# A Semantic-Specific Model for Chinese Named Entity Translation

**Yufeng Chen** and **Chengqing Zong**
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
Beijing, China, 100190
{chenyf,cqzong}@nlpr.ia.ac.cn

## Abstract

We observe that (1) it is difficult to combine transliteration and meaning translation when transforming named entities (NE); and (2) there are different translation variations in NE translation, due to different semantic information. From this basis, we propose a novel semantic-specific NE translation model, which automatically incorporates the global context from corpus in order to capture substantial semantic information. The presented approach is inspired by example-based translation and realized by log-linear models, integrating monolingual context similarity model, bilingual context similarity model, and mixed language model. The experiments show that the semantic-specific model has substantially and consistently outperformed the baselines and related NE translation systems.

## 1 Introduction

Named entity (NE) translation, which transforms a name entity from source language to target language, plays a very important role in translingual language processing tasks, such as machine translation and cross-lingual information retrieval.

Generally, NE translation[1] includes transliteration and meaning translation. Recently, many researches have been devoted to NE transliteration (most person names) or NE meaning translation (organization names) individually. However, there are still two main challenges in statistical Chinese-English (*C2E*) NE translation.

(1) The combination of transliteration and meaning translation. Either transliteration or meaning translation is only a subtask of NE translation. There has been less work devoted to the combination of transliteration and meaning translation for translating NEs.

(2) The selection of NE translation variations. Segments in different NEs could be translated differently due to NEs' origins and enrich language phenomenon (Huang et al., 2005). As shown in Table 1, the same Chinese character "金" is translated into different English variations (highlighted in aligned parts).

| **Transliteration variations** |
|---|
| 金炳华 —— **Jin** Binghua |
| 金成勋 —— **Kim** Sung-Hoon |
| 何塞 华金 布伦纳 —— Jose Joa**quin** Brunner |
| 若阿金·希潘德 —— Joa**quim** Chipande |
| 马丁路德金 —— Martin Luther **King** |
| 金丸信 —— **Kane**maru Shin |
| 米斯金 —— Mis**kine** |
| 麦金托什 —— Aaron M**cin**tosh |
| 文森特·伯金 —— Vincent Bur**gen** |
| 埃尔金·杰拉辛 —— Er**gin** Celasin |
| 阿利亚夫金 —— Alyav**din** |
| 卡列伊金 —— Kalei**kin** |
| ...... |
| **Meaning translation variations** |
| 阿斯特基金 —— Astor **Fund** |
| 北京冶金学院 —— Beijing Institute of **Metallurgy** |
| ...... |

Table 1. *C2E* Translation variations of a character "金" in different instances

Furthermore, we randomly extract 100 Chinese characters from the person names of LDC2005T34 corpus, and find out all the characters have more than one translation variations. And each character has about average 7.8 translation variations. Also, (Li et al., 2004) have indicated that there is much confusion in *C2E* transliteration and Chinese NEs have much lower perplexity than English NEs.

According to the above two problems, we find that a crucial problem of *C2E* NE translation is selecting a correct syllable/word at each step,

---

[1] NE translation referred to in this paper denotes bilingual NE transformation (either transliteration or meaning translation), and meaning translation is proposed as distinct from transliteration.

unlike traditional Statistical machine translation (SMT), which mainly focuses on (word, phrase or syntax) alignment and reordering. The selection in NE translation is much related to its *semantic information*, including NE types, origins, collocations of included Chinese characters, and position-sensitive etc. We want the translation model could automatically learn the semantic information. However, this semantic information for translation is various and difficult to classify.

Given an input "卡科夫金 (Kakov**kin**)", how to identify the translation of "金"? Only selecting high probable translation across the training set is not reliable in this case. After simply comparing "卡科夫金" with the instances in Table 1, we find that the input is much relevant to "卡列伊金 (**kin**)", since both of them include "金" at the end position, and their contexts are much related (they share a common Chinese character usage mainly due to the same origin (Russia), such as "卡", "列", and "夫" etc., according to clues supplied by global context). If we only considers the left/right context of "金", "卡科夫金" would have been related to "阿利亚夫金 (**din**)" wrongly. From this view, this strongly suggests using a global context as the knowledge base for the final translation decision.

Therefore, we propose a semantic-specific NE translation model, which makes use of those related instances in the training data (defined as *global context*), to capture semantic information. The main idea is: for each input Chinese NE segment, it is assumed that its correct translation exists somewhere in the instances of the training set. What we need to do is to find out the correct answers based on semantic clues. It is achieved by selecting relevant instances, of which the semantic information is much relevant with the input. In other word, we choose those relevant instances from corpus to imitate translation. Here, semantic information is not directly learned, but is used as a bridge to measure the relevance or similarity between the input and those instances.

