A Novel Conditional Random Fields Aided Fuzzy Matching in Vietnamese Address Standardization

Hong-Ngoc Bui
Hanoi University of Science and Technology
Hanoi, Vietnam
ngocjr7@gmail.com

Viet-Trung Tran*
Hanoi University of Science and Technology
Hanoi, Vietnam
trungtv@soict.hust.edu.vn

ABSTRACT

Address standardization is the process of recognizing and normalizing free-form addresses into a common standard format. In today's digital economy, this process is increasingly challenging such as in e-commerce fulfillment, logistic planning, geographical data analysis, real-estate, and social network mining, etc. Traditional approaches mostly follow two directions: Named Entity Recognition (NER) and fuzzy matching. Particularly, for Vietnamese address, neither these two approaches are efficient due to sparse and erroneous data. In this paper, we propose a novel approach that leverages NER model as a suggestion to re-rank potential address candidates obtained by the fuzzy matching stage. We develop a *log-linear model* for this re-ranking purpose. Our experiments showed that it outperforms both NER and fuzzy matching approaches with an accuracy of 88%, and suggested further applications on different language data.

CCS CONCEPTS

• **Information systems → Probabilistic retrieval models**.

KEYWORDS

address normalization, named entity recognition, conditional random fields, fuzzy matching

ACM Reference Format:
Hong-Ngoc Bui and Viet-Trung Tran. 2019. A Novel Conditional Random Fields Aided Fuzzy Matching in Vietnamese Address Standardization. In *The Tenth International Symposium on Information and Communication Technology (SoICT 2019), December 4–6, 2019, Hanoi - Ha Long Bay, Vietnam, Viet Nam.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3368926.3369687

1 INTRODUCTION

Address standardization is the process of recognizing and normalizing free-form addresses into a common standard format. In today's digital economy, this process is increasingly needed for speed and accuracy such as in e-commerce order fulfillment, logistic planning, large-scale geographical data analysis, real-estate data mining, and social network mining, etc.

street	**ward**	**district**	**city**
tây sơn	null	đống đa	hà nội
cát linh	null	đống đa	hà nội
null	văn miếu	đống đa	hà nội
...	...	...	...
đống đa	null	hải châu	đà nẵng
null	bình hiên	hải châu	đã nẵng
null	null	hải châu	đà nẵng
...	...	...	...

x_1: 'tay son - dong da'
y_1: {'street': 'Tây Sơn', 'district': 'Đống Đa', 'city': 'Hà Nội'}
x_2: 'phường vân giang ninh bình ninh bình'
y_2: {'ward': 'vân giang', 'district': 'ninh bình', 'city': 'ninh bình'}
x_3: 'vĩnh lại phú thọ'
y_3: {'ward': 'vĩnh lại', 'district': 'lâm thao', 'city': 'phú thọ'}

Figure 1: An example of referenced address table (w) and input (x), output(y) of our task.

Given raw address string provided by the users, address standardization process generally consists of these three following steps: (1) recognizing the entities mentioned in the raw input string, (2) standardizing the recognized entities, (3) fulfilling the missing address fields according to the predefined standard format. Step (3) requires a complete set of valid addresses, usually organized in a multiple-columns table as in Figure 1.

Precisely, we formulate an address standardization task as finding $y = f(x, w) | y \in w$ for a given input x. For the sake of simplicity, x is assumed to map correctly to one and only one record y in the referenced address table w. Figure 1 also shows some examples of inputs ($x1, x2, x3$), and expected outputs ($y1, y2, y3$). To date, step (1) is commonly addressed by Named Entity Recognition models such as *Hidden Markov Model* (HMM) [2, 7, 14], *Conditional Random Field* (CRF) [18, 21],... Step (2) and (3) are mostly rule-based that hard code how to match the entities obtained from the NER model with the referenced address table to find the most relevant address [6]. However, this schema has a drawback: the accuracy of the model greatly depends on the dataset. In order to build a practical model, we have to build a high-coverage dataset which is especially difficult for Vietnamese addresses.

