COMP90049 Project 1: Lexical Normalisation of Twitter Data

Yue Cao 843282

1. Introduction

Twitter is a social networking site which provides microblogging service (Kumar, Morstatter, & Liu, 2014). On account of its timeliness and scan-friendly features, Twitter has gained worldwide popularity. Massive amounts of "Tweets" generated by users contain rich information and have a huge potential for data mining. However, "Tweets" are often written in an informal style and contain a lot of abbreviations, acronyms and typos, which hinders the further analysis of Twitter data.

The purpose of this report is to implement different approximate matching techniques to conduct lexical normalisation of Twitter tokens and assess different methods' performances. Implemented approximate matching techniques include Global Edit Distance, N-Gram Distance and Refined Soundex. In the improvement section, a slang collection (Liu, Weng, & Jiang, 2012) will be used as a supplement to the original dictionary according to knowledge gained in this report.

2. Related Work

Compared with formal writing, texts like SMS or "Tweets" contain various of non-standard words (Sproat et al., 2001). Previous work on text normalization includes spelling correction, Noisy Channel Model and Machine Translation (Xue et al., 2011). Aw et al. (2006) established a Phrase-based Statistical Model, which translated SMS language to the English language as a preprocessing step for Machine Translation. Cook and Stevenson (2009) proposed an Unsupervised Noisy Channel Model, which comprised a word model based on word formation process and the frequency of word formation types. Han and Baldwin (2011) indicated most non-standard words in "Tweets" and SMS were based on morphophonemic variation. They proposed using an "OOV" (Out of Vocabulary) whitelist to improve detection of ill-formed tokens.

3. Dataset

The dataset used for this report is a sub-sample of actual Twitter data, which is curated by Han and Baldwin (2011). The labelled-tweets file (labelled-tweets.txt), which contains 545 tweets, has already been case-folded and tokenized into a labelled-tokens file (labelled-tokens.txt), which comprises 8841 tokens.

Code	Code Name	Number of tokens
OOV	**"out of vocabulary"**	**1881**
IV	"in vocabulary"	6418
NO	"not a normalisation candidate"	542

Table 1: The Code categories and the number of tokens

Each labelled token has a Code and a Canonical Form. "OOV" means the token cannot match an exact word in the vocabulary. "IV" means a token can be identified in vocabulary. And "NO" means the token contains the "#", the Twitter hashtag, or the "@" symbol, which follows a username. In this report, only the "OOV" tokens will be considered as candidates for lexical normalisation.

Another dataset is a dictionary (dict.txt) containing 234371 words. In addition, a slang collection (Slang_dict.txt) curated by Liu, Weng, and Jiang (2012) will be used. It comprises 3802 non-standard tokens along with their canonical forms. These slangs were normalised at the sentence level manually and the accuracy may subject to annotators' understanding.

4. Methodologies

4.1. Global Edit Distance (GED)

Global Edit Distance (GED), which measures the similarity between two strings, is the minimum number of edit operations to transform a string to the other (Christopher, Prabhakar, & Hinrich, 2008, p. 58). Edit operations include Insertion, Deletion and Replacement of a character in the string.

In this project, for each "OOV" token, Levenshtein Distance[1] is applied to return the best match in the dictionary (dict.txt). Each edit operation has a positive cost of 1. The less the distance, the more similar the vocabulary entry is.

In the implementation of GED, two approaches are utilized: the **first** approach returns **multiple** best matches which have the same minimum GED; the **second** approach returns a **single** prediction, which

[1] Global Edit Distance imported from editdistance Python Library (https://pypi.python.org/pypi/editdistance).

is the first one among multiple predictions in terms of their orders in the dictionary.

4.2 N-Gram Distance

N-Gram Distance[2] method measures the similarity between two strings based on the number of common substrings of length n (Kondrak, 2005). Since the GED technique may generate multiple predictions, in order to increase the precision of the matching system, a **2-Gram Distance** technique will be used to further rank multiple predictions returned by **GED** and obtain predictions with the best N-Gram Distance scores.

4.3. Phonetic Method

"Tweets" contain a lot of homophones and abbreviations, which have the same or similar pronunciations but differ in spelling. For example, "`c u 2morrow`" has similar pronunciation as "`see you tomorrow`". Hence, a Phonetic Method can be applied to convert "OOV" tokens and vocabularies into 4-character reduced forms and make comparisons (Christopher, Prabhakar, & Hinrich, 2008, p. 58).