The proposed semantic-specific model has two advantages. Firstly, traditional translation approaches only exploit a general model to transform a source name into the target name with the same rules or distributions. Whereas our model could capture the transformation differences by measuring semantic similarity among different instances (global context). Secondly, we do not need define exact semantic labels for translation, such as various origins or NE types.

## 2 Framework

Formally, given a source (Chinese) name $C = c_1,...,c_k,...c_K$, which consists of $K$ Chinese segments, we want to find its target (English) translation $E = e_1,...e_k,...e_K$ of the highest probability. Here, it is assumed that an NE is literally translated, without insertion or deletion during the transformation. Within a probabilistic framework, a translation system produces the optimum target name, $E^*$, which yields the highest posterior probability given the source Chinese name.

$$E^* = \arg\max_{E \in \Phi_E} P(E \mid C) \qquad (1)$$

where $\Phi_E$ is the set of all possible translations for the Chinese name. In order to incorporate enrich language phenomenon of NEs (i.e. origins or other semantic information that affect NE translation) for capturing more exact translation, $P(E \mid C)$ is rewritten as:

$$P(E \mid C) = \sum_S P(E, S \mid C)$$
$$\cong \max_S P(E, S \mid C) \qquad (2)$$

where $S$ is the semantic-specific information for $C$ and $E$. Inspired by example-based machine translation model (Nagao, 1984; Sato and Nagao, 1990), we assume that certain mappings in the training set are identical with the transformation of the input NE. Thus we materialize the semantic information as a set of *C2E* mappings coming from the training set $S = s_1^K = s_1,...,s_k,...s_K$. A mapping $s_k$ is defined as a segment [2] pair $[sc_k, se_k]$, where $sc_k$ is similar to the input NE segment $c_k$ on the source side, while $se_k$ is the corresponding transformation of $sc_k$ on the target side. Such as [金, Jin], [金, din], or [基金, Fund]. Therefore,
$S = [sc_k, se_k]_1^K = \{[sc_1, se_1],...,[sc_k, se_k],...,[sc_K, se_K]\}$.
For example, given an input NE "日本松山芭蕾舞团 (Japanese Matsuyama Ballet Troupe)", one of its mapping sets would be {[日本, Japanese],[松山, Matsuyama],[芭蕾舞团, Ballet Troupe]}, or {[日本, Japanese],[松, Matsu], [山, yama],[芭蕾, Ballet], [舞团, Troupe]} and so on. Therefore, the semantic-specific translation model incorporates semantic information by finding out the most likely mappings coming

---

[2] The source side of one mapping could be a character, a word or several words. The target side of one mapping could be several syllables or words. Therefore one mapping is defined as a segment pair.

from the training set to capture the semantic structure. If the mappings are known, the translation is achieved. Thus the semantic-specific model is further derived as:

$$P(E,S\,|\,C)$$
$$= P(E,[sc_k,se_k]_1^K\,|\,C) = P(E,sc_1^K,se_1^K\,|\,C)$$
$$= P(sc_1^K\,|\,C) \times P(E,se_1^K\,|\,sc_1^K,C)$$
$$= P(sc_1^K\,|\,C) \times P(se_1^K\,|\,sc_1^K,C) \times P(E\,|\,se_1^K,sc_1^K,C)$$
$$\cong P(sc_1^K\,|\,C) \times P(se_1^K\,|\,sc_1^K,C) \times P(E)$$

$$\text{(3)}$$

where $P(sc_1^K\,|\,C)$ is the probability to segment the input $C$ into several source parts $sc_1^K$. And $P(se_1^K\,|\,sc_1^K,C)$ is used to assign preference to target segments $se_1^K$ across global context given the input and the source segments $sc_1^K$. Finally, $P(E)$ is the probability to connect the target segments as the final translation $E$. Therefore, in our semantic-specific model, the traditional NE translation problem is transferred as searching the most probable (higher semantic similarity) mappings from the training data and then constructing the final translation.

In the proposed model (Eq (3)), those features are equally weighted. However, they should be weighted differently according to their contributions. Considering the advantages of the maximum entropy model (Berger et al., 1996) to integrate different kinds of features, we use this framework to model the probability $P(E,S\,|\,C)$. Suppose that we have a set of $M$ feature functions $h_m(C,E,S)$, $m=1,...M$. For each feature function, there exists a model parameter $\lambda_m$, $m=1,...M$. The decision rule is used to choose the most probable target NE (Och and Ney, 2002):

$$(\hat{E},\hat{S}) = \arg\max_{E,S}\left\{\sum_{m=1}^{M}\lambda_m h_m(C,E,S)\right\} \quad \text{(4)}$$

Here, the feature functions $h_1^M(C,E,S)$ are modeled by the probabilities of $P(sc_1^K\,|\,C)$, $P(se_1^K\,|\,sc_1^K,C)$, and $P(E)$ respectively. Next, we discuss these three features in detail.

