Accurate Pinyin-English Codeswitched Language Identification

Meng Xuan Xia and **Jackie Chi Kit Cheung**
McGill University
3480 University, Rm. 318
Montreal, Quebec H3A 0E9, Canada
meng.xia@mail.mcgill.ca, jcheung@cs.mcgill.ca

Abstract

Pinyin is the most widely used romanization scheme for Mandarin Chinese. We consider the task of language identification in Pinyin-English codeswitched texts, a task that is significant because of its application to codeswitched text input. We create a codeswitched corpus by extracting and automatically labeling existing Mandarin-English codeswitched corpora. On language identification, we find that SVM produces the best result when using word-level segmentation, achieving 99.3% F1 on a Weibo dataset, while a linear-chain CRF produces the best result at the letter level, achieving 98.2% F1. We then pass the output of our models to a system that converts Pinyin back to Chinese characters to simulate codeswitched text input. Our method achieves the same level of performance as an oracle system that has perfect knowledge of token-level language identity. This result demonstrates that Pinyin identification is not the bottleneck towards developing a Chinese-English codeswitched Input Method Editor, and future work should focus on the Pinyin-to-Chinese character conversion step.

1 Introduction

As more people are connected to the Internet around the world, an increasing number of multilingual texts can be found, especially in informal, online platforms such as Twitter and Weibo[1](Ling et al., 2013). In this paper, we focus on short Mandarin-English mixed texts, in particular those that involve intra-sentential *codeswitching*, in which the two languages are interleaved within a single utterance or sentence. Example 1 shows one such case, including the original codeswitched text (CS), and its Mandarin (MAN) and English (EN) translations:

(1) CS: 这个thermal exchanger的thermal conductivity太低.
MAN: 这个换热器的热传导系数太低.
EN: The thermal conductivity of this thermal exchanger is too low.

A natural first step in processing codeswitched text is to identify which parts of the text are expressed in which language, as having an accurate codeswitched language identification system seems to be a crucial building block for further processing such as POS tagging. Recently, Solorio et al. (2014) organized the first shared task towards this goal. The task is to identify the languages in codeswitched social media data in several language pairs, including Mandarin-English (MAN-EN). Since Chinese characters are assigned a different Unicode encoding range than Latin-script languages like English, identifying MAN-EN codeswitched data is relatively straightforward. In fact, the baseline system in the shared task, which simply stores the vocabularies of the two languages seen during training, already achieves 90% F1 on identifying Mandarin segments. Most of the remaining errors are due to misclassifying English segments and named entities, which constitute a separate class in the shared task.

We focus in this paper on performing language identification between Pinyin and English,

[1]Weibo is a micro-blogging service similar to Twitter that is widely used in China.

Proceedings of the Second Workshop on Computational Approaches to Code Switching, pages 71–79,
Austin, TX, November 1, 2016.

where Pinyin is the most widely used romanization schemes for Mandarin. It is the official standard in the People's Republic of China and in Singapore. It is also the most widely used method for Mandarin speaking users to input Chinese characters using Latin-script keyboards. Example 2 shows the same codeswitched sentence, in which the Chinese characters have been converted to Pinyin:

(2) *Zhege* Thermal Exchanger *de* Thermal Conductivity *taidi*.

Distinguishing Pinyin from English or other languages written with the Roman alphabet is an important problem with strong practical motivations. Learners of both English and Chinese could benefit from a system that allows them to input codeswitched text (Chung, 2002). More generally, accurate Pinyin-English codeswitched language identification could allow users to input Mandarin-English codeswitched text more easily. A Chinese Input Method Editor (IME) system that detects Pinyin and converts it into the appropriate Chinese characters would save users from having to repeatedly toggle between the two languages when typing on a standard Latin-script keyboard.

Since Pinyin is written with the same character set as English[2], character encoding is no longer a reliable indicator of language. For example, *she*, *long*, and *bang* are Pinyin syllables that are also English words. *Tisane* is a English word, and is also a concatenation of three valid Pinyin syllables: *ti*, *sa*, and *ne*. Thus, contextual information will be needed to resolve the identity of the language.

