Clustering and Classifying Person Names by Origin

Fei Huang Stephan Vogel Alex Waibel

Language Technologies Institute, School of Computer Sciences
Carnegie Mellon University, Pittsburgh, PA 15213
{fhuang, vogel, ahw}@cs.cmu.edu

Abstract

In natural language processing, information about a person's geographical origin is an important feature for name entity transliteration and question answering. We propose a language-independent name origin clustering and classification framework. Provided with a small amount of bilingual name translation pairs with labeled origins, we measure origin similarities based on the perplexities of name character language and translation models. We group similar origins into clusters, then train a Bayesian classifier with different features. It achieves 84% classification accuracy with source names only, and 91% with both source and target name pairs. We apply the origin clustering and classification technique to a name transliteration task. The cluster-specific transliteration model dramatically improves the transliteration accuracy from 3.8% to 55%, reducing the transliteration character error rate from 50.3 to 13.5. Adding more unlabeled name pairs to the cluster-specific name transliteration model further improves the transliteration accuracy.

Introduction

Knowing where we are from is a philosophical question carrying both historical and cultural meanings. In natural language processing, knowing a person's origin provides useful information for named entity transliteration, question answering and semantic analysis. For example, name transliteration generates a phonetically similar equivalent in the target language for a given source name, where the transliteration patterns highly depends on the person's geographical origin, e.g., the country this person is from. The following name pairs illustrate how the same Chinese character "金" is transliterated into different English letters (highlighted in different English translations), according to the origin of each person.

金 人 庆 --- **Jin** Renqing (China)
金 大 中 --- **Kim** Dae-jung (Korea)
金 丸 信 --- **Kane**maru Shin (Japan)
马丁 路德 金 --- Martin Luther **King** (USA)
何塞 华金 布伦纳 --- Jose Joa**quin** Brunner (Chile)

Traditional name transliteration approaches (Knight and Graehl, 1997, Stalls and Knight, 1998, Meng et al. 2001 and Virga and Khudanpur, 2003) exploit a general model to transliterate a source name into the target language with the same rules or distributions, which fails to capture the origin-dependent transliteration differences.

In this paper we propose a language-independent name clustering and classification framework. Considering that several origins may share the same pattern of transliteration, we would like to group these origins into clusters and build cluster-specific transliteration models. Starting from a list of bilingual name translation pairs whose origins are manually labeled, we build source character and target letter language models (LM) and translation models (TM) for each origin. We measure the similarity between origins using LM and TM perplexities, and cluster similar origins into one group. Given a source name or name translation pair, we classify it into the most likely cluster with a Bayesian classifier.

We apply the name clustering and classification technique to a name transliteration task. We train a transliteration model and a character language model for each name cluster. During transliteration, we select the most likely cluster for a given source name, then transliterate with the corresponding models under the statistical machine translation paradigm. By doing this, not only can we achieve dramatic improvement on transliteration performance, as shown in the following experiments, we can also tell the origin of a person from his/her name, which is useful for other semantic processing.

The rest of the paper is organized as follows: in section 2 we introduce the name clustering scheme, in section 3 we describe the name origin classifier. We discuss the cluster-specific name transliteration in section 4. Experiment and results are given in section 5, followed by our conclusions.

Name Origin Clustering

Provided with a list of bilingual name translation pairs, whose origins are already labeled, we want to find the origin clusters where closely related origins (countries sharing similar languages or cultural heritages) are grouped together and less related origins are apart. We consider the following factors for clustering:

- Define a similarity measure between clusters;
- Select a clustering algorithm: hierarchical clustering vs. flat clustering. If hierarchical, bottom-up or top-down clustering;
- Define the clustering termination condition: how many clusters should optimally be generated?

Assuming a generative process in creating these name translation pairs from cluster-specific models, we define the similarity measure between two clusters as their LM and TM perplexities, i.e., the probability of generating one cluster's name pairs using the other cluster's character LMs and TM. We choose bottom-up hierarchical clustering, starting with each origin as a separate cluster. Finally we select the desirable number of clusters based on source and target LM perplexities.

