Whose Nickname is This? Recognizing Politicians from Their Aliases

Wei-Chung Wang[†], **Hung-Chen Chen**[†], **Zhi-Kai Ji**[*], **Hui-I Hsiao**[*], **Yu-Shian Chiu**[*], **Lun-Wei Ku**[†]

[†]Academia Sinica, Taiwan
{anthonywang, hankchen, lwku}@iis.sinica.edu.tw
[*]Data Analytics Technology & Applications Research Institute,
Institute for Information Industry, Taipei, Taiwan
{kevinchi, hhsiao, samuelchiu}@iii.org.tw

Abstract

Using aliases to refer to public figures is one way to make fun of people, to express sarcasm, or even to sidestep legal issues when expressing opinions on social media. However, linking an alias back to the real name is difficult, as it entails phonemic, graphemic, and semantic challenges. In this paper, we propose a phonemic-based approach and inject semantic information to align aliases with politicians' Chinese formal names. The proposed approach creates an HMM model for each name to model its phonemes and takes into account document-level pairwise mutual information to capture the semantic relations to the alias. In this work we also introduce two new datasets consisting of 167 phonemic pairs and 279 mixed pairs of aliases and formal names. Experimental results show that the proposed approach models both phonemic and semantic information and outperforms previous work on both the phonemic and mixed datasets with the best top-1 accuracies of 0.78 and 0.59 respectively.

1 Introduction

Due to the casual nature of social media, there exist a large number of non-standard words in text expressions which make it substantially different from formal written text. It is reported in (Liu et al., 2011b) that more than four million distinct out-of-vocabulary (OOV) tokens occur in the Edinburgh Twitter corpus (Petrovic et al., 2010). This variation poses challenges for natural language processing (NLP) tasks (Sproat et al., 2001). According to (Baldwin et al., 2015), on encountering the many non-standard words, found actually to be unknown named entities, they were obliged to manually eliminate these while normalizing the dataset. Thus, we believe that named entity matching (NEM) is an important problem in noisy text analysis.

NEM is the task of matching the different alias names for entities back to their respective formal names. Many applications benefit from this technique, including name transliteration (Knight and Graehl, 1998), entity linking (Rao et al., 2013), entity clustering (Green et al., 2012), and entity identification for paraphrase mining (Barzilay and Lee, 2003).

This task is different from most in noisy text normalization. While most work on normalization includes two parts—informal word identification and word recovery—this work is different in two ways. First, we focus on the recovery of informal names of named entities, in contrast to the general typo and abbreviation recovery. Second, the identification of these aliases is a named entity recognition (NER) problem, while conventional normalization works search for every OOV words. The challenges of NEM lie in variation, which can be attributed to many factors: nicknames, acronyms, and differences in transliteration. As a result, exact string matching can yield poor results. It is reported (Peng et al., 2015) that

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/

Proceedings of the 2nd Workshop on Noisy User-generated Text,
pages 61–69, Osaka, Japan, December 11 2016.

most tokenization errors are caused by named entities in documents. It is difficult to utilize analysis tools which depend on tokenized text. While most normalization can be achieved using features within words, for example, ('u', 'you') and ('2mr', 'tomorrow'), aliases and formal names may have no graphemic or phonemic connections, for instance, ('Slick Willie', 'Bill Clinton').

In this paper, we propose a novel method to train word models for each named entity with full-connected hidden Markov models (HMMs). We create two datasets of aliases and formal name pairs: one similar-phonemic pair only dataset and one mixed dataset including both similar-phonemic and non-similar-phonemic pairs. In both datasets, our method outperforms all of the baselines, including (Peng et al., 2015). In the mixed dataset, some aliases have no connection with the formal name within words. Combining our model with the statistical PMI measurement improves performance.

Our contributions are as follows. First, we propose a new method to solve the character sequence disorder problem for the NEM task. Second, we propose a new method for pairs of aliases and formal names which contain no graphemic or phonetic similarity. Third, in our experiment, our method outperforms other baselines on both datasets.