Our contributions are as follows. First, we construct two datasets of Pinyin-English codeswitched data by converting the Chinese characters in Mandarin-English codeswitched data sets to Pinyin, and propose a new task to distinguish Pinyin from English in this codeswitched text. Then, we compare several approaches to solving this task. We consider the level of performance when training the model at the level of words versus individual letters[3] in order to see whether having word boundaries would affect performance. Two standard classification methods, SVMs and linear-chain CRFs are compared for both settings. We find that SVM produces better results on the word-level setting, achieving 99.3% F1 on a Weibo dataset. CRF produces better results on the letter-level setting, achieving 98.2% F1 on the same dataset. Lastly, we pass the output of our models to a system that converts Pinyin back to Chinese characters as an extrinsic evaluation. The result shows that word-level models produce better conversion performance. Our automatic conversion method achieves the same level of performance as an oracle system with perfect knowledge of token-level language identity. This result demonstrates that Pinyin identification is not the bottleneck towards developing a Chinese-English codeswitched IME, and that future work should focus on the Pinyin-to-Chinese character conversion step.

2 Related Work

Several models for MAN-EN codeswitched language identification were developed as part of the *First Shared Task on Language Identification in Codeswitched Data* (Chittaranjan et al., 2014; King et al., 2014). The most common technique was to employ supervised machine learning algorithms (e.g., extended Markov Models and Conditional Random Field) to train a classifier.

Codeswitched language identification has been previously studied with other language pairs, (Carter et al., 2013; Nguyen and Dogruoz, 2014; Vyas et al., 2014; Das and Gambäck, 2013; Voss et al., 2014). However, very few articles discuss codeswitched Pinyin-English input specifically. There has been research on improving the error tolerance of Pinyin-based IME. Chen and Lee (2000) propose a statistical segmentation and a trigram-based language model to convert Pinyin sequences into Chinese character sequences in a manner that is robust to single-character Pinyin misspellings. They also propose a paradigm called *modeless Pinyin* that tries to eliminate the necessity of toggling on and off the Pinyin input method. While their modeless Pinyin works on Pinyin generating a single Chinese character or a single English word, our experiments in

[2]We consider the version of Pinyin without tone diacritics, which is very common in Chinese IMEs.

[3]We chose the term letter rather than character to avoid confusion with the use of character as in Chinese characters.

this paper attempt to generate *an entire sequence* of Chinese characters and English words.

Research in improving the codeswitched text input experience also exists for other languages that use a non-alphabetic writing system, such as Japanese. Ikegami and Tsuruta (2015) propose a modeless Japanese input method that automatically switches the input mode using models with n-gram based binary classification and dictionary.

3 Task and Dataset

3.1 Task Definition

Given a Pinyin-English codeswitched input as shown in Example 2, the main task is to identify the segments of the input that are in Pinyin as *pinyin*, segments that are in English as *non-pinyin*, punctuation and whitespaces as *other* as shown in Example 3. The *other* label is used to tag tokens that do not represent actual words in both languages.

(3) Input:

Zhege␣Thermal␣Exchanger␣de␣Thermal␣Conductivity␣taidi.

Output:

Zhege␣**Thermal**␣**Exchanger**␣*de*␣**Thermal**␣**Conductivity**␣*taidi*.

Segments in bold, italic and underlined are labeled as *non-pinyin*, *pinyin* and *others*, respectively.

Separating *other* label from *non-pinyin* prevents such tokens identifiable using a simple dictionary method from artificially inflating the performance of the models during evaluation.

We do not follow the annotation scheme of the shared task in putting named entities into their own class (Solorio et al., 2014). In the Pinyin-English case, named entities clearly belong to either the *pinyin* class or the *non-pinyin* class, and Pinyin named entities would eventually need to be converted to Chinese characters in any case. Furthermore, named entity annotations are not available in the Weibo corpus that we construct.

3.2 Corpus Construction

We created two Pinyin-English codeswitched corpora automatically, by converting Mandarin-English codeswitched data. Our Mandarin-English corpora are obtained via two sources.

ENMLP We used the training data provided by the First Workshop on Computational Approaches to Code Switching of EMNLP 2014 (Solorio et al., 2014). The workshop provides a codeswitched Mandarin-English training data that contains 1000 tweets crawled from Twitter. The Chinese part of the data is in traditional Chinese.