Alternative approaches consider address standardization as fuzzy search/matching on the referenced address table [3, 5, 15]. These approaches rely on linguistic similarity and do not require training phrases. However, fuzzy matching works poorly with a structured or semi-structured address. It is often difficult to identify important

[*] This is the corresponding author

SoICT 2019, December 4–6, 2019, Hanoi - Ha Long Bay, Vietnam, Viet Nam
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-7245-9/19/12…$15.00
https://doi.org/10.1145/3368926.3369687

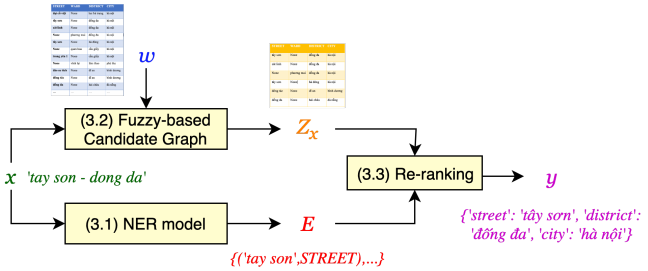
w
(3.2) Fuzzy-based Candidate Graph
Z_x
(3.3) Re-ranking
y
x 'tay son - dong da'
(3.1) NER model
E
{'street': 'tây sơn', 'district': 'đống đa', 'city': 'hà nội'}
{('tay son',STREET),...}

Figure 2: Vietnamese Address Standardizer

features such as indicators, punctuations, structures of text, etc, which are the advantages of NER models. For example, with the input: ’phường vân giang ninh bình ninh bình’ . Fuzzy matching has no mechanism to distinguish ’vân giang’ from the name of the ward or the name of the street, even though the prefix ’phường’ has been included before.

In this work, we propose a novel approach combining fuzzy matching and NER models that achieved high accuracy without heavily depending on the quality of the underlying blocks (eg. NER and fuzzy matching). Our approach can be applied to Vietnamese and also other low-resource languages where NER based models are simply not good due to data sparsity. In this paper, we detail our approach and implementation in Section 2 and Section 3. Section 4 discusses our experiments and results. Then, Section 5 details other related researches. Finally, we will conclude and discuss about future directions in section 6.

2 A HYBRID APPROACH

Vietnamese address standardization faces some particularly unique challenges that greatly affect the accuracy of an address standardizer. Firstly, there is very sparse address data. Several open source systems such as *gisgraphy.com*, *Photon* cannot parse Vietnamese addresses. Secondly, typographical errors are worse due to input methods (Vni or Telex), sign omission (’vĩnh lại’ - ’vinh lai’), the variant types of unicode (’hòa bình’ - ’hoà bình’), inconsistent abbreviations (’hà nội’ - ’hn’) or the ambiguity between indicator and place names (’đường lâm gia lâm’).

Our approach aims to maximize the accuracy in the situation that a corresponding NER model is not efficient due to the sparseness of training data. Taking inspiration from the ranking step in a few recent studies on the question answering challenges [11, 16], we treat the address standardization problem as a ranking problem on a referenced address table w, where the candidates are documents in that table. Instead of using NER as a preprocessing data step of fuzzy matching, we leverage a *log-linear model* [8] as the additional layer to combine the results of NER and the fuzzy matching processes. To be specific, fuzzy matching process will propose a shortlist of potential address candidates. NER results are used as suggestions so that the re-ranking model can reevaluate fuzzy matching results. In this scheme, our re-ranking model may fix NER errors if it is provided with the appropriate features.

The general flow of our approach is shown in Figure 2. Given an input string x, the main task of the NER model block is to produce a set of entities E that are mentioned in the input string. The NER results will give us an overview of the potential entities and their labels. It also shows the structure of the input, based on special features such as capitalization, punctuation and the combination of all syllables in the sentence. The fuzzy matching block preforms fuzzy-query/matching for input string x on referenced address table w. This block will quickly review all documents in the reference table and return the most potential candidates Z_x. Finally, a re-ranking block implements a log-linear model to re-rank Z_x by the labels E. We will choose the candidate with the highest probability as the final outcome of the model. An example of the input and output of each block is shown in Figure 2.

3 IMPLEMENTATION

In this section, we will detail the implementation of each of the framework components. The following subsection describes our NER model, subsection 3.2 present our work in performing fuzzy matching and pruning its results. Finally, we detail how we re-rank the candidates in subsection 3.3.