Define Cluster Similarity Measure

Let $S_i = \{(F_i, E_i)\}$ denote a set of name translation pairs from origin i, from which origin i's model θ_i is trained:

$$\theta_i = (P_{c(i)}, P_{e(i)}, P_{t(i)})$$

where

$P_{c(i)}$: N-gram source character LM trained from F_i;

$P_{e(i)}$: N-gram target letter LM trained from E_i;

$P_{t(i)}$: IBM-1 character translation models trained from S_i, including $P_{t(i)}(E \mid F)$, the probability of a target letter given a source character, and symmetrically $P_{t(i)}(F \mid E)$ (Brown et. al. 1993).

The distance between origin i and origin j can be symmetrically defined as:

$$d(i,j) = -\frac{1}{|S_i|}\log P(S_i \mid \theta_j) - \frac{1}{|S_j|}\log P(S_j \mid \theta_i) \cdot$$

Assuming name pairs are generated independently,

$$P(S_i \mid \theta_j) \propto \sum_{t=1}^{|S_i|} \log[P_{c(j)}(F_i^t) P_{t(j)}(E_i^t \mid F_i^t) + P_{e(j)}(E_i^t) P_{t(j)}(F_i^t \mid E_i^t)]$$

where $P(S_j \mid \theta_i)$ is defined in a similar way.

To ensure each origin has enough name pairs for reliable model training, we select *M* origins from a list of name translation pairs such that each origin has at least *c* pairs. Name pairs from the remaining origins are treated as unlabeled data for model re-training (see the following section). We calculate the pair-wise distances among these origins, and cluster them based on group-average agglomerative clustering (Manning and Schutze 1999), where the distance between clusters C_i and C_j is the average distance over all member origin pairs, defined as:

$$D(C_i, C_j) = \frac{\sum_{i \in C_i} \sum_{j \in C_j} d(i,j)}{|C_i| \times |C_j|}$$

Clustering Scheme

The group-average agglomerative clustering algorithm implements bottom-up hierarchical clustering, as follows:

- Initialize:
  - Initialize current cluster number: $m = M$;
  - Specify desirable number of clusters: n;
  - For $i = 1,...M$, $C_i = i$, i.e., each origin is a separate cluster.
- Repeat: while $m > n$
  - $\forall (i,j) \in [1,m]$, calculate $D(C_i, C_j)$;
  - if $(i', j') = \arg\min_{(i,j)} \{D(C_i, C_j)\}$, $C_{i'} = C_{i'} \cup C_{j'}$, $C_{j'} = \phi$, $m--$.

The bottom-up clustering algorithm can generate *M* different cluster partitions, ranging from the initial *M* individual origin clusters to the final single general cluster. As a result, the order of merging origins represents a clustering tree, where most similar origins are merged in the early stage and closer to leaves in the tree. If we associate each node in the tree with its merging order, every ordered node represents a clustering configuration, which indicates existing clusters at that point.

Select Optimal Cluster Number

To select the optimal number of clusters from the clustering tree, we calculate the probabilities of generating a held-out name pair list *L* from different cluster configurations, and select the one with minimum perplexity. Formally, the optimal cluster configuration

$$\omega^* = \arg\max_{\omega \in \Omega} P(L \mid \Theta_\omega);$$

where

Ω: The set of *M* clustering configurations in the tree, $\Omega = \{\omega_1, \omega_2, ..., \omega_M\}$;

Θ_t: Cluster-based LMs under configuration ω;

$$\Theta_\omega = \{\theta_j \mid j \in \omega\}$$

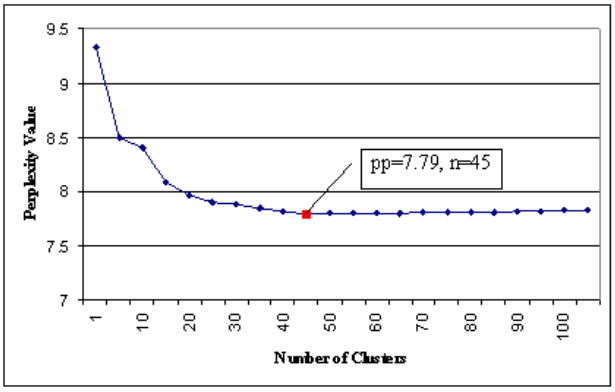

Figure 1. Perplexity value of LMs with different number of clusters

$P(L|\Theta_\omega)$: The probability of generating held-out name pairs from Θ_ω, which is the product of generating each name pair from its most likely name origin cluster:

