Automatic English-to-Chinese Postal Mail Address Translation System

Xiao Tu¹,², Yue Lu¹,²

¹Dept of Computer Science and Technology, East China Normal University, Shanghai 200241, China
²Shanghai Research Institute of China Post Group, Shanghai 200062, China
Background

Objective: to translation English-written address on an envelope to its corresponding Chinese

Input: scanned image

Output: translated address text

Image of Postal Mail

English-written Address of the Postal Mail
Automatic English-to-Chinese Postal Mail Address Translation System
The following is OCR results of the post mail

Professor Yue Lu
Chair, Department of
Computer
Science and Technology
East China Normal University
500 Dongehuan Road
Shanghai 200241, CHINA

The errors are unavoidable in any OCR system

Solution: using the inexact word matching technology
Chinese address may be expressed in different ways in English and sometimes mixed with homonymic Chinese Pinyin

Chinese Address: 中山北路392弄32号703室
English-written:
- Room 703, No 32, Lane 392, North zhongshan Road
- 703 shi 32 hao 392 nong Zhong Shan Bei Lu
- Unit 703 32# Long 392 Zhongshan Road (N)

No specific language rules are available

Solution: create specific language rules for address understanding
Based on the inexact word matching technology and the rules, an unstructured address text gotten from the OCR system is converted to structured address information items.

Address Information Item is a POSTCODE, a CITY, a DISTRICT, a ROAD, a ZONE, a BUILDING, NUMBERS, a COMPANY, or ADDRESSEES and so on.

Make use of ‘keyword’s like such as ‘Road’, ‘Room’, ‘Building’ and ‘University’ in an address.
## Address Understanding

### POS Tagging and Disambiguation

<table>
<thead>
<tr>
<th>POS Tagset</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CityKeyword (CK)</td>
<td>CityName (CN)</td>
</tr>
<tr>
<td>DistrictKeyword (DK)</td>
<td>DistrictName (DN)</td>
</tr>
<tr>
<td>RoadKeyword (RK)</td>
<td>OrientalKeyword (OL)</td>
</tr>
<tr>
<td>BuildingKeyword (BK)</td>
<td>ZoneKeyword (ZK)</td>
</tr>
<tr>
<td>CompanyKeyword (CPK)</td>
<td>TitleKeyword (TK)</td>
</tr>
<tr>
<td>FloorKeyword (FK)</td>
<td>Postcode (PC)</td>
</tr>
<tr>
<td>Number (NB)</td>
<td>Punctuation (PN)</td>
</tr>
<tr>
<td>LetterString (LS)</td>
<td>NoKeyword (NK)</td>
</tr>
</tbody>
</table>
Inexact word matching method

For a string A of length m and a string B of length n, V(i, j) is defined as the similarity value of the prefixes \([a_1, a_2, \ldots, a_i]\) and \([b_1, b_2, \ldots, b_j]\). The similarity of A and B is precisely the value V(m,n). The similarity of two strings A and B can be computed by dynamic programming with recurrence.

The base conditions are:  \(V(i,j) = 0, \ 0 \leq i \leq m, \ 0 \leq j \leq n\)

The general recurrence relation is:  \(\text{For } 1 \leq i \leq m, \ 1 \leq j \leq n\)

\[
V(i, j) = \max \begin{cases} V(i - 1, j - 1) + \sigma(a_i, b_j) \\ V(i - 1, j) \\ V(i, j - 1) \end{cases}
\]

Here, the matching score between the two letters \(a_i\) and \(b_j\) is defined as

\[
\sigma(a_i, b_j) = \begin{cases} 2 & a_i = b_j \\ -2 & a_i \neq b_j \end{cases}
\]
Inexact word matching method

We define the similarity between A and B as

$$Sim(A, B) = \frac{V(m, n)}{\tilde{V}_A}$$

$$\tilde{V}_A = m \times \sigma(a_i, a_i) = 2m$$ is the similarity between A and itself.

If $Sim(A, B) \geq \theta$, B matches A, or B is equal to A.

$\theta$ is a predefined threshold.
In the disambiguation rules, a word’s POS depends on the POSs of both its previous word and post word.