Below, we describe in Section 2 the related work for the task and in Section 3 the phonetic similarity formula. The implementation of the word model and the proposed refined version are described in Sections 4 and 5. The dataset collection, evaluation, and error analysis are in Section 6.

2 Related Work

Entity disambiguation is not an uncommon task. Related work such as entity coreferencing (Soon et al., 2001; Ponzetto and Strube, 2006; Moosavi and Strube, 2014) identifies entities which are mentioned more than once in formal documents. The literature contains extensive discussion on the effects of using segmentation, POS taggers, and semantic role labeling, which have achieved considerably high resolution. However, most of those canonical technologies fail to provide accurate information in user-generated text. Normalization on noisy text is also a highly debated topic. Most words must be normalized in this task; this includes abbreviations, non-standard spellings, and phrases prevalent in social media. Rather than considering all parts of speech, our task focuses on aliases of name entities, especially person entities.

Among those different tasks, some useful metrics can be applied to NEM given appropriate refinements. Phonetic similarity is frequently applied in normalization tasks (Li and Liu, 2012; Han and Baldwin, 2011; Xu et al., 2015). Choudhury et al. (2007) represented both non-formal words and formal words in phonemics and word pairs are evaluated on the phonetic level using left-to-right HMM model. Peng et al. (2015) propose a name matching system with a linear SVM model featuring string match methods. They apply string matching at both the character and phonemic levels. Their experiments show that adding phonetic features does improve performance.

Another assumption is that aliases contain strong contextual similarity with normal words. Liu et al. (2012) measures the cosine similarity of the word TF-IDFs in context. Li and Liu (2014) utilize word embeddings in their work.

3 Phonetic Similarity

Most Chinese characters can serve as meaningful words. Each Chinese character contains only one syllable, and each syllable is composed of an initial, a final and a tone. *Initial* refers to the first part of the Chinese syllable, which is a consonant. *Final* refers to the second part of the syllable, which is a vowel with an optional nasal sound. While some Chinese syllables have a null initial which is denoted as $\emptyset$, each syllable has a final. Mandarin Chinese is a tonal language. There are four main tones and one neutral tone, each of which has a distinctive pitch contour. According to (Liu et al., 2011a), frequently misused similar-phonemic characters have one of following features: the same sound and same tone (SS), the same sound but different tones (SD), and similar sounds and the same tone (MS). Similar sounds can be divided into two categories: the same initial (MS-DF) or the same final (MS-DI). Generally, MS-DI is more common than MS-DF. Table 1 lists the four features and some examples. We apply this theory to our word model to create a phonetic similarity character set for each character in our dictionary.

Category	Example
SS (Same sound, Same tone)	(蔡[Tsai4], 菜[Tsai4])
SD (Same sound, Differ in tone [2] and tone [3])	(才[Tsai2], 採[Tsai3])
MS-DF (Similar sound, Differ in final [en] and final [eng])	(人[Jen2], 仍[Jeng2])
MS-DI (Similar sound, Differ in initial [b] and initial [p])	(奔[Ben1], 噴[Pen1])

Table 1: Similar-phonemic features and examples

4 Left-to-right Word Model

A modified version of the word model was proposed by Choudhury et al. (2007). The main idea is to map the formal name to its alias taking into account the graphemic and phonemic aspects. Suppose that word W_1 is composed of a sequence of characters $c_1c_2...c_{l_1}$, where c_i is a character and l_1 is the length of W_1; word W_2 is composed of $t_1t_2...t_{l_2}$, where t_j is a character and l_2 is the length of W_2 Our goal is to measure the transition probability from W_1 to W_2 given the transition probability between c_i and t_j, i and j are indexes of characters. As the problem is a sequence prediction problem, it can be modeled as follows as a left-to-right HMM (Rabiner, 1989).