3.1 Named Entity Recognition block: Conditional Random Fields

We choose Conditional Random Fields (CRFs) [12], a discriminative undirected probabilistic graphical model as our Named Entity Recognition block for its popularity, robustness and ease of implementation. Given the observation sequences $X = (x_1, x_2, ..., x_n)$, CRFs infers the label sequences $Y = (y_1, y_2, ..., y_n)$ by the conditional probability distribution $P(y|x)$, rather than a joint distribution $P(Y, X)$ over both label and observation sequences. As an undirected graphical model, CRFs represents X - input and Y - output variables as nodes in two disjoint sets, and learns potential functions that are conditional on X.

We follow IOB format [19] *(Inside-Outside-Begining)* to tag address field names at the syllabic level. Beside *STREET*, *WARD*, *DISTRICT*, *CITY* labels, we also use other labels to tag the indicator of entity names. For example, with input string ’đường đại cồ việt’ , we will label syllable ’đường’ as indicator of street field *STREET_INDICATOR*. Particularly, with an indicator in front of a number is entity name (e.g. ’quận 7’), we consider ’quận’ to be a part of an entity name *DISTRICT* rather than an indicator because digits alone do not make sense.

For feature extraction, beside using common feature like *isupper*, *istitle*, *isdigit*, *ispunctuation*, *etc*, we also query the referenced address table to extract additional features to syllables. A syllable will have feature text *B_f_nm* if that syllable starts the name of an entity in the field f and *I_f_nm* if the syllable is found in the middle or the end of the name of an entity in field f of our referenced address table. For instance, with an example of referenced address table shown in Figure 1, the syllable ’đống’ will have additional features: *’B_street_nm’* and text *’B_district_nm’*. In this work, we use open source *python-crfsuite*[1] to implement this model.

After CRF tagging, we concatenate continuous labels (in IOB format) to produce entities and their labels. We will discuss how we use these labels to extract features for the log-linear model in subsection 3.3.

[1] https://github.com/scrapinghub/python-crfsuite

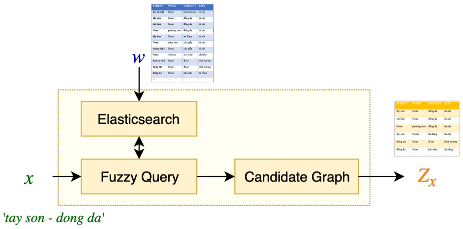

Figure 3: Fuzzy-based Candidate Graph

3.2 Fuzzy matching block: Fuzzy-based Candidate Graph

The purpose of our fuzzy matching block is to limit the number of potential address candidates, $\{Z_x|Z_x \subset w\}$ from the given input x for further re-ranking.

Candidate Graph. We leverage a fuzzy-base candidate graph proposed in [5] for computing candidate scores in *address geocoding* problem. Candidate Graph is a non-cyclic hierarchical graph, a node is an address entity represented by its field name and entity name. The *admin level* of the field that the node contains is equal to the depth of that node. The root node is the default node that has admin level as 0 and contains the field {Country: 'Việt Nam'}. Each branch traversing from the root node to the leaf node in the candidate graph represents a standardized address with all necessary field levels.

Fuzzy based candidate graph construction. A candidate graph is constructed for each input query x. We query the input on each field (*admin level*) of the referenced address table to find the addresses which are likely to be mentioned in the input contain the entity name on that field. We implement this method using *elasticsearch*[2] framework to perform fast fuzzy query for the input string. We also use a mixture of TF-IDF score to boost *elasticsearch* ranking. We first index referenced table w to *elasticsearch indices.*

For each field, we split the input string into n-grams tokens, query n-grams on that field and the rest of input on the remaining fields. The score of each candidate in this query will be set to the score of the queried field on that candidate. It is also the score if the node contains that field. For example, with the input string:

'Tay Son Dong Da'

we split into 2-grams tokens including: *{'Tay Son', 'Son Dong', 'Dong Da'}*. And the rest for each 2-grams are *{'Dong Da', 'Tay Da', 'Tay Son'}*. When querying on $STREET$ field, the 2-grams *'Tay Son'* is queried on $STREET$ field, and the rest, *'Dong Da'*, we will query it on the remaining fields ($WARD$, $DISTRICT$, $CITY$,...).