WEIBO We downloaded 102995 Weibo entries using Cnpameng (Liang et al., 2014), a freely available Weibo crawler. Most of the entries are written in Simplified Chinese. Only a small proportion of the entries (about 1%) contain Mandarin-English codeswitched content. We removed entries that are not codeswitched and sampled 3000 of the remaining entries. In this corpus, most tokens are Chinese, with only one or two English words embedded. Chinese characters account for about 95% of the tokens.

Preprocessing and labeling The Mandarin-English codeswitching corpora are not directly usable in our experiments; we need to first convert the Chinese characters to Pinyin. We used *jieba*[4], a library for segmenting a Chinese sentence into a list of Chinese words. For each word, we used *pypinyin*[5], a Python library that converts Chinese words, both Traditional Chinese and Simplified Chinese, into Pinyin. We then label each Pinyin sequence as *pinyin*, white spaces and punctuation as *other*, and English words as *non-pinyin*, as described above.

The Mandarin-English codeswitched data we collected all contain short sentences[6]. This was by design, as we are interested in intra-sentential codeswitching. We expect that inter-sentential

[4] https://github.com/fxsjy/jieba

[5] https://github.com/mozillazg/python-pinyin

[6] Twitter and Weibo impose 140 max characters per microblog

Segmentation	Label	Count	Percentage
word-based	pinyin	2704	52.4%
	non-pinyin	716	13.8%
	others	1736	33.8%
letter-based	pinyin	12619	69.8%
	non-pinyin	3659	20.3%
	others	1777	9.9%

Table 1: Frequency count of labels on EMNLP corpus

Segmentation	Label	Count	Percentage
word-based	pinyin	16883	61.3%
	non-pinyin	1506	5.5%
	others	8970	33.2%
letter-based	pinyin	84305	84.6%
	non-pinyin	6294	6.3%
	others	9105	9.1%

Table 2: Frequency count of labels on WEIBO corpus

codeswitching would not require frequent Pinyin IME mode toggling, and labeling them for their language would also be easier.

The frequency counts of each label in the EMNLP corpus and the WEIBO corpus are shown in Tables 1 and 2, respectively.

4 Models

We propose two classes of models to solve the task: Word-Based Models and Letter-Based Models. They differ in how the input is segmented. We compared these two segmentation schemes with the goal to test whether automatic Pinyin word segmentation is needed to accurately identify Pinyin and English tokens.

Word-Based Models (WBM) The input is segmented into one of (1) a Pinyin sequence representing Chinese words[7], (2) an English word, or (3) other (space and punctuation). Each chunk is labeled as one of *pinyin*, *non-pinyin* or *other*. The Pinyin sequences representing Chinese words are indirectly segmented according to the word segmentation of the corresponding Mandarin characters. Example 3 illustrates the WBM.

Letter-Based Models (LBM) The input is segmented into individual letters. Each letter is labeled as one of *pinyin*, *non-pinyin* or *other*.

For each of the schemes above, we experimented with two discriminative supervised machine learning algorithms: Support Vector Machines (SVMs) and linear-chain Conditional Random Fields (CRFs). We chose to experiment with SVMs and CRFs in order to see whether a standard classification approach suffices or if a sequence model is needed. These methods have also been shown to perform well in previous work on language identification tasks (King and Abney, 2013; Chittaranjan et al., 2014; Lin et al., 2014; Bar and Dershowitz, 2014; Goutte et al., 2014).

4.1 Feature Extraction

We selected features to pass into our models by drawing upon recent work in codeswitched language identification (Chittaranjan et al., 2014; Lin et al., 2014; Nguyen and Dogruoz, 2014). We explored a number of different options for features, and the final set was chosen based on performance on a held-out development set, as follows.

Word-Based Models (WBM) The following features were chosen for each segment s:

- Identity of s, converted to lower case
- Whether s is a legal sequence of Pinyin
- Whether s is upper-cased
- Whether s is capitalized
- Whether s is a number
- The token occurring prior to s in the sequence
- The token occurring after to s in the sequence

Letter-Based Models (LBM) The following features were chosen for each segment t:

- Identity of t, converted to lower case
- Whether t is upper-cased
- Whether t is a number
- The token occurring prior to t in the sequence
- The token occurring after to t in the sequence

[7]Note that a Chinese word can be either a single Chinese character or a concatenation of multiple Chinese characters.