We represent the rule as:

<Rule No><Word><[Condition1], [Condition1’], POS1>[Condition2>, [Condition2’], POS2>…<[Condition m], [Condition m’], POSm><POS_0>.

- [Conditionx] is a subset of the tagset, denoted by POS1| POS2|…| POSk (a “|” is a logic or).
- If the previous word’s POS is in the [Conditionx] and the post word’s POS is in the [Conditionx’], the current word is POS_x.
- If not, consider the couple of conditions in the next pair of angle brackets.
- If all the couples of conditions fail, then the current word is POS_0.
Address Understanding

Address Information Items

An unstructured address text is converted to nine address information items by Deterministic Finite Automata (DFA).

A ROAD consists of three parts, viz. a prefix, a road keyword and a suffix.

Take ‘Room 301 No 329 Nan Jing Road West Shanghai 200031 China’ as an example of an OCRd address. ‘Road’ is a RK (road keyword), ‘Jing Nan 329 No 301 Room’ is $S_1$ in reserve order, and ‘West shanghai 200031 China’ is $S_2$. DFA1 accepted ‘Jing Nan’ as the longest subsequence and DFA2 accepted ‘West’ as the longest one. Hence, a ‘Road’ is ‘Nan Jing Road West’.
Address Translation

- $\text{Addr}X (\text{Sec}_1, \text{Sec}_2, \ldots, \text{Sec}_9)$ denotes the address understanding result of an address, where $\text{Sec}_j (1 \leq j \leq 9)$ denotes a postcode, a city name, a district name, a road name, a zone name, a building name, numbers, a company name and addressee names, respectively.

- $\text{DB}_k (\text{Item}_1, \text{Item}_2, \ldots, \text{Item}_9, \text{Clitem}_1, \text{Clitem}_2, \ldots, \text{Clitem}_9)$ denotes a record in the database, where $1 \leq k \leq N$ ($N$ is the total of the records in the database). $\text{Item}_j, (1 \leq j \leq 9)$ denotes a postcode, a city name, a district name, a road name, a zone name, a building name, numbers, a company name and addressee names, respectively. And $\text{Clitem}_j (1 \leq j \leq 9)$ is the translation text of $\text{Item}_j$. 
Address Translation

The similarity value between $AddrX$ and $DB_k$ is defined as

$$
\varphi_k(AddrX, DB_k) = \frac{\sum_{j=1}^{9} \text{Sim}(Sec_j, Item_j)}{m},
$$

Let $\varphi_i = \max_{1 \leq k \leq N} \varphi_k(AddrsX, DB_k)$

If $\varphi_i \geq \lambda$, $AddrX$ matches $DB_i$ and $DB_i(Clitem_1, Clitem_2, \ldots, Clitem_9)$ is the translation result of $AddrX$.

$\lambda$ denotes a predefined address matching threshold.
We implemented the presented system with C++ on Windows 2000. The experiments with the proposed method were carried on 10,000 real envelop images. We respectively selected 1.00, 0.95, 0.90, 0.85 and 0.80 as the value of the address matching threshold. The experiment results showed that 0.80 is the best one with high translation rate and low error rate.
The similarity is $\varphi_i = 0.9515 \geq \lambda$, so $AddrX$ and $DB_i$ are matched.
System Implementation

DB_i’s CItems

<table>
<thead>
<tr>
<th>CItems</th>
</tr>
</thead>
<tbody>
<tr>
<td>200241</td>
</tr>
<tr>
<td>上海</td>
</tr>
<tr>
<td>东川路</td>
</tr>
<tr>
<td>500</td>
</tr>
<tr>
<td>华东师范大学</td>
</tr>
<tr>
<td>计算机科学技术系</td>
</tr>
<tr>
<td>吕岳教授</td>
</tr>
</tbody>
</table>

Translation result printed on the envelop
Conclusions

- The present Automatic English-to-Chinese Postal Mail Address Translation System is a novel attempt in both machine translation and postal automation.

- To solve the influence caused by OCR errors, we propose an address translation method based on the inexact word matching technology.

- The experimental results have showed that the system has achieved a translation rate of 82% with a low error rate and proved the proposed method is effective and feasible. The system has been already put into application in Shanghai Post Office.

- In further work, entropy and relationship of each address information item will be taken into account to further improve the robustness of address understanding.
Thank you!