Node score calculation in candidate graph. We define a type field at each node of the graph. Node explicitly retrieved from a fuzzy matching query is of type EXPLICIT. Ancestral nodes associated with this EXPLICIT node are of type IMPLICIT. Besides, we also define *net score* of a node: the highest score of a branch from that node to a leaf node. Mathematically, the net score of a node is calculated by the following *dynamic programing* [9] formula:

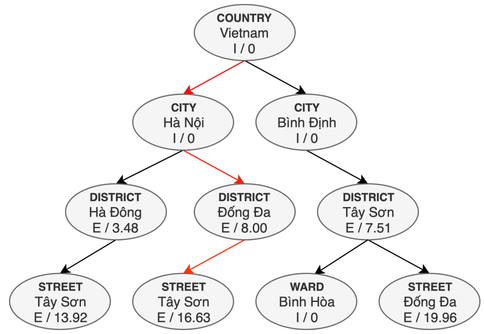

Figure 4: Candidate Graph Example: Each node includes: field, entity name, node type (I denote *IMPLICIT* and E denote *EXPLICIT*), the score of the node. The red arrow denote branch with highest score.

$$S(d) = s(d) + max(S(c(d))) \tag{1}$$

where $S(d)$ is the *net score* of node d, $s(d)$ is the elaticsearch score of the node d if that node is EXPLICIT, 0 if it is IMPLICIT. $c(d)$ is sub-branches ($c(d)$ is a child of the address d in terms of admin level hierarchy). After calculation, a candidate graph consists of addresses represented by branches with the highest scores.

Candidate selection. We prune candidate graph by eliminating branches with low scores. We only keep $K = 8$ nodes with the highest *net scores* at each admin level. The reason why we use beam search instead of taking the top rankings is to avoid this bias for a high score node. Overall, we produce potential candidates Z_x for further re-ranking.

3.3 Re-ranking block: log-linear estimation model

Given Z_x candidates, our main task is to infer the most likely standardized address $y \in w$ that is corresponded to input raw address x. To this end, we extract a feature vector $\phi(x, E, z)$ for each candidate $z \in Z_x$ and define a log-linear distribution [8] over the candidate that ranks every candidate $z \in Z_x$:

$$p_\theta(z|x, E) \propto exp\{\theta^T \phi(x, E, z)\} \tag{2}$$

where θ is parameter vector. The candidate with the highest probability is the most likely standardized address output.

Objective function. The parameter θ that maximizes the objective function was defined by following equation:

$$L(\theta) = \sum_{i=1}^{N} \log p_\theta(y^{(i)}|x^{(i)}, E^{(i)}) - \frac{\lambda}{2}||\theta||^2 \tag{3}$$

where $y^{(i)} \in Z_{x^{(i)}}$, $E^{(i)}$ is a set of entities and labels were generated from $x^{(i)}$. Because fuzzy matching process does not always produce the standardized address $y^{(i)}$, we will remove the instances that $y^{(i)} \notin Z_{x^{(i)}}$ from training set.

[2] https://github.com/elastic/elasticsearch

Table 1: Example features for input string x_3, correct candidate z and entities E obtained from NER model

x_3: *'vĩnh lại phú thọ'*
E: [('street',*'vĩnh lại'*)*, ('city',*'phú thọ'*)]
z: {'ward': *'vĩnh lại'*, 'district': *'lâm thao'*, 'city': *'phú thọ'*}
*: *NER was wrong in this case, 'vĩnh lại' indicates a ward name*

Feature Name	Value	Comments
$EL(district)$	*float*	elastic score of *'lâm thao'*
$E[city] == z[city]$	*binary*	*'phú thọ' == 'phú thọ'?*
$f_{jac}(E[street], z[ward])$	*float*	the jaccard similarity between *'vĩnh lại'* and *'vĩnh lại'*
$f_{lev}(E[street], z[ward])$	*float*	the levenshtein distance between *'vĩnh lại'* and *'vĩnh lại'*
missing z_{ma}	*binary*	missing ('ward:*'vĩnh lại'*) in E?
$REP(street, z_{ma})$	*float*	confidence when replacing (street:*'vĩnh lại'*)=(ward:*'vĩnh lại'*)

For parameter estimation, we use L-BFGS [4], a commonly-used estimation method in log-linear models. λ is a regularized hyperparameter obtained from cross-validation.

