, because of its broader land, complex geographical properties and uneven distribution of populations, address in New Taipei City has more irregular components and descriptive terms. On top of that, names of geopolitical entities and roads in the two cities are different. As a result, the patterns and forms emerge in address of New Taipei City are supposed to be unfamiliar to a model trained solely by address of Taipei City.

5.2 Passive learning and active learning for model development

One appeal of active learning is that a competent model can be trained using fewer training data. To test if this assumption is valid, a model was trained with the typical passive learning procedure for comparison (model α). Conventionally, a large training set is required for the typical passive learning procedure. To efficiently access a labelled dataset large enough for this model, we utilised a temporary expedient method to fulfil the need. Regular expression (RE) was adapted to format general observations of the standard and common address patterns in each address field. Based on the formats, addresses could be automatically segmented with labels corresponding to address fields assigned. For fear that RE formats would be insufficient and errors might result, we developed a hybrid model to complement the RE approach. The hybrid model left the work of segmentation to an open-source Chinese segmentation tool, Ckiptagger (Li and Ma, 2019), and coded label-assigning rules checking suffix characters and their relative positions (Lin, 2021). To ensure the quality of processed data for training, only those entries that both the RE approach and the hybrid approach agreed upon with respect to segmentation and labelling were collected. It should be noticed that due to restrictions of this data preparation method, only addresses that match the predefined patterns by RE and the rules could be processed and collected. Without further manual processing, the size of the obtained dataset is 1,147,211. 10% of these data were reserved to form a testing set, which is, for the ease of reference, named test A. The rest 90% were further split into 90% for training and 10% for validation. Dataset prepared using this method consists of only regular addresses. Hence, the dataset is biased.

Another appeal of active learning is that the query selection mechanism allows the model to participate in the formation of training data that help itself learn better. This assumption was examined by comparing two models trained on the training sets of the same size whilst one was formed following the active learning procedure by the query selection algorithm [equation (3)], and the other was randomly sampled. To make the two models comparable, the two models used the equal data size with identical instances involved aside from the additional 415 instances, which were randomly sampled in

one-shot passive learning (model β) and strategically selected in active learning (5 per iteration; model γ). Details of data splitting and the active learning procedure have been provided in 4.4 and Figure 7. Evaluation results of the above-mentioned three models could answer the first research question concerning the effectiveness and efficiency of active learning in address parser development.

5.3 Passive learning and active learning for model adaptation

As for the question related to model generalisation, we took addresses of New Taipei City for experiments and trained two more models by active learning (model δ) and one-shot passive learning (model ε). The base model to be adapted is model γ, the one developed by active learning with addresses of Taipei City. At first, 20,036 instances were drawn from the dataset of New Taipei City and were manually labelled to form the third set of testing data, named test C henceforth. The remaining unlabelled data were divided into 84 subsets. The active learning procedure then iterated over the 84 subsets, and, at the end, 420 additional instances were added to the initial training set. As for one-shot passive learning, the 420 additional instances were randomly selected over the subsets. In summary, the two models were trained on 8,945 labelled addresses from Taipei City and 420 labelled addresses from New Taipei City. No new data from New Taipei City was added to the validation data.

Table 3 Summary of models

Model	*Training procedure*	*Training data source*			*Data size*
		Taipei City		*New Taipei City*	
		Regular	*Irregular*		
Model α	Typical passive learning	✓			929,241
Model β	One-shot passive learning	✓	✓		8,945
Model γ	Active learning	✓	✓		8,945
Model δ	Active learning	✓	✓	✓	9,365
Model ε	One-shot passive learning	✓	✓	✓	9,365

Note: Shading represents identical instances shared in the training data.

Table 4 Summary of testing sets

	Taipei City		*New Taipei City*	*Data size*
	Regular	*Irregular*		
Test A	✓			114,721
Test B	✓	✓		20,036
Test C			✓	20,036

To wrap up, we obtained five models, and the five models were respectively evaluated by three testing sets. Table 3 summarises the five models, including the training procedure, the sources of training data, and the data size. Cell shading represents identical instances involved in the training data. As explained above, the data distribution of the training data for typical passive learning is different from that of other models due to the limitations of the method used to prepare labelled data of such a great amount. Table 4 presents the

summary of the three testing sets, including the data source and the data size. It should be noted that since test A was compiled by a separate data preparation procedure along with its corresponding training data, test A and test B are not mutually exclusive. Moreover, instances in the training set of model α may also overlap with test B, and vice versa. Therefore, we should be cautious and conservative about the comparison of model α with other models as well as the evaluation results on test A.