We initially experimented with several other features, but found that they did not improve performance on the development set, so we did not include them in the final system. In the WBM setting, we tried adding Part-Of-Speech (POS) tags as features, but found that existing POS taggers do not handle codeswitched data well. For both WBM and LBM, we tried to add a boolean feature to indicate whether the segment is at the start or end of the input but this turned out to be unhelpful.

4.2 Baseline Dictionary-Based Method

We compared these methods against a baseline, which labels a concatenation of valid Pinyin syllables[8] as *pinyin*, whitespaces and punctuation as *others* and the rest as *non-pinyin*.

5 Experiment 1: Language Identification

We tested our models on the two codeswitching corpora that we created. We split each corpus into training (80%) and testing (20%) subsets. We also created a held-out development set by randomly sampling 100 entries from EMNLP corpus, and used it to select the feature sets described in Section 4.1. We kept the same set of features for the WEIBO corpus, without performing any additional tuning.

We trained the CRF model using CRFsuite (Okazaki, 2007) and the SVM model using Scikit-learn (Pedregosa et al., 2011). The models were tested using commonly defined evaluation measures — Precision, Recall and F1 (Powers, 2011) at the word level for WBMs and at the letter level for LBMs.

5.1 Results

As shown in Table 3, all the WBM machine learning algorithms performed better than the baseline. The average P, R and F1 for each model were calculated without taking into account the values from the *other* label. This prevents the *other* class, which can largely be predicted by whether the segment is a whitespace or punctuation character, from artificially inflating results. The SVM-WBM model performed the best with an F1 of 0.980 on EMNLP corpus and 0.993 on WEIBO corpus. In LBM settings, only CRF outperformed the baseline with an F1 of 0.962 on the EMNLP corpus and 0.982 on the WEIBO corpus.

Note that the baseline F1 for the *other* class is not at 1.0 because baseline method's dictionary of punctuation and whitespace characters were constructed from the training set, and does not exhaustively cover all possible characters of this class.

Since there is, to our knowledge, no previous study on Pinyin-English codeswitched text input, we cannot perform direct comparison against existing work. In terms of similar tasks involving codeswitched text, the top-performing MAN-EN language identification system achieved an F1 of 0.892 (Chittaranjan et al., 2014), but the annotation scheme includes a category for named entities. Ikegami and Tsuruta (2015) achieved an F1 of 0.97) on codeswitched Japanese-English text input using an n-gram-based approach.

In the WEIBO corpus, the F1 performances of the models are very high, at up to 0.982 and 0.993. This could be because each entry in the Weibo corpus contains only one or two occurrences of single English words embedded into a sequence of Pinyin. The non-pinyin words are often proper nouns (these tokens are often capitalized), English words or acronyms that do not have a translation in Chinese. In this context, it is less common to see English words that are also valid Pinyin.

While SVM performs the best with WBM, it does not perform as well with LBM. The lower performance of SVM-LBM is caused by the limited access to contextual information (only the letter directly before and after each token). By contrast, CRF-LBM can naturally take into account the sequence ordering information. This result shows that a sequence model like CRF is needed for LBM.

[8] We used the list of 411 valid symbols available at `https://github.com/someus/Pinyin2Hanzi/blob/master/Pinyin2Hanzi/util.py#L127`

Model	label	P	R	F1
baseline	non-pinyin	0.762	0.875	0.815
	pinyin	0.962	0.950	0.956
	other	0.986	0.945	0.965
	avg / total	0.920	0.934	0.926
SVM-WBM	non-pinyin	0.967	0.944	0.956
	pinyin	0.980	0.992	0.986
	other	0.990	0.980	0.985
	avg / total	**0.977**	**0.982**	**0.980**
CRF-WBM	non-pinyin	0.948	0.919	0.934
	pinyin	0.981	0.989	0.985
	other	0.974	0.974	0.974
	avg / total	0.974	0.974	0.974