Feature extraction. We define features $\phi(x, E, z)$ to capture the relationship between the input string x with the candidate z through entities $E = \{e_i, l_i\}$. We try to map the entities in each field in candidate z and the entities E obtained by the NER stage. Table 1 shows some examples from each feature type. We have the following feature types:

- **elastic-score:** Elastic score for each field in candidate z. There are floating features.
- **entity-score:** Indicates whether a field in candidate z can be detected in the input by the NER model. These are binary features.
- **similarity-score:** The similarity between an entity obtained by NER and a field in candidate z. We use both *Levenshtein Distance* [13] and *Jaccard-similarity score* [10] to assess the similarity between two entities.
- **miss-min-adminlevel:** If a candidate is what the user intended, the field with the smallest admin level z_{ma} in candidate z should appear somehow in the input string. So these features indicate whether that field z_{ma} has been ignored by NER result.
- **replace-min-adminlevel:** Since NER is the wrongest in the field with the smallest admin level z_{ma}. We match that field by other entities e_i in E. The value of the features is calculated by the probability e_i on label l_i and the similarity between the two entities e_i and z_{ma}.

4 EXPERIMENT AND RESULTS

4.1 Datasets

We developed, hand-labeled two datasets to evaluate our CRF and log-linear model. Most of the raw address data are crawled from real-estate, e-commerce, and social networks. On *Vietnamese Address Entity Recognition Dataset (VAER Dataset)*, we have 16603 sentences: a split of 30% is reserved for a test set and the remaining sentences are used to develop the model. On *Vietnamese Address Standardization Dataset (VAS Dataset)*, because the features we use are quite light, we only need 1000 sentences to train a model. The remaining sentences are for the test set. The data is guaranteed that the sentences in the training set of the CRF model do not appear in the test set of the re-ranking model. The detail of the two datasets is shown in Table 2.

Table 2: Datasets detail (sentences)

	training set	test set
VAER dataset	12772	3831
VAS dataset	1000	7002

Table 3: Average precision, recall, and f1-score of CRF model on label level and entity level.

	precision	recall	f1-score
(a) **entity level**			
city	97.78	98.47	98.12
district	94.95	96.77	95.85
ward	90.01	84.83	87.34
street	91.88	92.43	92.15
		accuracy	
(b) **address level**		83.79	

4.2 CRF model evaluation

We train the CRF model using *VAER dataset* discussed in the previous subsection. We evaluate this model on two levels. First, at *entity level*, we report *precision, recall,* and *f1-score*. Second, at the whole *address level*, where an address is correct if the model identifies every field in that standardized address. We report *accuracy* metrics on address level.

We brief the results of our CRF model evaluation in Table 3. The model achieved good results at high admin levels but gradually reduced in the fields with low admin levels such as *ward* with F1-score of 87.34% and *street* with F1-score of 92.15%. At the address level, this model achieved 83.79% accuracy.

Table 4: The accuracy of the re-ranking test set of fuzzy-based baseline, CRF-based baseline and our system.

	accuracy
Fuzzy-based	71.32
CRF-based	84.02
Our system	**87.63**

4.3 Evaluation of address standardization

In this section, we present our evaluation on the ultimate objective: finding $y = f(x, w)|y \in w$ for a given input raw address x. We report *accuracy*, the number of *true* examples (x, y) on which the system outputs the correct answer y divided by number of examples, as main metric to evaluate our system.

We compare our system to two baselines: fuzzy-based and CRF-based. *Fuzzy-based baseline* refers to the fuzzy matching model only where the branch with the highest score in the candidate graph is selected (Subsection 3.2). *CRF-based baseline* consists of CRFs and post-processing rules. These rules aim to complete missing fields in CRF output by iterating searching over the referenced address table to produce the structured address which user had most likely intended. We use the same CRF model in the previous evaluation.

Table 4 shows the results of our system compared with two baselines on test set of *VAS dataset.* All numbers are reported in percentage accuracy. Our model achieved significantly better score than the two baselines with an *accuracy* of 87.63%.

We conducted another experiment and reported precision, recall, and f1-score on each field for each approach in Table 2. We consider an address field being recognized correctly if and only if its direct parent is also recognized correctly. The fuzzy-based approach gave good performance at high admin levels but not as good at low admin levels, due to poor ability to distinguish the fields. On the other hand, the CRF-based baseline depends heavily on the performance of the CRF model. Our system achieved a smaller number of errors at both high and low admin levels. Our system has the highest recall and average f1-score of all admin levels.