6 Results and discussion

In 6.1, we will first present the evaluation results of models trained solely on addresses of Taipei City. The model loss and accuracy imply the effects of different learning procedures on address parser development. In 6.2, we will continue to inspect the performance of two address parsers updated to manage addresses of New Taipei City and discuss whether active learning helps update a trained model and improve model capability on out-of-domain data. In 6.3, we further analyse the applicability of each parser in real-world tasks. A parser is expected to label as more addresses correctly as possible. We will calculate the accuracy on the address level. To have a detailed comparison between different parsers, we also calculate the prediction accuracy of each label.

6.1 Model performance on in-domain data

The three parsers trained solely on addresses of Taipei City all show superb evaluation results on testing sets composed solely of addresses of Taipei City (i.e., test A and test B). Table 5 presents the loss and accuracy evaluated by different testing sets. It is obvious that the loss and accuracy of the parser trained by typical passive learning are reaching the perfect extreme on the biased testing data which this model is familiar with and has been trained for. Outstanding performance on the unbiased testing data was also found in the two models trained on datasets of the same size respectively by one-shot passive learning and active learning. The former has an accuracy of 0.9992, and the latter 0.9984. This indicates that regardless of the training procedure, the bi-directional RNN architecture is robust enough for the model to learn effectively from the training data and perform well during inference stages.

Table 5 Model loss and accuracy evaluated by different testing sets of Taipei City

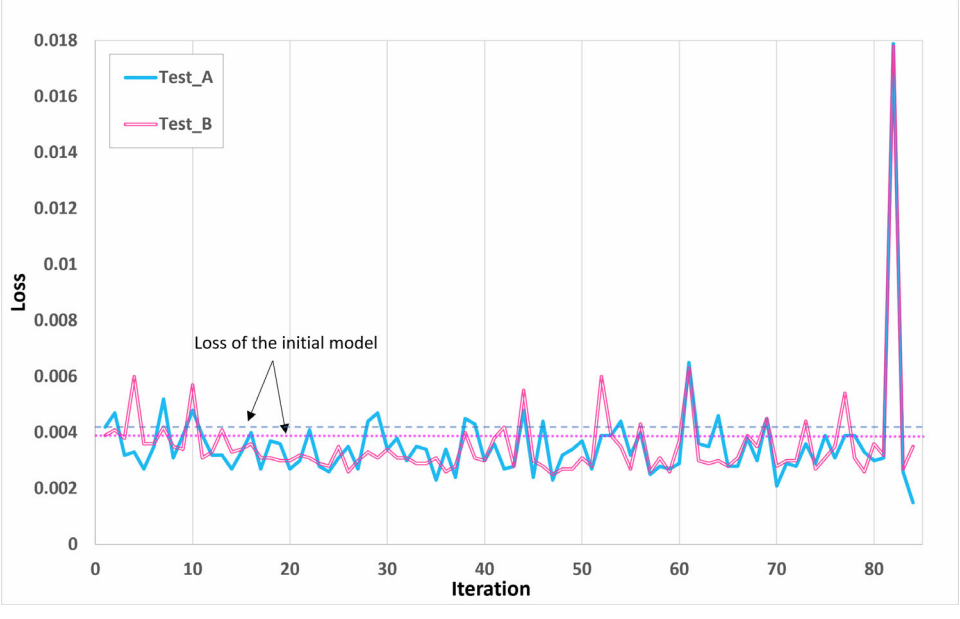
Model	*Training procedure*	*Test A (biased data)*		*Test B (unbiased data)*	
		Loss	*Accuracy*	*Loss*	*Accuracy*
Model α	Typical passive learning	0.0001	**1.0000**	0.0907	0.9956
Model β	One-shot passive learning	0.0031	0.9990	0.0031	**0.9992**
Model γ	Active learning	0.0015	0.9997	0.0035	0.9984

While model α is almost perfect for regular types of addresses, its performance on irregular and atypical instances is compromised. As evaluated by the unbiased dataset, the performance of model α declines by a slim drop in accuracy from 1 to 0.9956. We can find that though model α outperforms model β and model γ on test A, it is surpassed

as evaluated by the unbiased test B. The drop is a sign that the diversity of data types in the training data should be maintained in order to achieve general competence.