## 3 Feature Functions

### 3.1 Monolingual Similarity Model

The First feature $P(sc_1^K\,|\,C)$ segments the source into several related segments assumed independence.

$$h_1(C,E,S) = P(sc_1^K\,|\,c_1^K) \approx \prod_{k=1}^{K} P(sc_k\,|\,c_k) \quad \text{(5)}$$

The probability $P(sc_k\,|\,c_k)$ describes the relationship of $sc_k$ and the source NE segment $c_k$. Since $sc_k$ and $c_k$ are on the same language side, $P(sc_k\,|\,c_k)$ can be commonly measured by the frequency of $sc_k$. However, this measurement usually produces short and high frequent segments, which is not really suitable for NE translation with multiple variations.

To better estimate the distribution $P(sc_k\,|\,c_k)$, this paper proposes a much more generic model called monolingual similarity model, which captures phonetic characteristics and corpus statistics, and also removes the bias of choosing shorter segment.

$$P(sc_k\,|\,c_k)$$
$$\cong sim_l(sc_k,c_k) \times tf(sc_k) \times idf(sc_k) \times \log(|\,sc_k\,|+1) \quad \text{(6)}$$

Here we first adopt a local similarity function $sim_l(sc_k,c_k)$ to measure the relationship of the input Chinese segment $c_k$ and a possible Chinese segment $sc_k$. It is measured on literal level (shallow level based on Chinese character and phonetic similarity).

$$sim_l(sc_k,c_k) = \begin{cases} 1.0, & \text{if } sc_k = c_k \\ \frac{1}{I}\sum_{i=1}^{I}P(e_i\,|\,sc_k) \times P(e_i\,|\,c_k), & \text{otherwise} \end{cases} \quad \text{(7)}$$

If all the characters of the two segments are identical ($sc_k = c_k$), their similarity is assigned as a high score 1.0. However, many phonetically similar segments are usually translated into a same syllable, such as "肯" and "坎" could align to a same syllable "cam". So we use NE alignment result to evaluate the phonetic similarity of two segments by $\frac{1}{I}\sum_{i=1}^{I}P(e_i\,|\,sc_k) \times P(e_i\,|\,c_k)$, where $e_i$ denotes the same syllables they aligned in the training set.

On the other hand, a global concept, which is borrowed from tf×idf scheme in information retrieval (Chen et al., 2003), is used in Eq (7). Term frequency (tf) of a Chinese segment $tf(sc_k)$ denotes the number of occurrences of $sc_k$. Document frequency (df) of $sc_k$ is the number of English segments that $sc_k$ is translated to. And $idf(sc_k)$ is formulated as $\log(N\,/\,df(sc_k))$. Here, it is assumed there are totally $N$ English segments according to *C2E* NE alignment result.

Therefore, Eq (7) prefers Chinese segments that occur frequently, but rarely have different English transformations. Besides, since a longer segment has less disambiguation of its translation variations, we also favor longer Chinese segments, so that the length of a Chinese segment, i.e., $|sc_k|$, is also considered.

## 3.2 Bilingual Similarity Model

The second feature is formulated as follows:

$$h_2(C,E,S) = P(se_1^K \mid sc_1^K, c_1^K) \approx \prod_{k=1}^{K} P(se_k \mid sc_k, c_k) \quad (8)$$

The probability $P(se_k \mid sc_k, c_k)$ identifies the target segment $se_k$, of which the semantic information is consistent with the input $c_k$. This distribution estimates the bilingual similarity of $se_k$ and $c_k$, thus is formulated as follows:

$$\begin{aligned} &P(se_k \mid sc_k, c_k) \\ &\cong \sum_{sc_k \in KNN} sim_s(sc_k, c_k) y(sc_k, se_k) \end{aligned} \quad (9)$$

Here, we borrow the idea of KNN (K Nearest Neighbor) algorithm. Translation variations for each $c_k$ could be seen as different categories. To classify $c_k$ into a correct translation, we could find the instance $sc_k$ in the training set that is most semantically similar to $c_k$, and then assign the translation (category) $se_k$ of this nearest neighbor to $c_k$. Since there would be $K (K > 1)$ nearest neighbors for $c_k$, we generalize the nearest neighbor to K nearest instances of $c_k$. If the translation of $sc_k$ in the instance is $se_k$, $y(sc_k, se_k) = 1$, otherwise $y(sc_k, se_k) = 0$.