Each word which consists of l characters is modeled by a word model containing $2l+2$ states. The state set includes a graphemic path with l graphemic states $G_1, G_2, ..., G_l$, a phonemic path with l phonemic states $P_1, P_2, ..., P_l$, a start state, and an end state. The start state $Start$ is appended before G_1 and P_1, and the end state End is appended after G_l and P_l; these denote the start and end of the character sequence.

Each state S, including $Start$, End, P_i and G_j, consists of transition parameters, an emission character set, and emission parameters. The transition parameters denote the probability of a transition from S to the other states. The emission character sets define the candidate characters of S and the emission parameters denote the probability of the emission characters. The implementation details of the left-to-right word model are described as follows.

4.1 Graphemic Path

We define each state G_i as a graphemic state and the path from $Start$ to End through the sequence of graphemic states as the graphemic path. For each graphemic state, we allow three types of emissions: the original character, the deleted character, and typo, which are respectively denoted as $G_i(g_i)$, $G_i(\epsilon)$, and $G_i(@)$. In Figure 1 the process is illustrated for the word "李登輝" [Li3 Deng1 Hui1].

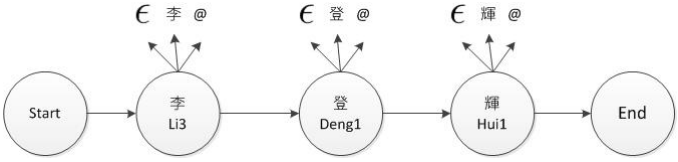

Figure 1: Graphemic path

4.2 Phonemic Path

We define l phonemic states P_1 to P_l, corresponding to the pinyin of each character p_1 to p_l. The emission set of each phonemic state is defined as follows. We construct the set according to the phonetic features of p_i. The emission set thus consists of all the pinyin characters that have the same sounds and tones (SS), the same sounds but different tones (SD), or the same finals but different initials (MS-DI). Take the character "輝" [Hui1] as example: the pinyin of "輝" is [Hui1]. Same-sound and same-tone characters include but are not limited to "輝" [Hui1], "揮" [Hui1], "灰" [Hui1], "徽" [Hui1] and "恢" [Hui1]. Table 2 lists other similar-phonemic types and examples thereof. These similar-phonemic characters compose the emission set of P_i. The emission probability of each character is set uniformly at the beginning. In Figure 2 the process is illustrated for the word "李登輝" [Li3 Deng1 Hui1].

Category	Pinyin	Example
SS	Hui1	輝、揮、灰、徽、恢
SD	Hui2	回、迴、佪、恛、茴
	Hui3	毀、悔、燬、賄、誨
	Hui4	會、惠、繪、匯、檜
MS-DI	Wei1	威、萎、薇、巍、偎
	Zhui1	追、錐、椎、隹、揣
	Cui1	催、崔、摧、墔、墔

Table 2: Phonemic emission set of 輝[Hui1]

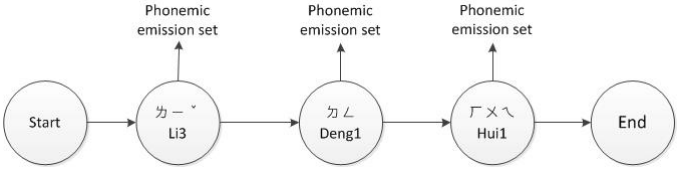

Figure 2: Phonemic path

4.3 Crosslinkages

Most aliases are created from formal names with both similar-graphemic and similar-phonemic characters. For example "李等會" [Li3 Deng3 Hui3] and "李登輝" [Li3 Teng1 Hui1]: these share as the first character the similar-graphemic "李" [Li3]. The second and third characters are similar-phonemic. In order to capture this phenomenon, we model crosslinkages between the graphemic and the phonemic paths. In Figure 3 the process is illustrated again for "李登輝" [Li3 Teng1 Hui1].