Comparing the two models trained with the same training data size following different procedures (i.e., model β and model γ), we can find that active learning is unexpectedly better than one-shot passive learning on the biased set of addresses (test A) while the situation is opposite when the evaluation dataset is unbiased (test B). It is unexpected because we would assume that since the selection algorithm favours irregular types of addresses which are harder to label, with more irregular types of addresses selected, the trained model was supposed to have an advantage on unbiased testing data. However, the evaluation result conflicts with this expectation.

Figure 8 Learning curves of models trained by active learning on the biased and unbiased testing data of Taipei City (see online version for colours)



The learning curve of active learning evaluated by the two sets of testing data may account for this unexpected outcome. As shown in Figure 8, in general, there is no significant difference observed over the learning trajectory on the two testing sets. At the beginning, the initial model has similar competence in the two testing sets. As the iteration progresses, despite that no stable positive inclination observed against the initial model, most interim models along the learning course have smaller loss than the initial one. An abrupt decline in learning performance occurs at the 82nd run on both biased and unbiased testing data. Nonetheless, in the immediately following iteration, the deviation is immediately suppressed, and the performance is resumed.

In light of the unstable learning curve, we suggest that it is a reflection of the gradual increase in the number of more complex and noisy instances in the training data. With more heterogeneous instances added, the learner was still learning how to manage both regular and irregular types of addresses. Taking advantage of the iterative query selection mechanism, the learner was able to keep learning and adjusting itself before the deviation

went extreme. Under our termination condition, the learner was forced to stop as soon as all subsets were iterated before striking the balance, leading to overfitting effects on test A and inferior performance on test B, as compared to model β. However, it should be noted that even with ebb and flow, the fluctuations are contained in a tiny range.

In these experiments, advantages of active learning are not demonstrated yet. The results seem not compelling enough for practitioners to apply active learning at the cost of iterative querying, labelling, and training especially when its counterpart model trained by one-shot passive learning has almost the same performance or even better on unbiased data.

6.2 Model performance on out-of-domain data

To test the effects of active learning on improving model generalisation, we conducted another experiment using addresses of New Taipei City, which contains some values and patterns absent in the dataset of Taipei City. Model γ was adapted with 420 new address entries of New Taipei City added. Again, one model was trained following the passive learning procedure and the other active learning.

Table 6 shows the overall evaluation results of models on the three testing sets. Apparently, the first three models have limited competence in labelling addresses of New Taipei City which they are unfamiliar with. Nevertheless, they, especially model α, still maintain a certain level of accuracy. This implies that addresses of Taipei City and New Taipei City have features overlapped to certain extent and the underlying features can be captured by the NN-based models.

Table 6 Model loss and accuracy evaluated by three testing sets

Model	*Training procedure*	*Test A*		*Test B*		*Test C*	
		Loss	*Acc*	*Loss*	*Acc*	*Loss*	*Acc*
Model α	Typical passive	0.0001	*1.0000*	0.0907	0.9956	0.6778	0.9307
Model β	One-shot passive	0.0031	0.9990	0.0031	*0.9992*	0.3918	0.8314
Model γ	Active learning	0.0015	0.9997	0.0035	0.9984	1.1211	0.7581
Model δ	Active learning	0.0031	0.9992	0.0027	0.9991	0.0045	*0.9986*
Model ε	One-shot passive	0.0025	0.9992	0.0027	0.9991	0.0099	0.9966

For the newly trained models catering to addresses of New Taipei City, we can see that they both reach accuracy over 0.99, and the model δ narrowly outperforms model ε. In addition to the high accuracy on the new data, their competence in labelling the old datasets, addresses of Taipei City, also improved (test B). The results cohere with our argument above that addresses share common features so that the newly added data should impose no or few adverse effects on the prior learning outcomes. In fact, the new data seems to provide more relevant instances for the model to learn better, as demonstrated by the improvement on unbiased data of Taipei City from 0.9984 to 0.9991. As for the narrow advantage of model δ over model ε, it can be attributed to the active learning mechanism that precisely, instead of by chance, selects instances containing features the previous model is ignorant of, especially features of redundant elements which will be elaborated in 6.3.