On the other hand, $sim_s(sc_k, c_k)$ measures the semantic consistency between $sc_k$ and $c_k$, which ensures the two have the same translation. Note that $sim_s(sc_k, c_k)$ is different from $sim_l(sc_k, c_k)$, which only measures the literal similarity based on characters or syllables as shown in Eq (7). Because it is difficult to measure the semantic similarity of two segments directly, we quantify their similarity in terms of their specific contexts. The context of $c_k$ is the input NE $C$, while the context of $sc_k$ is an instance $SC$ that includes $sc_k$ in the training set. For example: given an input NE "日本松山芭蕾舞团" that acts as a context, we want to find the translation of a segment "松", the segment "松" in the training data have different global contexts, such as "斯文松

(Sven**sson**)", "亚**松**森 (A**sun**cion)", and "赤松广隆 (Aka**matsu** Hirotaka)" and so on.

To address this problem, we adopt a vector space model that describes the context of $c_k$ and $sc_k$. Some notions are defined here. A term set $T = \{t_{-n}, ..., t_{-1}, t_1, ..., t_n\}$ is an orderly character set of the context of $c_k$, where $[-n, n]$ is a Character-based n-range context window for $c_k$. This term set not only represents the character set of the context, but also presents the position information of the context. The similar action is applied to $SC$ (the context of $sc_k$). Therefore, the context of $c_k$ (the input Chinese NE) and each instance that includes $sc_k$ would be transformed into vectors. For example, given a segment "松" in the input NE "日本松山芭蕾舞团", its term vector is {/s, 日, 本, 山, 芭, 蕾} when $n = 3$, "/s" denotes the start position. While "松" in the instance "赤松广隆", its vector is {/, /s, 赤, 广, 隆, /e}, where "/" denotes a valid character and "/e" represents the end position.

We don't use Boolean weighting or tf/idf conceptions as traditional information retrieval (IR) to calculate the terms' weight, due to the sparse data problem. The mutual information is adopted to calculate the weight of $t$, which expresses the relevance between the context of $c_k$ and the context of $sc_k$.

$$t_{weigh} = MI(t_C, t_{SC}) = \log \frac{p(t_C, t_{SC})}{p(t_C) \times p(t_{SC})} \quad (10)$$

After transferring the contexts into general vectors, the similarity of two vectors is measured by computing the cosine value of the angle between them. This measure, called cosine-similarity measure, has been widely used in information retrieval tasks (Baeza-Yates and Ribeiro-Neto, 1999), and is thus utilized here.

$$sim_s(sc_k, c_k) = \frac{V_C \bullet V_{SC}}{\|V_C\| \times \|V_{SC}\|} \quad (11)$$

The numerator is the inner product of two vectors. The denominator is product of the length of $V_C$ and the length of $V_{SC}$. If an instance $SC$ (including the segment $sc_k$) is much related to the input NE $C$ (including the segment $c_k$), this case suggests that the semantic similarity between $c_k$ and $sc_k$ is much high. In other words, the two probably have the same translation $se_k$. Here $sc_k$

acts as a bridge to realize the transformation from $c_k$ to $se_k$.

## 3.3 Mixed Language Model

The probability $P(E)$ in Eq (3) encodes the popularity distribution of an English NE $E$, i.e. English language model. As mentioned above, there are two transformation styles for NEs: transliteration and meaning translation. Hence the glue rules for the final result are different. Transliteration is syllable-connecting without space on the English side, such as "Matsu (松)" and "yama (山)" are connected as "Matsuyama (松山)", its language model can be defined as a syllable-based n-gram model

$$P_{LM}(E_{tl}) = \prod_{k=1}^{K} \prod_{j=1}^{J} P(e_{k,j} \mid e_{k,j-n+1}^{k,j-1}) \quad \text{(suppose}$$

there are $j$ letters in the $k$ segment). In contrast, the output of meaning translation is chained word by word with spaces, for example, "Wuyi (武夷)" and "Mountain (山)" are connected as "Wuyi Mountain", of which the language model is presented as a general word-based n-gram model $P_{LM}(E_{ts}) = \prod_{k=1}^{K} P(e_k \mid e_{k-n+1}^{k-1})$ . For some NEs (most organization names), transliteration and meaning translation coexist. Hence we denote $E_{tl}$ as the included transliteration part, while $E_{ts}$ as the meaning-translation part. Intuitively, the whole language model is estimated as follows.

$$h_3(C,E,S) = P(E) = P_{LM}(E_{tl}) \times P_{LM}(E_{ts}) \quad (12)$$

Moreover, the language models $P_{LM}(E_{tl})$ and $P_{LM}(E_{ts})$ could be further normalized for removing the bias induced by different word/syllable lengths.

## 4 Training and Search

Without Chinese word segmentation, we have to calculate every possible mapping to determine the most probable one in a large corpus, which will make the search space significantly huge. Therefore, we only measure those instances that including at least one character of the input NE. And the candidates, of which the feature values are below a threshold, are discarded.