5 RELATED WORK

In the process of building address standardization with three objectives: (1) recognizing the entities mentioned in the raw input string, (2) standardizing the recognized entities, (3) fulfilling the missing address fields according to the predefined standard format, we found several studies related to our objectives. In terms of recognizing the entities, known as named entity recognition problem, most approaches are based on the Markov property. Borkar et al. [2] first proposed using a Hidden Markov Models (HMM) in address parser problem. This model was later used by Li et al. [14] to develop new models with large amounts of data. In [7], Churches et al. build an alternative model based on HMM, where the key difference is the use of the referenced address table to extract features for each token. In [18], the authors proposed a composition of CRF and a post-processing step based on learned stochastic regular grammar (SRG) that captures segment-level dependencies.

In order to reach all three objectives, Christen et al. [6] used HMM to achieve the recognizing phase and used rules to lookup recognized entities on the geocoded national address file (G-NAF) to end standardizing and fulfilling phases. In [22], Zhu et al. used an alternative CRF model to standardize an input string and query structured address on *address database* to get a map display of the input string. Recently, [17] has discussed an approach based on a multi-layer feed-forward neural network. This approach has been demonstrated by the author to give better results than the previous approaches based on HMM and CRF. However, like other probabilistic models, this approach is unlikely to be applied in places with sparse data like Vietnam.

Alternatively, fuzzy matching approaches for address standardization are researched in [3, 5, 15]. These approaches do not require a training phase but the cost of that is lower accuracy. In [5], Chatterjee et al. used an open-source platform *Apache Solr* to perform the fuzzy-query input string on the map data provided by a third party. The authors proposed a candidate graph to calculate the score of candidates obtained from the search engine, which later became the reference to our fuzzy matching block.

In recent studies, the combination of fuzzy matching and probability models has been investigated [1, 20]. However, these studies leverage probability models as a pre-processing step before the fuzzy matching phase. As a consequence, the results of the fuzzy matching process are dramatically affected by the outcomes of the probability model.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a novel approach to address standardization challenges. We demonstrated our approach for Vietnamese address and observed promising results. Our key idea is to leverage the entities detected by the NER model as a suggestion to re-rank potential address candidates obtained by the fuzzy matching stage. We develop a *log-linear model* for this re-ranking purpose. This approach solves two main problems: (1) combining the advantages of the two approaches, fuzzy matching and named entity recognition, without too much dependence on each other's outcomes. (2) our log-linear model is trained with a limited dataset. We have collected a dataset with around 8000 addresses for re-ranking, in which only 1000 addresses are used for model training. Our system achieved significantly better scores over two baselines, one solely based on the fuzzy matching process and one solely based on the NER model.

In the future, we plan to demonstrate our approach for addresses in different languages. We also investigate in handling out of known addresses. These are cases where the input string refers to an address that is outside the referenced address table and the input string does not refer to an address. Besides, we also plan to expand our approach to related problems such as *address geocoding* and in *biological named entity domain.*

ACKNOWLEDGMENTS

This research is supported by Ministry of Science and Technology (MOST) under project number ĐTĐLCN.36/19.

REFERENCES

[1] Balu Bhasuran, Gurusamy Murugesan, Sabenabanu Abdulkadhar, and Jeyakumar Natarajan. 2016. Stacked Ensemble Combined with Fuzzy Matching for Biomedical Named Entity Recognition of Diseases. *Journal of Biomedical Informatics* 64

Table 5: Error metrics on each admin level.

	fuzzy-based			CRF-based			our system		
	precision	recall	f1-score	precision	recall	f1-score	precision	recall	f1-score
city	96.10	95.96	96.03	99.52	88.40	93.63	98.70	98.56	98.63
district	92.99	92.86	92.93	98.70	87.67	92.86	97.28	97.14	97.21
ward	43.35	58.20	49.69	82.83	72.75	77.46	72.28	78.97	75.47
street	76.19	82.17	79.07	97.83	86.11	91.59	91.71	91.25	91.48
average	77.16	82.30	79.43	**94.72**	83.73	88.89	89.99	**91.48**	**90.70**