(a) WBM performance on EMNLP corpus

Model	label	P	R	F1
baseline	non-pinyin	0.865	0.880	0.872
	pinyin	0.965	0.965	0.965
	other	0.986	0.948	0.967
	avg / total	0.943	0.946	0.944
SVM-LBM	non-pinyin	0.785	0.612	0.688
	pinyin	0.893	0.953	0.922
	other	0.995	0.968	0.982
	avg / total	0.869	0.877	0.870
CRF-LBM	non-pinyin	0.930	0.902	0.916
	pinyin	0.969	0.982	0.975
	other	0.995	0.959	0.977
	avg / total	**0.960**	**0.964**	**0.962**

(b) LBM performance on EMNLP corpus

Model	label	P	R	F1
baseline	non-pinyin	0.510	0.905	0.652
	pinyin	0.990	0.948	0.969
	other	0.991	0.941	0.965
	avg / total	0.951	0.945	0.943
SVM-WBM	non-pinyin	0.973	0.948	0.960
	pinyin	0.995	0.996	0.996
	other	0.992	0.995	0.994
	avg / total	**0.993**	**0.992**	**0.993**
CRF-WBM	non-pinyin	0.963	0.913	0.937
	pinyin	0.992	0.995	0.994
	other	0.989	0.992	0.991
	avg / total	0.990	0.988	0.989

(c) WBM performance on WEIBO corpus

Model	label	P	R	F1
Baseline	non-pinyin	0.632	0.920	0.749
	pinyin	0.993	0.966	0.979
	other	0.990	0.935	0.962
	avg / total	0.968	0.963	0.963
SVM-LBM	non-pinyin	0.932	0.565	0.703
	pinyin	0.967	0.997	0.982
	other	1.000	0.987	0.993
	avg / total	0.965	0.967	0.963
CRF-LBM	non-pinyin	0.929	0.820	0.871
	pinyin	0.986	0.995	0.990
	other	0.997	0.988	0.992
	avg / total	**0.982**	**0.983**	**0.982**

(d) LBM performance on WEIBO corpus

Table 3: The performance of the models in terms of precision (P), recall (R), and F1, for each of the three classes. The avg/total row represents the average of the pinyin and non-pinyin classes, weighted by their frequencies in the dataset. We excluded the *other* category from the avg, because that class mostly consists of whitespace and punctuation.

5.2 Discussion and error analysis

We consider here the causes of the remaining errors. First, some errors are due to segments that are both legal English words and legal Pinyin sequences, as discussed in Section 4.2. For example, *you (有)*, a word that occurs both in Mandarin and English with high frequency, is difficult for our models. Having additional POS information available to the models would be helpful, as 有 is a verb in Mandarin, while *you* is a pronoun in English.

Another source of errors is the presence of mixed Pinyin, Chinese characters and English *within individual words*. These errors are often found in user names (i.e., Twitter handlers). For example:

(4) @*eggchen* 呼叫nico我完全不到你耶

The twitter handle *eggchen*, labeled as *non-pinyin* in the gold standard, is a concatenation of English word *egg* and raw Pinyin *chen*. With CRF-LBM, since the word boundary information was not available, CRF-LBM wrongly labels *chen* as *pinyin*, sep-

arated from *eggchen*.

Finally, the LBMs sometimes fail to correctly predict codeswitching boundaries. Taking an example in the dataset: "desledge" was an input sequence where "de" has gold standard label pinyin and "sledge" has gold standard label non-pinyin. In the CRF-WBM, the word boundary information is given, so the model is able to predict the labels correctly. The CRF-LEMB model predicted that the entire sequence "deseldge" is "non-pinyin".

6 Experiment 2: Converting Pinyin to Chinese characters

Next, we experimented with converting the Pinyin-English codeswitched inputs back to the original Mandarin-English sentences in an end-to-end, extrinsic evaluation. Pinyin is the most widely used method for Mandarin users to input Chinese characters using Latin-script keyboards. Improvements in the language identification step transfer over to the next step of Chinese characters generation.

Converting Pinyin to Chinese characters is not an easy task, as there are many possible Chinese characters for each Pinyin syllable. Modern Pinyin Input Methods use statistical methods to rank the possible character candidates in descending probability and predict the top-ranked candidate as the output (Chen and Lee, 2000; Zheng et al., 2011).