### 4.1 ME Parameter Training

The weighting coefficients for the three features in Eq (3) can be learned from the development set via Maximum Entropy (ME) training.

One way to get the associated weighting coefficients for those log-probability-factors adopted in the model is to regard each of them as real-valued features, and then use ME framework to find their corresponding lambda values, which are just the weighting coefficients that we look for. Following (Och et al. 2002; Liu et al. 2005), we use the GIS (Generalized Iterative Scaling) algorithm (Darroch and Ratcliff, 1972) to train the model parameters $\lambda_1, \ldots \lambda_M$ of the log-linear models according to Eq (4). In practice, YAS-MET[3] package is adopted here to train the model parameters $\lambda_1, \ldots \lambda_M$. In our case, $M = 3$.

### 4.2 Search

We use a greedy search algorithm to search the translation with highest probability in the space of all possible mappings. A state in this space is a partial mapping. A transition is defined as the addition of a single mapping to the current state. Our start state is the empty translation result, where there is no selected mapping. A terminal state is a state in which no more mappings can be added to increase the probability of the current alignment. Our task is to find the terminal state with the highest probability.

We can compute *gain*, a heuristic function, to figure out a probability when adding a new mapping, which is defined as follows:

$$gain(S, s_k) = \frac{\exp[\sum_{m=1}^{M} \lambda_m h_m(C, E, S \bigcup s_k)]}{\exp[\sum_{m=1}^{M} \lambda_m h_m(C, E, S)]} \quad (13)$$

where $S \bigcup s_k$ means a single mapping $s_k$ is added to $S$. Since we have assumed that NE is literally translated in our model, there is a restriction: no overlap is allowed between the mapping $s_k$ and the mapping set $S$.

The greedy search algorithm for general log-linear models is formally described as follows:

**Input: *C* and aligned training set**
**Output: $E$, $S$**
1. Start with $S = \phi$;
2. Do for each $s_k$ and $s_k \bigcap S = \varnothing$:
   Compute $gain(S, s_k)$;
3. Terminate if $\forall s_k, gain(S, s_k) \leq 1$ or $sc_1^K$ covers all segments in $C$;
4. Add $s_k$ with the maximal $gain(S, s_k)$ to S;
5. Go to 2.

---

[3] http://www.fjoch.com/YASMET.html

The above search algorithm generates the final translation result by adding one mapping for each time.

## 5 Experiments

The training-set, testing-set, and development-set all come from Chinese-English Named Entity List v1.0 (LDC2005T34). The training-set consists of 218,172 proofread bilingual entries: 73,052 person name pairs, 76,460 location name pairs and 68,660 organization name pairs. Besides, 300 person names, 300 organization names, and 300 names of various NE types (including person names, location names and organization names) are used as three testing-sets respectively. Development-set includes 500 randomly selected name pairs of various NE types. There is no overlap between the training set, the development set and the open test sets.

Note that in the training set, the included transliterated parts and the meaning translated parts, which have been manually labeled, are trained separately. 218,172 NE pairs are split into 185,339 transliterated pairs (*TL-training set*) and 62,453 meaning translated pairs (*TS-training set*) (since transliteration and meaning translation would occur in one NE pair, so 185,339+62.453>218,172).

In the TL-training set, the Chinese name of an NE pair is transformed into a character-based sequence and its aligned English name is split into syllables, of which the split rules are described in (Jiang et al., 2007). Afterwards, GIZA++[4] tool is invoked to align characters to syllables. On the other hand, for TS-training set, the Chinese part of an NE is also treated as a character-based sequence, while the English part is regarded as a word-based sequence. The alignment between Chinese characters and English words are achieved by GIZA++ toolkit as well.

We use the recall of top-N hypotheses (Yang et al, 2008) as the evaluation metrics, and also adopt the Mean Reciprocal Rank (MRR) metric (Kantor and Voorhees, 2000), a measure that is commonly used in information retrieval, assuming there is precisely one correct answer. Each NE translation generates at most top-50 hypotheses for each input when computing MRR.

First, we will show the experimental results when setting different parameters for the semantic similarity model, which is done on the development set with equal feature weightings. We set

different ranges of the context window (the parameter $n$) to find which range could get the best performance. Figure 1 illustrates the effect of the range parameter $n$ for the final translation result (by MRR metric). From Figure 1, we could find that when n=3, the proposed model gets the best performance (MRR value=0.498). Therefore, n=3 is chosen for further study.
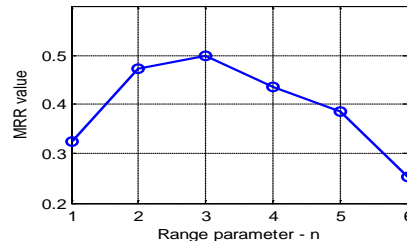


Figure 1. Effects of different context ranges ($n$) on translation results (by MRR metric)

Because the proposed three features cannot be used separately, we do not compare their individual effectiveness. Those normalized weighting coefficients (i.e., normalized lambda-values) obtained from YASMET package is 0.248, 0.565 and 0.187 (we all use 3-gram in the mixed language model). It is not surprising to find that $\lambda_2$ (corresponding to the bilingual similarity feature) receives the highest value. This clearly indicates that the bilingual similarity model plays a critical role in our semantic-specific translation model.