In this project, for each "OOV" token, a **Refined Soundex**[3] algorithm will be used to identify the best match(es) from multiple predictions returned by **GED** method.

5. Evaluation

5.1. Evaluation Metrics

Three different evaluation metrics are applied to evaluate the effectiveness of different approximate matching methods listed above.

5.1.1. Accuracy

Accuracy is used to calculate the fraction of correct matches among the number of "OOV" tokens, especially for the second approach in Section 4.1. Predictions will be measured by comparing with token's Canonical Form in the labelled-token file (labelled-tokens.txt). If the two strings match, then a prediction will be identified as a "correct match".

5.1.2. Precision and Recall

In some cases, one token may have multiple best matches. Then metrics of precision and recall will be applied. **Precision** metric will be used to calculate the fraction of correct matches among the sum of multiple predictions. And **Recall** is the fraction of correct matches among the number of candidates for lexical normalisation.

5.2. Global Edit Distance

5.2.1. Results

The results for two approaches of GED are shown below.

Precision	1.19%
Recall	22.86%

Table 2: Results of multiple predictions approach

Accuracy	11.11%

Table 3: Results of single prediction approach

Token	Canonical Form	Single Prediction	Multiple Predictions
comming	coming	coaming (×)	coaming, combing, coming (√), tomming
greenbay	greenbay	greenback (×)	greenback, greenbark, greenery, greeney, greenly, greeny
cuz	because	coz (×)	coz, cub, cud, cue, cum, cup, cur, cut, guz, suz
lol	lol	col (×)	col, dol, gol, kol, lo, loa, lob, lod, lof, log, lola, loll, lolo, loo, lop, lot, lou, low, lox, loy, pol, sol, tol, vol
2	to / too	a (×)	a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z

Table 4: Examples of Global Edit Distance

5.2.2. Analysis

- Firstly, the result indicates that the Twitter data contains a lot of noise. The table below lists some examples.

No.	Noisy Categories	Examples

1	Misspelling	comming(coming)
		tomoroe(tomorrow)

2	Abbreviations	ppl(people)
		congrats(congratulations)
		bc(because)

3	Acronyms	lol(lol)
		lmao(lmao)

4	Homophones	2(to/too)
		c(see)
		b4(before)

5	Proper nouns	wal-mart(wal-mart)
		ipod(ipod)

Table 5: "Noise" analysis of "OOV" tokens

- Secondly, through analysing "OOV" tokens, it is found that 35.14% of "OOV" tokens are identical to their Canonical Forms. However, after the implementation of GED, we only get 22.86% recall. It indicates that GED wrongly matches some non-standard tokens, such as

[2] N-Gram Distance imported from ngram Python Library (https://pypi.python.org/pypi/ngram).

[3] Refined Soundex imported from pyphonetics Python Library (https://pypi.python.org/pypi/pyphonetics/0.4.1).

acronyms “lol”, whose Canonical Form is itself. Other examples include product names like “ipod”, address like “greenbay”, company names like “wal-mart” do not need to be normalised. However, these proper nouns are not included in the dictionary because languages are constantly changing and new words are not added to the dictionary.

- Thirdly, GED technique works well on misspelling words. However, for some intended misspelling, such as “sooooo” and “liveeee”, in which duplications are written on purpose to enhance the tone, GED performs ineffectively. Another counter example is that vowels are often swapped to create abbreviations. For instance, “abt” means “about”. As a result, it costs more edit operations to transform these non-standard words into the right ones. Hence, the right matches will be missed.
- Finally, for numeric homophones like “2” and “4”, GED is not applicable. In addition, “Tweets” tokens are context-based. For example, “2” could correspond to either “to” or “too”. It is difficult to predict which match is right without the boundary of original “Tweets”.

5.3. Global Edit Distance + N-Gram Distance

N-Gram technique is not applied alone because it is useless for very small alphabets. On account of that “Tweets” contains a lot of abbreviations and homophones, such as “b”(be), “c”(see), and “r”(are), the precision of the N-Gram Distance system will be low. Hence, in this report, we combine GED and 2-Gram Distance. The results are shown below.