Task Given a Pinyin-English codeswitched input and the corresponding labels produced by our codeswitched language identification models, produce Mandarin-English codeswitched output by converting the parts labelled as Pinyin to Chinese characters.

Method We use a representative approach that models the conversion from Pinyin to Chinese characters as a Hidden Markov model (HMM), in which Chinese characters are the hidden states and Pinyin syllables are the observations. The model is trained from *SogouT* corpus (Liu et al., 2012), and the Viterbi algorithm is used to generate the final output. We used a Python implementation of this model[9] to convert *pinyin* segments to Chinese characters while leaving *others* and *non-pinyin* segments unchanged.

We use the Pinyin-English codeswitched input, paired with language identification labels from Baseline, SVM-WBM, or CRF-LBM to generate Mandarin-English codeswitched output. We then evaluated these outputs against the gold standard by measuring precision, recall, and F1 on the Chinese characters. We also compare against an oracle topline, which has perfect knowledge of the segmentation of the input into Pinyin vs non-Pinyin. For the CRF-LBM, we used the Smith–Waterman algorithm (Smith and Waterman, 1981) to align the output produced by the CRF-LBM method with the gold-standard words.

6.1 Results

Model	P	R	F1
Oracle	*0.576*	*0.576*	*0.576*
Baseline	0.511	0.576	0.541
SVM-WBM	**0.571**	**0.562**	**0.566**
CRF-LBM	0.405	0.407	0.406

Table 4: Performance of generated Mandarin-English codeswitched sentences – EMNLP Corpus

Model	P	R	F1
Oracle	*0.590*	*0.590*	*0.590*
Baseline	0.564	0.590	0.578
SVM-WBM	**0.589**	**0.590**	**0.590**
CRF-LBM	0.491	0.511	0.500

Table 5: Performance of generated Mandarin-English codeswitched sentences – WEIBO Corpus

As shown in Tables 4 and 5, with SVM-WBM, the F1 of the generated Mandarin-English codeswitched outputs are better than the baseline in both corpora for all labels, and achieves a level of performance close to the oracle method. This result shows that our contribution to accurate language identification in transliteration codeswitching pairs is able to improve the performance of Pinyin IME in codeswitching context. Furthermore, the result demonstrates

[9] *Pinyin2Hanzi*: `https://github.com/someus/Pinyin2Hanzi`.

that Pinyin identification is not the bottleneck towards developing a Chinese-English codeswitched IME, at least if word boundary information is given. Future work should focus on the Pinyin-to-Chinese character conversion step.

The higher performance of the WBM models compared to the LBM models suggests that having correct word boundaries is crucial for identifying Pinyin-English codeswitched input at higher accuracies.

Note that F1 measure of both the oracle, Baseline and SVM-WBM models are better in WEIBO corpus in comparison to EMNLP corpus. This is backed by their higher F1-measure in the language identification step.

6.2 Error analysis

There is much room for improvement in the results. Despite CRF-LBM achieving higher than Baseline F1-measure in the language identification steps, the Mandarin-English generation accuracy of CRF-LBM is lower than baseline. The sources of this lower accuracy are the following:

Presence of mixed raw pinyin. As described in Section 5.2, CRF-LBM labels the majority of raw pinyin as "pinyin". Consequently, it made the mistake of converting them to Chinese characters where they should not be.

Failure to properly predict codeswitching word boundary. An example was given previously in Section 5.2. Each failure in predicting codeswitching word boundary produces two errors, one for the first word and one for the second. This double penalty explains why despite of CRF-LBM having higher F1 than baseline in experiment 1, it is doing worse than baseline in experiment 2.

7 Conclusion

Having an accurate codeswitched language identification system serves as a crucial building block for further processing such as POS tagging. Our results on Pinyin-English codeswitched language identification experiments provide novel contributions to language identifications on transliteration pairs. We find that SVM performs the best at the word level while CRF performs the best at the letter level.

In the second experiment, we developed an automatic method that converts Pinyin-English codeswitched text to Mandarin-English text as an extrinsic evaluation of our models. We showed that word-level models produce better conversion performance. One of our automatic word-level methods achieves the same level of performance as an oracle system that has perfect knowledge of token-level language identity. This result demonstrates that Pinyin identification is not the bottleneck towards developing a Chinese-English codeswitched IME, and future work should focus on the Pinyin-to-Chinese character conversion step.