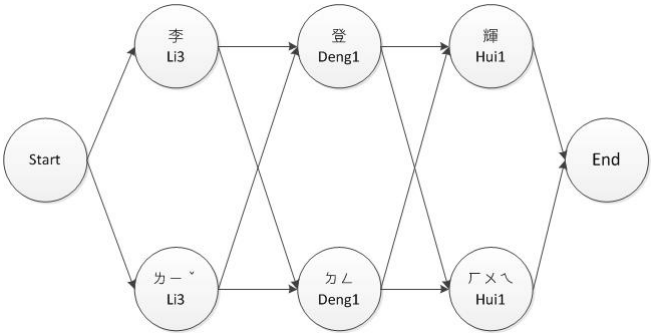

Figure 3: Graphemic and phonemic paths with crosslinks

4.4 Construction of Word HMM

Given a name w, we construct the left-to-right HMM for w as follows. First, the graphemic path is constructed. The pinyin p for w is extracted from a dictionary to construct the phonemic path. Then we create the phonemic emission set which collects the pinyin characters with the similar-phonemic of the word. We construct a left-to-right HMM for each of the 72 politician names, from w_1 to w_{72}, to create the training set. There are three model parameters, $\lambda \equiv (A, B, \pi)$, where A is the state transition probabilities, B is the emission probabilities, and π is the initial state distribution. We initialize the parameters as Choudhury et al. (2007) reported in their work. For each name w_j of the left-to-right HMM, the initial state distribution π_j^0 is defined as

$$\pi(x) = \begin{cases} 1, if X = S_0 \\ 0, otherwise \end{cases} \quad (1)$$

The emission probabilities for each graphemic state are $G_i(g_i) = 0.7$, $G_i(\epsilon) = 0.2$, $G_i(@) = 0.1$.

The emission probabilities of the phonemic states are assigned a uniform distribution. The transition probabilities for the edges from the start state are defined uniformly. For graphemic states, the transition probability to the next graphemic state is set to twice the probability of the transition to the next phonemic state. Similarly, for a phonemic state, the transition probability to another phonemic state is set to twice the probability of the transition to a graphemic state.

4.5 Training Process

Given λ_j^0, we use the Viterbi algorithm (Rabiner, 1989) to find the most likely sequences of states through the HMM for each observation (i.e., aliases in the training set) v_i^j for w_j. We keep count of every transition and emission in the HMM, which is increased by f_i^j (the frequency of the variant v_i^j) if and only if the particular transition or emission is used in the Viterbi path of v_i^j. Once the process is completed for all the variations of w_j, we re-estimate the model parameters of the HMM based on the final counts associated with each transition and emission. Add-α smoothing (Harris, 1985) is used to handle zero counts. We denote this re-estimated model for w_j by α_j^*. It is possible that we lack the training data to train word models for some politician names; for these politicians, we use the initial parameters to predict the probability.

5 Fully Connected HMM

The main disadvantage of the left-to-right HMM is that it fails on prediction when there is no sequence of similar character matches between the alias and formal name; this phenomenon is more common in this task than in canonical normalization. For example, "菊姐" [Ju2 Jie3] is the alias of "陳菊" [Chen2 Ju2]. However, for a left-to-right HMM, the model considers similarity with character mappings between "陳菊" [Chen2 Ju2] and "菊姐" [Ju2 Jie3] as (陳[Chen2], 菊[Ju2]) and (菊[Ju2], 姐[Jie3]). Even if the alias and formal name share several identical characters, simply changing their order can fool the left-to-right HMM. To account for this, we propose a fully connected HMM. We slightly alter the HMM model's parameter settings. We retain the emission parameters, which are independent of the character mapping order. / To achieve this goal, we thus alter the λ initial parameters. The transition probabilities are refined as follows. The transition probabilities from the start to the rest of the states, except the end, are distributed uniformly. For graphemic states, the transition to the next graphemic state is set to 0.5, the transition to the next phonemic state is set to 0.25, and 0.25 is distributed to the rest of the graphemic and phonemic states. Similarly, for a phonemic state, the transition probability to next phonemic state is set to 0.5, the transition to next graphemic state is set to 0.25, and 0.25 is distributed to the rest of the graphemic and phonemic states. After setting all the parameters, we train the fully connected HMM the same way we train the left-to-right HMM.