### 5.1 Semantic-Specific Model Vs. Baselines

We adopt a traditional statistical translation model (a phrase-based machine translation model, Moses[5] decoder) to process transliteration, meaning translation, and their combination as three baselines respectively. All of the baselines generate Top-50 candidates for each input. Table 2 shows their different settings comparing the proposed semantic-specific (SS) model.

| Setting | SS-model | Baseline I | Baseline II | Baseline III |
|---|---|---|---|---|
| Input | Un-segmented | Character-based | Word-based | Character-based |
| Training data | TL-training set + TS-training set | TL-training set | TS-training set | TL-training set + TS-training set |
| Language model | Mix of syllable-based and word-based | Syllable-based | Word-based | Word-based |

Table 2. The experiment configurations of baselines

---

[4] http://www.fjoch.com/GIZA++.html

[5] http://www.statmt.org/moses/

Note that baseline III combines transliteration and meaning translation only by training TL training set and TS training set individually, and then directly integrating generated syllable-based alignment and word-based alignment into a whole translation table.

Firstly, Table 3 compares the semantic-specific model (*SS-model*) with three baselines for the translation of person names. From Table 3, we find that the proposed model raises the recall of top-50 6.2% over Baseline I. It proves that our proposed model is effective for the transliteration of person names, and outperforms the traditional transliteration model. Baseline II can not output result due to its used TS-training set is out of the range of transliterating. It is interesting that the performance of baseline III even deteriorates after combing TS and TL training sets. One explanation might be that the language model of baseline III is only trained on word level, so that there is a severe data sparse problem.

| Metric | SS-model | Baseline I | Baseline II | Baseline III |
|--------|----------|-----------|-------------|--------------|
| Top1  | 25.6% | 17.7% | 0% | 14.2% |
| Top10 | 44.9% | 28.2% | 0% | 23.7% |
| Top50 | 62.5% | 56.3% | 0% | 39.8% |
| MRR   | 0.348 | 0.229 |  | 0.197 |

Table 3. Semantic-specific model vs. baselines for person names' translation

| Metric | SS-model | Baseline I | Baseline II | Baseline III |
|--------|----------|-----------|-------------|--------------|
| Top1  | 34.4% | 0% | 26.5% | 30.8% |
| Top10 | 38.7% | 0% | 29.8% | 36.4% |
| Top50 | 46.9% | 0% | 35.2% | 40.2% |
| MRR   | 0.381 |  | 0.297 | 0.336 |

Table 4. Semantic-specific model vs. baselines for organization names' translation

Secondly, the comparison between SS-model and three baselines for translating organization names are shown in Table 4. Baseline III outperforms baseline II for combining both TL-training set and TS-training set. Also SS-model has substantially raised the Top-N recall and MRR value over the baselines. Intuitively, we might expect that SS model could play a greater advantage on translating organization names, because organization names usually combine transliteration and meaning translation. However, comparing Table 3 with Table 4, the performance gaps between SS-model and baselines for organization names is smaller than that for person names. After checking those errors, this phenomenon is probably due to the word reordering problem, which

usually occurs in the translation of organization names, but has not been considered by SS-model. Further study would be required for this problem.

Thirdly, we measure the overall effect of SS-model in Table 5. Evidently, the proposed SS-model yields significantly better results than the three baselines at all aspects. It is not surprising to find that the proposed SS-model is effective in translating various NEs of different NE types.

| Metric | SS-model | Baseline I | Baseline II | Baseline III |
|--------|----------|-----------|-------------|--------------|
| Top1  | 30.7% | 9.5% | 11.8% | 22.4% |
| Top10 | 36.2% | 14.2% | 16.7% | 30.8% |
| Top50 | 55.3% | 23.5% | 32.8% | 42.3% |
| MRR   | 0.337 | 0.139 | 0.142 | 0.256 |