In other words, the continued active learning not only improves generalisation efficiently with the addition of a small number of new instances added to the training data

but also retains and even hones the previously established competence. To sum up, the advantage of active learning is demonstrated here. When the number of labelled data is limited, active learning is promising as it ensures that the labelled training data is informative and helpful for the learner. If the goal is to update an existing model, active learning makes the updating process effective and efficient by querying labels of a few informative new instances.

6.3 Model performance on real-world tasks

In real-world tasks, we expect a parser to correctly label every component of an address entry. To examine the applicability of trained parsers in real tasks, the address accuracy, the proportion of correctly labelled address entries out of all entries, was computed. Table 7 displays the address accuracy of the five models on three testing sets. The first three models all achieve high address accuracy on either biased or unbiased data of Taipei City. However, as applied to label addresses of New Taipei City, even with acceptable model accuracy as presented in Table 6, the parsers failed to satisfy the standard for real-world tasks. The low address accuracy indicates that a large portion of parsed addresses contain wrongly labelled components, resulting in additional costs in post-processing and correction.

Table 7 Address accuracy of models on three testing sets

Model	*Training procedure*	*Test A*	*Test B*	*Test C*
Model α	Typical passive learning	*0.9999*	0.9725	0.1158
Model β	One-shot passive learning	0.9928	*0.9931*	0.0001
Model γ	Active learning	0.9943	0.9851	0.0001
Model δ	Active learning	0.9934	0.9903	*0.9779*
Model ε	One-shot passive learning	0.9933	0.9904	0.9630

Similar to the results of model accuracy, the two reinforced models perform well on three testing sets. While they are not the best model for addresses of Taipei City, they handle addresses of New Taipei City remarkably better than others. Moreover, though the model δ and model ε present similar outcomes in model accuracy, the advantage of active learning becomes obvious when the performance is measured at the address level. More addresses can be correctly labelled by model δ than model ε. From this perspective, active learning yields a general and balanced parser for both addresses of Taipei and New Taipei City.

Further statistics of the label accuracy disclose what determines model performance at the address level. Table 8 presents the accuracy of predicting each label. Overall, the label *redundant* is a weak point of most models. It's a label that marks unexpected and redundant characters emerging in an address. Characters that should be assigned this label may appear anywhere in a string and the form is diverse. Some may appear normal but are in fact redundant because of its relative position with other components. For example, there are instances in which the city name is repeated twice. In this case, the second one should be labelled as *redundant*. Therefore, the ability to assign *redundant* correctly reflects a model's ability to identify noises. A closer examination reveals that models trained by active learning are generally more competent than models trained by passive learning in terms of assigning *redundant*.

Table 8 Label accuracy of the five models on three testing sets

Label	*Test A*				
	Model α	*Model β*	*Model γ*	*Model δ*	*Model ε*
City	1.0000	1.0000	1.0000	1.0000	1.0000
Town	1.0000	1.0000	1.0000	1.0000	1.0000
Tract	1.0000	1.0000	1.0000	1.0000	1.0000
Neighbourhood	1.0000	1.0000	1.0000	1.0000	1.0000
Road	1.0000	1.0000	1.0000	1.0000	1.0000
Lane	1.0000	1.0000	0.9999	1.0000	1.0000
Alley	1.0000	1.0000	1.0000	1.0000	0.9999
House number	1.0000	0.9998	0.9996	0.9999	0.9995
Ground floor	1.0000	1.0000	1.0000	1.0000	0.9998
Basement	1.0000	0.9988	0.9988	1.0000	1.0000
Redundant	0.9992	0.6732	0.9133	0.7267	0.7553
	Test B				
City	1.0000	1.0000	1.0000	1.0000	1.0000
Town	1.0000	1.0000	1.0000	0.9999	1.0000
Tract	1.0000	1.0000	1.0000	0.9999	1.0000
Neighbourhood	1.0000	1.0000	1.0000	1.0000	1.0000
Road	0.9995	0.9999	0.9998	0.9998	0.9998
Lane	1.0000	1.0000	0.9998	0.9999	1.0000
Alley	1.0000	1.0000	1.0000	0.9998	1.0000
House number	0.9998	0.9997	0.9983	0.9995	0.9990
Ground floor	0.9963	0.9991	0.9972	0.9980	0.9977
Basement	0.4589	0.9668	0.7634	0.9412	0.9550
Redundant	0.6929	0.9151	0.9799	0.9337	0.9435
	Test C				
City	0.7312	0.0862	0.3797	1.0000	1.0000
Town	0.8839	0.7215	0.2739	0.9995	0.9987
Tract	0.9637	0.9885	0.9011	0.9975	0.9991
Neighbourhood	0.9954	0.9980	0.8243	0.9987	0.9987
Road	0.9839	0.9951	0.9283	0.9994	0.9994
Lane	0.9858	0.9966	0.9165	0.9991	0.9984
Alley	0.9795	0.9977	0.9135	0.9994	0.9999
House number	0.9831	0.9953	0.8953	0.9993	0.9946
Ground floor	0.9819	0.9924	0.9132	0.9975	0.9964
Basement	0.4420	0.9655	0.8605	0.9608	0.9624
Redundant	0.2461	0.3408	0.6372	0.8948	0.6183