6 Experiment and Evaluation

We describe in Section 6.1 the dataset creation, and then in Section 6.2 describe the experiment settings and in Section 6.3 the evaluation.

6.1 Experiment Dataset

279 distinct alias and formal name pairs are collected from wikiptt [1].The size of formal name set, which is taken as the candidates for each alias recovery, is 72. From the collection of pairs, we created two datasets: a mixed dataset and a similar-phonemic dataset. To test the power of the phonetic features embedded in our word model, we built a similar-phonemic dataset, a subset of the mixed dataset containing 167 aliases and normal name pairs which were manually identified as bearing phonetic similarity. The mixed and similar-phonemic datasets are described in Table 3.

[1] http://zh.pttpedia.wikia.com/wiki/PTT%E6%94%BF%E6%B2%BB%E4%BA%BA%E7%89%A9%E7%B6%BD%E8%99%9F%E5%88%97%E8%A1%A8

	Similar-phonemic dataset	Mixed dataset
Train	104	194
Test	63	85
Total	167	279

Table 3: Similar-phonemic and mixed datasets

6.2 Experiment Settings

We evaluated performance on a ranking task (the setting of (Andrews et al., 2012)). We constructed as the candidates a set of normal names covering all of the entities referred to by aliases. The task is to identify for each alias the best-matched name entity in the normal name set. We acquired two sets of prior knowledge as follows. We collected 114,438 open-source character-to-pinyin mappings from OpenFoundry [2]. The list of possible initials and finals were provided by Weebly [3], a Chinese online remote learning system.

From PTT we collected 478,579 posts discussing of politicians. PTT is a well-known Chinese-language bulletin board system containing 20,000 boards covering a multitude of topics. Users post their opinions on the board. Under each post, other users can add short comments. As with most user-generated text, articles posted on PTT contain extremely short text, out-of-vocabulary tokens, and aliases. We calculated the PMI of each alias and formal name pair at the document level, and then ranked candidates according to their PMI value; the inspiration for this comes from entity clustering (Green et al., 2012), for which related entities tend to be mentioned in the same document.

The experiments were conducted on both the mixed and similar-phonemic datasets. A baseline (Baseline) was implemented based on a statistical character-level transition model. We calculated the frequency of character transitions between aliases and formal names in our training set and handled zero-count transitions using add-α smoothing. We use the pre-trained model (Mingpipe) provided by (Peng et al., 2015) as a comparable setting. The Fixed Order HMM setting denotes the basic left-to-right HMM model. To account for character sequence mismatches between alias and normal name, a simple solution is to rearrange the character order. Thus the (All sequence baseline) and (All sequence HMM) settings try all possible character sequences for each alias with brute force and report the result of the highest matching sequence. The (Fully Connected HMM) denotes a trained fully connected HMM model trained with our dataset and the (Fully connected No training) is a non-trained version. In Tables 4 we compare the rank-one prediction of each setting. The HMM + PMI Hybrid setting ranked candidates according to the average of ranks of PMI and Fully Connected HMM.