$$P(L|\Theta_\omega) = \prod_{t=1}^{|L|} \max_{j\in\omega} P(F^t, E^t|\theta_j)P(\theta_j) = \prod_{t=1}^{|L|} \max_{j\in\omega} P_{c(j)}(F^t)P_{e(j)}(E^t)P(\theta_j) \quad (1)$$

The language model perplexity is defined as:

$$pp(L,\Theta_\omega) = 2^{-\frac{1}{|L|}\log P(L|\Theta_\omega)} = P(L|\Theta_\omega)^{-1/|L|}.$$

We clustered 56K Chinese-English name pairs from $M = 112$ origins (we set $c = 50$ in our experiments) into different numbers of clusters. We evaluate the perplexities of different cluster configurations with regard to the held-out 3K name pairs from 112 origins. Figure 1 shows the perplexities curve. As one can see, it reaches its minimum when $n = 45$. This indicates that the optimal cluster number is 45.

Table 1 lists some typical origin clusters. It can be easily seen that countries are often grouped together according to their language families. These countries are either geographically adjacent or historically affiliated. For example, while the Kazakh language belongs to the Central Turkic language family, many Kazakh names in our training data have sub-strings like "-yev", "-chenko" and "-vich", thus Kazakhstan is clustered into the Russian group. In the English group, the Netherlands (Dutch) seems an abnormity. Actually it is first merged with South Africa, which was colonized by the English and Dutch in the seventeenth century, then further clustered into this English-speaking group. Additionally, some origins cannot be merged with any other clusters because they have very unique names and translation patterns, e.g., China and Japan, and they are kept as single origin clusters.

Arabic	Afghanistan, Algeria, Egypt, Iran, Iraq, Jordan, Kuwait, Pakistan, Palestine, Saudi Arabia, Sudan, Syria, Tunisia, Yemen, …
Spanish-Portuguese	Angola, Argentina, Bolivia, Brazil, Chile Colombia, Cuba, Ecuador, Mexico, Peru Portugal, Spain, Venezuela, …
English	Australia, Canada, Netherlands, New Zealand, South Africa, UK, USA, …
Russian	Belarus, Kazakhstan, Russia, Ukraine
East European	Bosnia and Herzegovina, Croatia, Yugoslavia
French	Benin, Burkina Faso, Cameroon, Central African Republic, Congo, Gabon, Ivory Coast
German	Austria, Germany, Switzerland
French	Belgium, France, Haiti
Korean	North Korea, South Korea
Danish-Swedish	Denmark, Norway, Sweden
Single Clusters	China Japan Indonesia Israel ……

Table 1. Typical name origin clusters (n=45)

Name Origin Classification

After similar name origins are grouped into clusters, we can train an origin classifier to classify source names or name translation pairs into their most likely cluster. Identifying the source name's origin enables appropriate cluster-specific modeling for name transliteration, as presented in the next section. Also, identifying a name pair's origin helps incorporate more unlabeled training data for each cluster, which could lead to train better name classification and transliteration models.

Identify Origin Cluster with Source Names

Given a source name, we want to find the most likely cluster it is from. We use the source character language model as the classifier, and assign the name to the cluster with the highest LM probability. Assuming a source name is composed of a sequence of source characters:

$F = \{f_1, f_2, ..., f_l\}$. We want to find the cluster j^* such that

$$j^* = \arg\max_j P(\theta_j \mid F)$$
$$= \arg\max_j P(\theta_j) P(F \mid \theta_j)$$
$$= \arg\max_j P(\theta_j) P_{c(j)}(F)$$

where $P(\theta_j)$ is the prior probability of cluster *j*, and $P_{c(j)}(F)$ is the probability of generating this source name under cluster j's character-based n-gram language model.