Table 5. Semantic-specific model vs. baselines for various names' translation

## 5.2 Semantic-Specific Model Vs. Joint Transliteration Model

Actually, the proposed semantic-specific model captures semantic information by incorporating the global context information in the corpus, which is similar to the joint transliteration model proposed by (Li et al., 2004). However, the joint model only utilized the local context of the input (joint n-gram model of transliteration pairs)
$$P(E \mid C) = \prod_{k=1}^{K} P(<e,c>_k \mid <e,c>_{k-n+1}^{k-1}),$$ whereas our model measures the similarity of the global context amongst corpus. Table 6 gives the comparison of the joint model and SS-model for person names' transliteration. Here previous used training-set I and 300 person names are adopted for training and testing here. Also we use 3-gram in both of the two models. As shown in Table 6, even though the performance gap of Top1 (+0.8%) is not much obvious, the performance gap gets larger when the top-N hypotheses increase. This evidently proves the superiority of the proposed model on selecting the correct translation variation from global context.

| System | Top1 | Top10 | Top50 | MRR |
|--------|------|-------|-------|-----|
| Joint model | 24.8% | 40.2% | 54.2% | 0.319 |
| SS-model | 25.6 % (+0.8%) | 43.9% (+3.7%) | 61.4% (+7.2%) | 0.348 |

Table 6. Semantic-specific model vs. joint model for person names' translation

## 5.3 Semantic-Specific Model Vs. Origin-Based Model

To further validate the capability of our proposed model, we measure its sensitivity to NE origin

information. Thus we compare it with a well-known semantic transliteration model (Li et al., 2007), which only deals with transliteration. Li's semantic transliteration model, called *origin-based model* here, firstly identifies the NE's origin $O$ by $O = \arg\max_O P(O|C)$, and then uses its corresponding trained model, which is trained on instances all from origin $O$. The training and decoding process also use the Moses decoder.

In this experiment, we adopt training-set II, which includes 7,021 person names from USA, Japan and Korea (International *whoswho* corpus in LDC2005T34). And then we randomly select 100 person names from USA, Japan and Korea respectively (also in *whoswho* corpus) as our test data. Also, there is no overlap between the training set II and those test data. Here, baseline I is also the transliteration model, but trained on training set II, and we use the MRR criterion as well.

| Test data | Baseline I | SS-model | Origin-based model |
|---|---|---|---|
| Origin=USA | 0.289 | 0.417 | 0.335 |
| Origin=Japan | 0.257 | 0.473 | 0.489 |
| Origin=Korea | 0.213 | 0.406 | 0.368 |

Table 7. Semantic-specific model vs. origin-based model for person names' translation

Considering Table 7, though there is a slight drop comparing our model with origin-based model for the Japanese person names, the translation improvements on the person names of the other two origins show the superiority of our semantic-specific translation model. Actually, there would be much more origins to classify. For instance, there are more than 100 origins in *whoswho* data; it is tedious to train a large number of models in practice. And the origin labeled data for person names is hard to acquire. By using semantic-specific model, we could directly cluster instances of similar origin, and generate final translation result for origin consistency. The experiments prove that the SS-model is effective on capturing NE origin information to assist NE translation, and it could further accommodate more different semantic information.

## 6 Related Work

There are two strategies for NE translation. One is to extract NE translation pairs from the Web or from parallel/comparable corpora. This is essentially the same as constructing NE-pair dictionary (lee et al., 2006; Jiang et al., 2009), which is usually not a real-time translation model and is limited by the coverage of the used corpus and the Web resource.

The other is to directly translate an NE phonetically or according to its meaning. For transliteration, several transliteration approaches have been applied to various language pairs (Knight and Graehl, 1998; Tsuji 2002; Li et al. 2004; Oh and Choi, 2005; Pervouchine et al., 2009; Durrani et al., 2010). In contrast, for NE meaning translation, (Zhang et al., 2005; Chen and Zong, 2008; Yang et al., 2009) have proposed different statistical translation models only for organization names.

So far, semantic transliteration has been proposed for learning language origin and gender information of person names (Li et al., 2007). However, semantic information is various for NE translation. It is complicated to define different semantic types, and is tedious to train a large number of models used for different semantic information. Moreover, a semantically labeled training corpus is hard to acquire. Hence this paper does not directly learn NE semantic information, but measures the semantic similarity between the input and global context to capture exact NE translation.

## 7 Conclusion

In this paper, we present a novel semantic-specific model which could adaptively learn semantic information via instance-based similarity measurement from global context. Accordingly, this model combines transliteration and meaning translation, and automatically selects most probable translation candidates on the basis of the NE semantic-specific information. In summary, our experiments show that the semantic-specific model is much more effective than the traditional statistical model for named entity translation, which achieves a remarkable 31.6% relative improvement in MRR (Table 5). Furthermore, the proposed model yields a comparable result with the joint transliteration model (also using context) and the origin-based model, which shows its advantage on capturing semantic information from global context, such as origin information.

It is expected that the proposed semantic-specific translation model could be further applied to other language pairs, as no language dependent linguistic feature (or knowledge) is adopted in the model/algorithm used.