6.3 Result and Discussion

In the upper part of the Table 4, we focus on phonetically similar pairs. Applying all sequence metrics improved performance both for the baseline and the HMM model, because different aliases for same formal names tend to contain the same character but in dissimilar order. The Mingpipe model considers several string-matching metrics. It defeats our character-level statistical model but falls behind the basic left-to-right HMM model. This supports our assumption that simple string matching is not the best solution for alias recovery; applying phonetic similarity can greatly improve performance. Fully connected HMM is designed for sequence disorder pairs. It can be improved by learning from the training set, while for different aliases for the same formal name, the trained model tends to contain the same character but in dissimilar order. The trained HMM is hence likely to identify the name entity given the alias if the named entity was seen in the training set. Among the 72 candidates, our proposed model achieved 0.936 at precision @ 5, which is considerably high.

We further investigated each sample and compared our best model with other HMM settings. The Fixed Order left-to-right HMM fails for samples such as (菊姐[Ju2 Jie3], 陳菊[Chen2 Ju2]), (世間

[2] https://www.openfoundry.org/of/download_path/cnsphone2010/2016.08.02/CnsPhonetic2016-08_GCIN.cin.zip

[3] http://tzenglaoshi.weebly.com/uploads/6/9/9/2/6992410/pinyin_table.pdf

情[Shi4 Jian1 Qing2], 王世堅[Wang2 Shi4 Jian1]) and (臭青母[Chou4 Qing1 Mu3], 周美青[Zhou1 Mei3 Qing1]). Though using the All sequence HMM setting seems to solve the problem, this setting occasionally goes too far and mistakes (金貝勒[Jin1 Bei4 Le4], 金溥聰[Jin1 Pu3 Cong1]) for (金貝勒[Jin1 Bei4 Le4], 王金平[Wang2 Jin1 Ping2]).

In the lower part of the Table 4, we consider the mixed dataset of common aliases and normal name pairs. The Fully connected HMM model outperforms on the mixed set, but the training process does not yield better performance because the connections of 24 out of 85 pairs require more knowledge than our word model has. For example, "台灣之子" [Tai2 Wan1 Zhi1 Zi3, Son of Taiwan] refers to entity "陳水扁" [Chen2 Shui3 Bian3] because he was the first president of Taiwan who was also born in Taiwan. "大甲王" [Da4 Jia3 Wang2, King of TaChia] refers to entity "顏清標" [Yan2 Qing1 Biao1] because he lived in TaChia, a district located in the middle of Taiwan. These samples are corrected in the setting HMM + PMI Hybrid.

Similar-Phonemic Dataset								
	Baseline		Mingpipe	HMM				Hybrid
	Baseline	All sequence	Mingpipe	Fixed order	All sequence	Fully connected	Fully connected No training	HMM+ PMI
Pre @ 1	0.29	0.32	0.37	0.62	0.65	0.78	0.68	NAN
Mixed Dataset								
	Baseline		Mingpipe	HMM				Hybrid
	Baseline	All sequence	Mingpipe	Fixed order	All sequence	Fully connected	Fully connected No training	HMM+ PMI
Pre @ 1	0.21	0.19	0.27	0.47	0.49	0.54	0.54	0.59

Table 4: The experiment result with similar-phonemic dataset and mixed dataset

	HMM					
	Similar-phonemic dataset			Mixed dataset		
	top 3	top 5	top 10	top 3	top 5	top 10
Correct	57	59	60	59	59	59
Total	63	63	63	85	85	85
Pre @ N	0.90	0.94	0.95	0.69	0.69	0.75

Table 5: Top-N HMM results

6.4 Error Analysis

We conducted an error analysis on our best setting, hoping to provide direction for others who also are interested in alias recovery. There were 39 alias and formal name pairs which were not predicted correctly by Fully connected HMM. Most of these errors were caused by aliases which shared no graphemic and phonemic similarity with the formal entity names. See for instance ("大甲王" [Da4 Jia3 Wang2], "顏清標" [Yan2 Qing1 Biao1]). Also, some politicians' formal names share last names, especially for politicians whose sons are also politicians; for example "馬鶴凌" [Ma3 He4 Ling2], "馬英九" [Ma3 Ying1 Jiu3]. A rare special case is when name entities share the same alias, for example ("黃敏惠" [Huang2 Min3 Hui4], "惠惠" [Hui4 Hui4]) and ("翁啓惠" [Weng1 Qi3 Hui4], "惠惠" [Hui4 Hui4]). In this case our model fails to distinguish the two.