Identify Origin Clusters with Name Translation Pairs

In addition to the bilingual name pairs whose origins are labeled for origin clustering, we have a lot more name pairs without origin labels. We want to classify these name pairs into appropriate clusters, and retrain each cluster's classification and transliteration models with augmented training data.

We adopt the metric defined in formula (1) for name pair classification. Given a name translation pair (F, E), the most likely cluster j^* is defined as:

$$j^* = \arg\max_j P(\theta_j \mid F, E)$$
$$= \arg\max_j P(F, E \mid \theta_j) P(\theta_j)$$
$$= \arg\max_j P_{c(j)}(F) P_{e(j)}(E) P(\theta_j)$$

As the character vocabulary sizes are relatively small (~30 for the English vocabulary and 6000+ for the Chinese one), we can use larger *N*'s in *N*-gram LM. A suffix array language model based on the implementation described in (Zhang and Vogel 2005) allows using arbitrary history length, and thus is a good candidate for this task.

After these unlabeled name pairs are classified into appropriate clusters, we re-train the origin classifiers with newly classified data, similar to the co-training algorithm (Blum and Mitchell, 1998). In our case, we combine the decision from two independent classifiers: source character LM (*CLM*) and target letter LM (*ELM*), and select name pairs which are confidently and consistently classified by both classifiers for model re-training.

Define the confidence measure of classifying name *F* into cluster *j* based on classifier *k*:

$$p_k(\theta_j \mid F) = \frac{P(\theta_j) P_{k(j)}(F)}{\sum_{j'} P(\theta_{j'}) P_{k(j')}(F)}$$

Define the most likely cluster based on *CLM*:

$$j_c^*(F) \equiv \arg\max_j p_{clm}(\theta_j \mid F),$$

and the most likely cluster based on *ELM*:

$$j_e^*(E) \equiv \arg\max_j p_{elm}(\theta_j \mid E).$$

The standard co-training algorithm selects name pairs satisfying

$$p_{clm}(j_c^* \mid F) > h \quad \| \quad p_{elm}(j_e^* \mid E) > h \qquad (2)$$

for classifier re-training, where *h* is the confidence score threshold (*h* = 0.9 in our experiments). A more strict constraint is:

$$\begin{cases} j_c^*(F) = j_e^*(E) \\ p_{clm}(j_c^* \mid F) > h \ \ \&\& \ \ p_{elm}(j_e^* \mid E) > h \end{cases} \qquad (3)$$

Based on these two criteria we selected unlabeled name pairs to re-train two classifiers, and we compared them with the baseline classifier, which is trained only from labeled data.

Application: Cluster-based Name Transliteration

One application of name origin classification is name transliteration, generating the translation of a source name based on its pronunciation. Rather than taking a general transliteration model and a language model to transliterate names from different origins, we train transliteration and language models for each origin cluster. Given a source name, we first classify it into the most likely cluster, then transliterate this name with the corresponding models under the statistical machine translation paradigm, where a name translation pair can be considered as a parallel sentence pair, and "words" are characters in source and target languages.

The name transliteration process can be formalized as:

$$E^* = \arg\max_E P(E \mid F, \theta_j)$$
$$= \arg\max_E P_{trl(j)}(F \mid E) P_{elm(j)}(E)$$

where *j* is the most likely cluster generating the source name *F*, $P_{trl(j)}(F \mid E)$ is the cluster *j*'s transliteration model and $P_{elm(j)}(E)$ is the target letter language model. Both transliteration and translation models are trained on cluster-specific name translation pairs whose origins are either pre-labeled or automatically classified. The language models are standard N-gram models.

The transliteration model provides a conditional distribution of target candidates for a given source transliteration unit (TU), which could be a single source character or a sequence of source characters. We use the likelihood ratio test (a hypothesis test based on two characters' co-occurrence frequency, details in (Manning and Schutze1999)) to discover these cluster-specific source TUs. We find that they are mostly names or sub-names capturing cluster-specific transliteration patterns, as shown in Table 2. It also illustrates the same source character having different transliteration candidates with different log-probabilities in different clusters.