## References

R. Baeza-Yates and B. RiBeiro-Neto. 1999. Modern Information Retrieval. *ISBN 0-201-39829-X*.

Adam L. Berger, Stephen A. Della Pietra and Vincent J. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39-72, March.

Hsin-His Chen, Changhua Yang and Ying Lin. 2003. Learning Formulation and Transformation Rules for Multilingual Named Entities. In *Proceedings of ACL 2003 Workshop on Multilingual and Mixed-language Named Entity Recognition*, pages 1-8.

Yufeng Chen, Chengqing Zong. 2008. A Structure-based Model for Chinese Organization Name Translation. *ACM Transactions on Asian Language Information Processing*, 7(1): 1-30, February 2008.

J. N. Darroch and D. Ratcliff. 1972. Generalized iterative scaling for log-linear models. *Annuals of Mathematical Statistics*, 43: 1470-1480.

Nadir Durrani, Hassan Sajjad, Alexander Fraser, and Helmut Schmid. 2010. Hindi-to-Urdu Machine Translation Through Transliteration. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 465–474.

Fei Huang. 2005. Cluster-Specific Name Transliteration. In *Proceedings of the HLT-EMNLP 2005*, Vancouver, BC, Canada.

Long Jiang, Ming Zhou, Lee-Feng Chien, and Cheng Niu. 2007. Named Entity Translation with Web Mining and Transliteration. In *Proceedings of IJCAI-2007*.

Long Jiang, Shiquan Yang, Ming Zhou, Xiaohua Liu, and Qingsheng Zhu. 2009. Mining Bilingual Data from the Web with Adaptively Learnt Patterns. In *Proc. of ACL-2009 and the 4th IJCNLP of the AFNLP*, pages 870-878.

Paul B. Kantor and Ellen M. Voorhees, 2000, The TREC-5 Confusion Track: Comparing Retrieval Methods for Scanned Text. *Informational Retrieval*, 2, pp. 165-176.

Kevin Knight and Jonathan Graehl. 1998. Machine Transliteration. *Computational Linguistics*, 24(4).

Chun-Jen Lee, Jason S. Chang and Jyh-Shing R. Jang. 2006. Alignment of Bilingual Named Entities in Parallel Corpora Using Statistical Models and Multiple Knowledge Sources. *ACM Transactions on Asian Language Information Processing (TALIP)*, 5(2): 121-145.

Haizhou Li, Min Zhang and Jian Su. 2004. A Joint Source Channel Model for Machine Transliteraltion. In *Proceedings of 42nd ACL*, pages 159-166.

Haizhou Li, Khe Chai Sim, Jin-shea Kuo, and Minghui Dong. 2007. Semantic Transliteration of Personal Names, In *Proceedings of 45th ACL*, pages 120-127.

Yang Liu, Qun Liu and Shouxun Lin. Log-linear Models for Word Alignment. 2005. In *Proceedings of the 43rd Annual meeting of the ACL*, pages 459-466.

M. Nagao. 1984. A Framework of a Mechanical Translation between Japanese and English by Analogy Principle, In *Artificial and Human Intelligence*, pages 173-180. NATO publications.

Franz Josef Och and Hermann Ney. 2002. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proceedings of the 40th Annual Meeting of the ACL*, pages 295-302.

J.-H. Oh and Choi, K.-S. 2005. An ensemble of grapheme and phoneme for machine transliteration. In *Proceedings of IJCNLP*, pages 450–461.

Vladimir Pervouchine, Haizhou Li and Bo Lin. 2009. Transliteration Alignment. In *Proceedings of ACL-09*, pages 136-144.

S. Sato and M. Nagao. 1990. Toward Memory-Based Translation. In *Proceedings of COLING 1990*, Vol.3. pages 247-252.

K. Tsuji. 2002. Automatic extraction of translational Japanese-KATAKANA and English word pairs from bilingual corpora. *Int. J. Comput. Process Oriental Lang*. 15(3): 261-279.

Fan Yang, Jun Zhao, Bo Zou, Kang Liu, Feifan Liu. 2008. Chinese-English Backward Transliteration Assisted with Mining Monolingual Web Pages, In *Proceeding of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 541-549, Columbus, OH.

Fan Yang, Jun Zhao, Kang Liu. 2009. A Chinese-English Organization Name Translation System Using Heuristic Web Mining and Asymmetric Alignment. In *Proceedings of the 47th Annual Meeting of the ACL*, Singapore. August 2 -7.

Min Zhang, Haizhou Li, Jian Su, and Hendra Setiawan. 2005. A Phrase-Based Context-Dependent Joint Probability Model for Named Entity Translation. *IJCNLP 2005*, pages 600-611.