7 Conclusions

In this paper, we proposed a refined HMM for alias recovery of person entities and further improved the model with document-level PMI ranking, which has not been done previously. Both proposed models outperform previous work for both datasets. Our experiments show that considering alias recovery merely as a sequence of character matches yields poor performance.

Acknowledgements

This study is conducted under the "Big Data Technologies and Applications Project (2/4)" of the Institute for Information Industry which is subsidized by the Ministry of Economic Affairs of the Republic of China.

References

Nicholas Andrews, Jason Eisner, and Mark Dredze. 2012. Name phylogeny: A generative model of string variation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 344–355. Association for Computational Linguistics.

Timothy Baldwin, Young-Bum Kim, Marie Catherine de Marneffe, Alan Ritter, Bo Han, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. *ACL-IJCNLP*, 126:2015.

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 16–23. Association for Computational Linguistics.

Monojit Choudhury, Rahul Saraf, Vijit Jain, Animesh Mukherjee, Sudeshna Sarkar, and Anupam Basu. 2007. Investigation and modeling of the structure of texting language. *International Journal of Document Analysis and Recognition (IJDAR)*, 10(3-4):157–174.

Spence Green, Nicholas Andrews, Matthew R Gormley, Mark Dredze, and Christopher D Manning. 2012. Entity clustering across languages. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 60–69. Association for Computational Linguistics.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a# twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 368–378. Association for Computational Linguistics.

Mary Dee Harris. 1985. *Introduction to natural language processing*. Reston Publishing Co.

Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24(4):599–612.

Chen Li and Yang Liu. 2012. Normalization of text messages using character-and phone-based machine translation approaches. In *INTERSPEECH*, pages 2330–2333.

Chen Li and Yang Liu. 2014. Improving text normalization via unsupervised model and discriminative reranking. In *ACL (Student Research Workshop)*, pages 86–93.

C-L Liu, M-H Lai, K-W Tien, Y-H Chuang, S-H Wu, and C-Y Lee. 2011a. Visually and phonologically similar characters in incorrect chinese words: Analyses, identification, and applications. *ACM Transactions on Asian Language Information Processing (TALIP)*, 10(2):10.

Fei Liu, Fuliang Weng, Bingqing Wang, and Yang Liu. 2011b. Insertion, deletion, or substitution?: normalizing text messages without pre-categorization nor supervision. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 71–76. Association for Computational Linguistics.

Fei Liu, Fuliang Weng, and Xiao Jiang. 2012. A broad-coverage normalization system for social media language. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 1035–1044. Association for Computational Linguistics.

Nafise Sadat Moosavi and Michael Strube. 2014. Unsupervised coreference resolution by utilizing the most informative relations. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 644–655, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

Nanyun Peng, Mo Yu, and Mark Dredze. 2015. An empirical study of chinese name matching and applications. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 377–383, Beijing, China, July. Association for Computational Linguistics.

Sasa Petrovic, Miles Osborne, and Victor Lavrenko. 2010. The edinburgh twitter corpus. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, pages 25–26.

Simone Paolo Ponzetto and Michael Strube. 2006. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 192–199. Association for Computational Linguistics.

Lawrence R Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Delip Rao, Paul McNamee, and Mark Dredze. 2013. Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, multilingual information extraction and summarization*, pages 93–115. Springer.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4):521–544.

Richard Sproat, Alan W Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. 2001. Normalization of non-standard words. *Computer Speech & Language*, 15(3):287–333.

Ke Xu, Yunqing Xia, and Chin-Hui Lee. 2015. Tweet normalization with syllables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 920–928, Beijing, China, July. Association for Computational Linguistics.