At the training time, we first identify a character alignment path between a source and target name pair using transliteration and translation models. We segment the source name into a TU sequence, and find their translations according to the character alignment path. The transliteration probabilities are estimated based on the frequency that they are aligned over all the training name pairs. During decoding, the source name is also segmented into possible TU sequences, and their translation candidates form a monotone decoding lattice. We search for the mostly likely hypothesis within the lattice according to the cluster-specific transliteration and language models.

Arabic	穆罕默德 mohamed 阿卜杜勒 abdul 艾哈迈德 ahmed
	尤: yo (-1.29) y(-1.67) you(-1.97)
English	约翰 john 威廉 william 彼得 peter
	尤: u(-1.38) you(-1.71) joo(-1.84)
Russian	弗拉基米尔 vladimir 伊万诺夫 ivanov -耶维奇 -yevich
	尤： yu(-0.71) y(-2.58) iu(-2.63)

Table 2. Typical transliteration units (TUs) from some name clusters

Experiments

We experimented with 56K Chinese-English name translation pairs with origin labels, as well as 486K name pairs without origin labels. All the name lists were from the Linguistic Data Consortium[1] bilingual person name lists. We extracted 3K name pairs as a development set and 3K as a test set, with the same origin distribution as in the training data. As mentioned above, 56K name pairs from 112 origins were clustered into 45 origin clusters. We evaluated both name origin classification accuracies and name transliteration performances.

Name Origin Classification

We classified source names and name translation pairs using different features: source language character LM (*CLM*), target language character LM (*ELM*), and the combination of both LMs (*CELM*). We also tried *N*-gram LM with different *N*s, and select the best configuration (different *N*s for CLM and ELM). Table 3 shows the classification accuracy (%). We found that 3-gram was sufficient for the Chinese LM, while 6-gram achieved the best result for the English LM. Under these configurations, the combined CELM achieved 91.15% classification accuracy. A detailed analysis indicated that some classification errors were due to the inherent uncertainty of some names, e. g, "骆家辉 (Gary Locke)", a Chinese American, is classified as a Chinese name while his origin is labeled as USA.

We applied the name origin classifiers to the 486K name pairs without origin labels, and selected confidently classified name pairs for model re-training. We applied the standard co-training constraint (*Cot*) and a more strict constraint (*CotStr*) to select qualified name pairs (see formulae 2 and 3), and re-trained origin classifiers with the originally labeled name pairs plus additionally classified name pairs. As a result, *Cot* selected 289K name pairs for model re-training, and *CotStr* selected 83K name pairs. We compared their performances with the model (*Baseline*) trained only on the 56K labeled name pairs.

N	2	3	4	5	6	7
CLM	83.62	**84.88**	84.00	84.04	83.94	83.94
ELM	83.74	88.09	89.71	89.96	**90.10**	90.02
CELM	89.58	91.13	91.07	91.07	90.97	90.91
Best	CLM, N=3, ELM, N=6, Accuracy = **91.15**%					

Table 3. Origin classification accuracies given source name and name translation pair, using different features.

Model	**Baseline**	**Cot**	**CotStr**
CLM (N=3)	84.88	83.95	84.97
ELM (N=6)	90.10	89.00	89.87
CELM	**91.15**	**90.06**	**91.02**

Table 4. Co-training classification accuracies on dev. set

Model	**Baseline**	**Cot**	**CotStr**
CLM (N=3)	84.20	83.59	84.26
ELM (N=6)	89.52	89.45	89.81
CELM	**90.76**	**90.25**	**90.92**

Table 5. Co-training classification accuracies on eval. Set

Table 4 and 5 list the classification accuracies on the development and test set. As we can see, classifiers trained with the standard co-training constraint (Cot) consistently had lower classification accuracy than the baseline classifiers, while CotStr achieved comparable or even better performance. One possible reason is that Cot aggressively adds misclassified name pairs, which misleads the baseline classifiers.