Another point worth noticing is that label accuracy can account for the low address accuracy of the first three models on addresses of New Taipei City. The city name and

the town name are essential components that are rarely omitted in an address, and the incapability of a parser to recognise these two components means that it is destined to be scored low at the address level. The weak performance of these models on assigning *city* and *town* is supposed to be the consequence of models relying too heavily on the limited variety of forms of city names and town names. When addresses containing unfamiliar city names and town names are encountered, the inference is then disturbed. Other address fields whose values are more various show no accuracy decline as dramatic as *city* and *town*. The high variability of field values encourages models to take more features, e.g., adjacent characters and label relations, into consideration.

7 Conclusions

In this study, we have adopted one-shot passive learning and active learning procedures to train address parsers based on the bi-directional RNN architecture. Three models were trained solely on addresses of Taipei City following different training procedures, including the typical passive learning where a large amount of labelled data was involved, active learning, and one-shot passive learning with the same data amount involved as active learning. Two more models were trained with addresses of New Taipei City added to the previous training set of Taipei City built for active learning. These five models were respectively evaluated by three testing sets of different scope and data diversity. Based on the results of experiments, we can answer the two research questions.

For the first question concerning whether active learning is an economical training procedure, the experiment result suggests that when the initial model has already shown high-quality performance, adding strategically selected instances to the training set may cause adverse effects due to the unstable learning trajectory, as presented in Figure 8. In this situation, one-shot passive learning which randomly samples instances for labelling to enlarge the training set is a better training strategy. From another perspective, the outstanding performance of the initial model may be attributed to the initial training set which is already large enough for the model to learn essential features to some extent. Therefore, the benefits of active learning are not demonstrated.

As for the effect of active learning on improving model generalisation, results indicate that the existing model trained on addresses of Taipei City can be adapted to accommodate addresses of New Taipei City with only 420 new labelled instances selected through the query selection algorithm. The model trained by active learning shows better performance than that trained by one-shot passive learning while the number of additional instances is equal in the training sets for the two models. Moreover, its performance on the original data, addresses of Taipei City, is retained and even improved. This demonstrates that active learning poses positive effects on improving model generalisation with a few but informative data.

From a practical perspective, it's common that a model is initially developed to deal with a task at hand without considering the future development and potential broader applications. When a new task of broader scope is given and the existing model is not compatible with the new data, a common option is to retrain the model. However, this may risk failing to train a model that manages both the old and new data. Alternatively, active learning allows the existing model to select instances that it is uncertain about and adjust itself by iteration. The training is then made efficient and the performance on both sides can be kept.

While the basic RNN architecture has achieved competent performance in this study, in the future, it can be examined if the model architecture is replaced with the state-of-the-art architecture, seq2seq or LSTM, whether the advantage of active learning over one-shot passive learning is still valid. In addition, several query selection algorithms have been proposed in literature (Craig et al., 2019; Li et al., 2021; Settles, 2010; Settles and Craven, 2008, Shen et al., 2017). Different algorithms influence the types of instances being selected and the distribution of the selection. The effect of different selection strategies on active learning when used for model adaptation is another variable that is worth future inspection.

This paper has provided an innovative way to adapt a developed model for new data without retraining the whole model. Though this study focused on address in Chinese in Taiwan, we suggest that the model design and active learning procedure proven capable here can be applied to address in other languages and countries with small changes made in the output layer corresponding to the labels of the target address system of interest and in the level of input token embedding suitable for the particular language (e.g., applying word-level embedding for languages with natural word segmentations). Moreover, the proposed active learning procedure should be effective in upgrading models of other tasks, apart from address parsing, in which variation in the style or genre of the input data has an impact, e.g., the task of information extraction on texts of Wikipedia and Twitter feeds (Wing and Baldridge, 2011).