(09 2016). https://doi.org/10.1016/j.jbi.2016.09.009

[2] Vinayak Borkar, Kaustubh Deshmukh, and Sunita Sarawagi. 2003. Automatic Segmentation of Text Into Structured Records. *ACM SIGMOD Record* 30 (01 2003). https://doi.org/10.1145/376284.375682

[3] James Buckley, Bill Buckles, and Frederick Petry. 2000. Processing noisy structured textual data using a fuzzy matching approach: Application to postal address errors. *Soft Comput.* 4 (12 2000), 195–205. https://doi.org/10.1007/s005000000054

[4] Richardh Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. 2003. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing* 16 (02 2003). https://doi.org/10.1137/0916069

[5] Abhranil Chatterjee, Janit Anjaria, Sourav Roy, Arnab Ganguli, and Krishanu Seal. 2016. SAGEL: Smart Address Geocoding Engine for Supply-chain Logistics. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPACIAL '16)*. ACM, New York, NY, USA, Article 42, 10 pages. https://doi.org/10.1145/2996913.2996917

[6] Peter Christen, Tim Churches, Alan Willmore, et al. 2004. A probabilistic geocoding system based on a national address file. In *Proceedings of the 3rd Australasian Data Mining Conference, Cairns*.

[7] Tim Churches, Peter Christen, Kim Lim, and Justin Xi Zhu. 2004. Preparation of name and address data for record linkage using hidden Markov models Tim Churches. *BMC Med Inform Decis Mak* 16 (07 2004).

[8] Michael Collins. [n. d.]. Log-linear models. *Self-published Tutorial* ([n. d.]).

[9] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. *Introduction to Algorithms, Third Edition* (3rd ed.). The MIT Press.

[10] Paul Jaccard. 1901. Etude de la distribution florale dans une portion des Alpes et du Jura. *Bulletin de la Societe Vaudoise des Sciences Naturelles* 37 (01 1901), 547–579. https://doi.org/10.5169/seals-266450

[11] Sujay Kumar Jauhar, Peter Turney, and Eduard Hovy. 2016. Tables as semi-structured knowledge for question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 474–483.

[12] John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001).

[13] Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. 707–710.

[14] Xiang Li, Hakan Kardes, Xin Wang, and Ang Sun. 2014. HMM-based Address Parsing with Massive Synthetic Training Data Generation. In *Proceedings of the 4th International Workshop on Location and the Web (LocWeb '14)*. ACM, New York, NY, USA, 33–36. https://doi.org/10.1145/2663713.2664430

[15] Yanling Li, Qingwei Zhao, and Yonghong Yan. 2013. Fuzzy Matching of Semantic Class in Chinese Spoken Language Understanding. *IEICE Transactions* 96-D (2013), 1845–1852.

[16] Panupong Pasupat and Percy Liang. 2015. Compositional Semantic Parsing on Semi-Structured Tables. *CoRR* abs/1508.00305 (2015). arXiv:1508.00305 http://arxiv.org/abs/1508.00305

[17] S. Sharma, R. Ratti, I. Arora, A. Solanki, and G. Bhatt. 2018. Automated Parsing of Geographical Addresses: A Multilayer Feedforward Neural Network Based Approach. In *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*. 123–130. https://doi.org/10.1109/ICSC.2018.00026

[18] Minlue Wang, Valeriia Haberland, Amos Yeo, Andrew O. Martin, John Howroyd, and J. Mark Bishop. 2016. A Probabilistic Address Parser Using Conditional Random Fields and Stochastic Regular Grammar. *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)* (2016), 225–232.

[19] Wikipedia. [n. d.]. Inside–outside–beginning (tagging). https://en.wikipedia.org/wiki/Inside-outside-beginning_(tagging)

[20] Li Yanling and Yonghong Yan. 2014. Robust Algorithms for Semantic Class Labeling in Chinese Query Understanding .

[21] Feng Zhu, Tingting Zhao, Yang Liu, and Ying Zhao. 2018. Research on Chinese Address Resolution Model Based on Conditional Random Field. *Journal of Physics: Conference Series* 1087 (09 2018), 052040. https://doi.org/10.1088/1742-6596/1087/5/052040

[22] Feng Zhu, Tingting Zhao, Yang Liu, and Ying Zhao. 2018. Research on Chinese Address Resolution Model Based on Conditional Random Field. In *Journal of Physics: Conference Series*, Vol. 1087. IOP Publishing, 052040.