Name Transliteration

We evaluated the effectiveness of name origin clustering in terms of name transliteration performance. We compared transliterations under three conditions:

- A traditional general model (*General*): that is, 56K name pairs from all clusters were merged to train a general transliteration model and language model.

[1] http://www.ldc.upenn.edu/

- Cluster-specific transliteration and language models trained for each cluster, using the 56K labeled name pairs (*C56K*). Given a source name, we used the source name origin classifier to identify its most likely cluster, then transliterated with the corresponding models.
- Cluster-specific transliteration and language models trained with additionally classified 486K name pairs, thus providing training on 542K name pairs total (*C542K*).

We evaluated the transliteration performance based on three metrics:

- Top 1 accuracy (*Top1*), the percentage that the top1 hypothesis was the same as the human transliteration;
- Top 5 accuracy (*Top5*), the percentage that the reference target name appeared in the generated top 5 hypotheses;
- Character error rate (*CER*), the percentage of incorrect characters (inserted, deleted, and substituted English letters) when the top 1 transliteration hypothesis was aligned to the reference name.

Table 6 shows the results. We found that the traditional general model achieved similar performance as reported in (Virga and Khudanpur), with about 50% character error rate. The cluster-based transliteration model dramatically improved the transliteration performance over the baseline, reducing the CER from 50.29% to 13.54 and increasing the top 1 accuracy from 3.78% to 55.08%. Furthermore, adding unlabeled name pairs into the most likely clusters also seemed to improve the transliteration accuracies, although the improvement has not been proved to be statistically significant.

Model	Top1 (%)	Top5 (%)	CER (%)
General	3.78±0.69	5.84±0.88	50.29±1.21
C56K	55.08±1.80	62.59±1.72	13.54±0.72
C542K	55.77±1.78	63.34±1.68	13.21±0.70

Table 6. General and cluster-specific transliteration

Conclusion and Future Work

We proposed a language-independent name origin clustering and classification framework. We clustered similar name origins into clusters according to language and translation model perplexities. Then based on cluster-specific language models, we classified a given source name or name pair into the most likely cluster with a Bayesian classifier. We applied such origin clustering and classification techniques to a name transliteration task, and achieved dramatic improvement over the traditional general transliteration model.

We hope to extend this work with soft classification of name origins, that is, to allow multiple membership of a country in several clusters with different membership weights. This will facilitate classifying and translating names from countries such as Switzerland where several languages are spoken. We would also like to investigate other approaches to make use of unlabeled data in a more efficient way.

References

A. Blum and T. Mitchell. Combining labeled and unlabeled data with cotraining. In Proceedings of the 11th Annual Conference on Computational Learning Theory. ACM, 1998.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra and R.L. Mercer. 1993. The Mathematics of Machine Translation: Parameter Estimation. In Computational Linguistics, vol 19, number 2. pp.263-311, June, 1993.

K. Knight and J. Graehl. 1997. Machine Transliteration. Proceedings of the ACL-1997. pp.128-135, Somerset, New Jersey.

C. D. Manning and H. Schütze. 1999. Foundations of Statistical Natural Language Processing. MIT Press. Boston MA.

H. Meng, W. K. Lo, B. Chen and K. Tang. 2001. Generating Phonetic Cognates to Handle Named Entities in English-Chinese Cross-Language Spoken Document Retrieval. Proceedings of the ASRU-2001, Trento, Italy, December.2001.

B. Stalls and K. Knight. Translating Names and Technical Terms in Arabic Text. In Proceedings of the COLING/ACL Workshop on Computational Approaches to Semitic Languages. Montreal, Quebec, Canada, 1998.

P. Virga and S. Khudanpur. 2003. Transliteration of Proper Names in Cross-Lingual Information Retrieval. Proceedings of the ACL-2003 Workshop on Multi-lingual Named Entity Recognition Japan. July 2003.

Y. Zhang and S. Vogel, An Efficient Phrase-to-Phrase Alignment Model for Arbitrarily Long Phrases and Large Corpora, Proceedings of the Tenth Conference of the European Association for Machine Translation (EAMT-05), Budapest, Hungary, May, 2005.