References

Abid, N., ul Hasan, A. and Shafait, F. (2018) 'DeepParse: a trainable postal address parser', *2018 Digital Image Computing: Techniques and Applications (DICTA)*, Canberra, Australia, pp.1–8, https://doi.org/10.1109/DICTA.2018.8615844.

Abrol, S., Khan, L. and Muhaya, F.T.B. (2013) 'MapIt: a case study for location driven knowledge discovery and mining', *International Journal of Data Mining, Modelling and Management*, Vol. 5, No. 1, pp.57–75, https://doi.org//10.1504/IJDMMM.2013.051923.

Chang, C.H., Chuang, H.M., Huang, C.Y., Su, Y.S. and Li, S.Y. (2016) 'Enhancing POI search on maps via online address extraction and associated information segmentation', *Applied Intelligence*, Vol. 44, No. 3, pp.539–556, https://doi.org/10.1007/s10489-015-0707-5.

Chiticariu, L., Li, Y. and Reiss, F.R. (2013) 'Rule-based information extraction is dead! Long live rule-based information extraction systems!', *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, Seattle, WA, pp.827–832.

Churches, T., Christen, P., Lim, K. and Zhu, J.X. (2002) 'Preparation of name and address data for record linkage using hidden Markov models', *BMC Medical Informatics and Decision Making*, Vol. 2, No. 1, pp.1–16, https://doi.org/10.1186/1472-6947-2-9.

Comber, S. and Arribas-Bel, D. (2019) 'Machine learning innovations in address matching: a practical comparison of word2vec and CRFs', *Transactions in GIS*, Vol. 23, No. 2, pp.334–348, https://doi.org/10.1111/tgis.12522.

Craig, H., Yankov, D., Wang, R., Berkhin, P. and Wu, W. (2019) 'Scaling address parsing sequence models through active learning', *SIGSPATIAL'19: Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Chicago, IL, pp.424–427, https://doi.org/10.1145/3347146.3359070.

Culotta, A. and McCallum, A. (2004) 'Confidence estimation for information extraction', *Proceedings of HLT-NAACL 2004: Short Papers on XX - HLT-NAACL'04*, Morristown, NJ, pp.109–112, https://doi.org/10.3115/1613984.1614012.

Culotta, A. and McCallum, A. (2005) 'Reducing labelling effort for structured prediction tasks', *Proceedings of the National Conference on Artificial Intelligence*, Pittsburgh, PA, pp.746–751.

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K. (2019) 'BERT: pre-training of deep bidirectional transformers for language understanding', *Proceedings of the 2019 Conference of the North*, Stroudsburg, PA, pp.4171–4186, http://dx.doi.org/10.18653/v1/N19-1423.

Gal, Y., Islam, R. and Ghahramani, Z. (2017) 'Deep Bayesian active learning with image data', *34th International Conference on Machine Learning, ICML 2017*, Vol. 3, pp.1923–1932, https://doi.org/10.17863/CAM.11070.

Javidaneh, A., Karimipour, F. and Alinaghi, N. (2020) 'How much do we learn from addresses? On the syntax, semantics and pragmatics of addressing systems', *ISPRS International Journal of Geo-Information*, Vol. 9, No. 5, pp.8–15, https://doi.org/10.3390/ijgi9050317.

Kong, L., Dyer, C. and Smith, N.A. (2016) 'Segmental recurrent neural networks', *ICLR*, pp.1–19, http://arxiv.org/abs/1511.06018.

Küçük Matci, D. and Avdan, U. (2018) 'Address standardization using the natural language process for improving geocoding results', *Computers, Environment and Urban Systems*, Vol. 70, pp.1–8, https://doi.org/10.1016/j.compenvurbsys.2018.01.009.

Lample, G. et al. (2016) 'Neural architectures for named entity recognition', *Proceedings of NAACL-2016*, San Diego, CA [online] http://arxiv.org/abs/1603.01360 (accessed 1 August 20021).

Li, H., Wang, Y., Li, Y., Xiao, G., Hu, P. and Zhao, R. (2021) 'Batch mode active learning via adaptive criteria weights', *Applied Intelligence*, Vol. 51, No. 6, pp.3475–3489, https://doi.org/10.1007/s10489-020-01953-4.