Our approach could also be considered for other languages with non-Latin-based writing systems and a corresponding romanization scheme, such as Japanese and Romaji (Krueger and Neeson, 2000).

References

Kfir Bar and Nachum Dershowitz. 2014. The Tel Aviv University System for the Code-Switching Workshop Shared Task. *EMNLP 2014*, page 139.

Simon Carter, Wouter Weerkamp, and Manos Tsagkias. 2013. Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text. *Language Resources and Evaluation*, 47(1):195–215.

Zheng Chen and Kai-Fu Lee. 2000. A new statistical approach to Chinese Pinyin input. In *Proceedings of the 38th annual meeting on association for computational linguistics*, pages 241–247. Association for Computational Linguistics.

Gokul Chittaranjan, Yogarshi Vyas, and Kalika Bali Monojit Choudhury. 2014. Word-level language identification using CRF: Code-switching shared task report of MSR India system. In *Proceedings of The First Workshop on Computational Approaches to Code Switching*, pages 73–79.

Kevin KH Chung. 2002. Effective use of Hanyu Pinyin and English translations as extra stimulus prompts on learning of Chinese characters. *Educational Psychology*, 22(2):149–164.

Amitava Das and Björn Gambäck. 2013. Code-Mixing in Social Media Text. The Last Language Identification Frontier? *Traitement Automatique des Langues (TAL): Special Issue on Social Networks and NLP, TAL*, 54(3):41–64.

Cyril Goutte, Serge Léger, and Marine Carpuat. 2014. The NRC system for discriminating similar languages. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pages 139–145.

Yukino Ikegami and Setsuo Tsuruta. 2015. Hybrid method for modeless Japanese input using N-gram based binary classification and dictionary. *Multimedia Tools and Applications*, 74(11):3933–3946.

Ben King and Steven P Abney. 2013. Labeling the Languages of Words in Mixed-Language Documents using Weakly Supervised Methods.

Levi King, Sandra Kübler, and Wallace Hooper. 2014. Word-level language identification in The Chymistry of Isaac Newton. *Digital Scholarship in the Humanities*, page fqu032.

Mark Henry Krueger and Kevin Daniel Neeson. 2000. Japanese text input method using a limited roman character set, August 1. US Patent 6,098,086.

Bin Liang, Yiqun Liu, Min Zhang, Shaoping Ma, Liyun Ru, and Kuo Zhang. 2014. Searching for people to follow in social networks. *Expert Systems with Applications*, 41(16):7455–7465.

Chu-Cheng Lin, Waleed Ammar, Lori Levin, and Chris Dyer. 2014. The CMU submission for the shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 80–86.

Wang Ling, Guang Xiang, Chris Dyer, Alan W Black, and Isabel Trancoso. 2013. Microblogs as Parallel Corpora. *ACL (1)*, pages 176–186.

Yiqun Liu, Fei Chen, Weize Kong, Huijia Yu, Min Zhang, Shaoping Ma, and Liyun Ru. 2012. Identifying web spam with the wisdom of the crowds. *ACM Transactions on the Web (TWEB)*, 6(1):2.

Dong Nguyen and A Seza Dogruoz. 2014. Word level language identification in online multilingual communication. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

Naoaki Okazaki. 2007. CRFsuite: a fast implementation of Conditional Random Fields (CRFs).

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

David Martin Powers. 2011. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.

Temple F Smith and Michael S Waterman. 1981. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, et al. 2014. Overview for the first shared task on language identification in code-switched data. *EMNLP 2014*, page 62.

Clare R Voss, Stephen Tratz, Jamal Laoudi, and Douglas M Briesch. 2014. Finding Romanized Arabic Dialect in Code-Mixed Tweets. In *LREC*, pages 2249–2253.

Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. 2014. POS Tagging of English-Hindi Code-Mixed Social Media Content. In *EMNLP*, volume 14, pages 974–979.

Yabin Zheng, Chen Li, and Maosong Sun. 2011. Chime: An efficient error-tolerant chinese pinyin input method. In *IJCAI*, volume 11, pages 2551–2556.