Li, P.H. and Ma, W.Y. (2019) *Ckiptagger, 0.1.0*, CKIP Lab [online] https://github.com/ckiplab/ckiptagger (accessed 1 September 2020).

Lin, Y., Kang, M., Wu, Y., Du, Q. and Liu, T. (2020) 'A deep learning architecture for semantic address matching', *International Journal of Geographical Information Science*, Vol. 34, No. 3, pp.559–576, https://doi.org/10.1080/13658816.2019.1681431.

Lin, Y.X. (2021) 'Chinese address parsing through neural network', Paper presented at *Taiwan Geographic Information Society Annual Conference and Symposium*, 20–21 October, Taichung, Taiwan.

Mikolov, T. Sutskever, I., Chen, K., Corrado, G. and Dean, J. (2013) 'Distributed representations of words and phrases and their compositionality', *Advances in Neural Information Processing Systems*, Stateline, NV, pp.3111–3119.

Mokhtari, S. Mahmoody, A., Yankov, D. and Xie, N. (2019) 'Tagging address queries in maps search', *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, HI, pp.9547–9551, https://doi.org/10.1609/aaai.v33i01.33019547.

Palaz, D., Collobert, R. and Doss, M.M. (2013*) End-to-end Phoneme Sequence Recognition using Convolutional Neural Networks* [online] http://arxiv.org/abs/1312.2137 (accessed 3 January 2022).

Settles, B. (2010) *Active Learning Literature Survey*, *Computer Sciences Technical Report*, Computer Sciences Technical Report 1648, University of Wisconsin-Madison [online] http://burrsettles.com/pub/settles.activelearning.pdf (accessed 28 May 2021).

Settles, B. (2011) 'From theories to queries: active learning in practice', *Workshop on Active Learning and Experimental Design*, Sardinia, Italy, pp.1–18 [online] http://jmlr.org/proceedings/papers/v16/settles11a/settles11a.pdf (accessed 23 July 2021).

Settles, B. and Craven, M. (2008) 'An analysis of active learning strategies for sequence labelling tasks', *EMNLP'08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Honolulu, HI, pp.1070–1079, https://doi.org/10.3115/1613715.1613855.

Sharma, S., Ratti, R., Arora, I., Solanki, A. and Bhatt, G. (2018) 'Automated parsing of geographical addresses: a multilayer feedforward neural network based approach', *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, Laguna Hills, CA, pp.123–130, https://doi.org/10.1109/ICSC.2018.00026.

Shen, Y., Yun, H., Lipton, C.Z., Kronrod, Y. and Anandkumar, A. (2017) 'Deep active learning for named entity recognition', *Proceedings of the 2nd Workshop on Representation Learning for NLP*, Vancouver, Canada, pp.252–256.

Sun. J.Y. (2020) *Jieba* [online] https://github.com/fxsjy/jieba (accessed 16 February 2022).

Wang, M., Haberland, V., Yeo, A., Martin, A., Howroyd, J. and Bishop, J.M. (2016) 'A probabilistic address parser using conditional random fields and stochastic regular grammar', *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, Barcelona, Spain, pp.225–232, https://doi.org/10.1109/ICDMW.2016.0039.

Waszczuk, J., Glowinska, K., Savary, A., Przepiórkowski, A. and Lenart, M. (2013) 'Annotation tools for syntax and named entities in the National Corpus of Polish', *International Journal of Data Mining, Modelling and Management*, Vol. 5, No. 2, pp.103–122, https://doi.org/10.1504/IJDMMM.2013.053691.

Wing, B.P. and Baldridge, J. (2011) 'Simple supervised document geolocation with geodesic grids', *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, OR, pp.955–964.

Xu, L., Du, Z., Mao, R., Zhang, F. and Liu, R. (2020) 'GSAM: a deep neural network model for extracting computational representations of Chinese addresses fused with geospatial feature', *Computers, Environment and Urban Systems*, Vol. 81, pp.1–12, https://doi.org/10.1016/j.compenvurbsys.2020.101473.

Yassine, M., Beauchemin, D., Laviolette, F. and Lamontagne, L. (2020) 'Leveraging subword embeddings for multinational address parsing', *2020 6th IEEE Congress on Information Science and Technology (CiSt)*, Agadir, Morocco, pp.353–360, https://doi.org/10.1109/CiSt49399.2021.9357170.