Precision	9.73%
Recall	22.86%

Table 6: Results of GED + N-Gram Distance

Token	GED	GED+N-Gram
comming (coming)	coaming, combing, coming (√), tomming	coming (√)
lol (lol)	col, dol, gol, kol, lo, loa, lob, lod, lof, log, lola, loll, lolo, loo, lop, lot, lou, low, lox, loy, pol, sol, tol, vol	log, tol
bf (boyfriend)	b, ba, be, bo, bu, by, f, if, of	b, ba, be, bo, bu, by f, if, of

Table 7: Difference between GED and GED+N-Gram

Through the combination of two techniques, the precision of the system has improved to 9.73%. Because the 2-Gram Distance technique is useful to reduce some multiple predictions. However, for short terms like “bf”, the performance is still bad.

5.4. Global Edit Distance + Phonetic Method

According to knowledge gained in the above systems, it is indicated that many “OOV” tokens are phonetic spellings of intended words. Therefore, phonetic approximate matching algorithms can be applied to predict correct matches which have similar pronunciations with “OOV” tokens.

In this report, the Refined Soundex algorithm, which is a modification of the original Soundex algorithm, is used to make further approximate matching. Although digits like “2” and “4” are frequently used in “Tweets” to respectively represent “to/too” and “for”, tokens contains digits are not considered in the phonetic method. The results of this system are shown below.

Precision	23.73%
Recall	23.73%

Table 8: Results of multiple predictions of GED + Refined Soundex

Token	GED+N-Gram	GED+Refined Soundex
greenbay (greenbay)	greenback, greenbark, greenery, greeney, greenly, greeny	greenback (×)
cuz (because)	coz, cub, cud, cue, cum, cup, cur, cut, guz, suz	coz (×)

Table 9: Difference between GED+N-Gram and GED + Refined Soundex

The GED + Refined Soundex system significantly improved the system precision to 23.73%, the same as the system recall, which indicates the phonetic nature of lexical normalisation.

6. Improvement

According to knowledge gained in the above analysis, we can find that traditional methods work well for regular misspellings but less applicable for many slangs. Because slangs are formed in different ways: prefixion, suffixation, affixation, homonym, acronym, etc. As such, one approximate matching method is difficult to cover all the possibilities. Even the combination of multiple methods has bad performance.

For example, in Soundex system, “congress” and “congratulations” both sound alike to “congrats”.

Tokens	GED	2-Gram Distance
congrats & congratulations	7	0.47
“congrats & congress”	2	0.5

Table 10: Examples of traditional spelling correction methods

When we use GED and 2-Gram Distance to further measure the similarity between the two pairs, the results are shown above. Both methods identify that "congress" is a better match, which is wrong. Because the formation of non-standard words is different from regular misspellings.

6.1. The application of a slang dictionary

With the emerging of various "texting abbreviations" and "internet acronyms" (http://www.smart-words.org/abbreviations/text.html), many of them have evolved into frequently-used slangs. These slangs cannot be effectively transformed to formal forms by using the GED, N-Gram Distance or Phonetic Methods. Hence, a slang collection (Liu, Weng, & Jiang, 2012), which comprises 3802 common slangs with their canonical forms, is applied as a complement for the original dictionary.

Before conducting GED and Refined Soundex, the slang dictionary (Slang_dict.txt) will be used as a preprocessing step to return the exact matches for common slangs.

6.2. Results

The results of Slang dictionary + GED + Refined Soundex are shown below.

Precision	59.54%
Recall	62.53%

Table 11: Evaluation of Slang dict + GED + RS

Token	Canonical Forms	Prediction
pix	pictures	pictures (√)
def	definitely	definitely (√)
ppl	people	people (√)
wuz	what's	was (×)

Table 12: Examples of Slang dict + GED + RS

After the application of the slang dictionary, both the precision and recall are significantly improved. It indicates the unique characteristics of social media language. However, the limitation of this method is that since English is changing, a slang set needs to be constantly trained to "understand" emerging words.

In terms of context, for tokens which have multiple Canonical Forms, such as "2" could mean "to" or "too", it is challenging to identify the correct match even if we got a slang dictionary. For future improvement, these multiple Canonical Forms can be prioritised based on their frequency of occurrences in real world.

7. Conclusions

In summary, this report assesses the effectiveness of different methods for lexical normalisation of Twitter Data. However, subject to characteristics of "Tweets", such as abbreviations, acronyms and homophones, the performance of traditional spelling correction methods is limited. With a complement of a slang dictionary, the effectiveness of the system is significantly improved. Further improvements could focus on context-based lexical normalisation and conduct constantly training on Twitter data to keep the slang set up-to-date.

References

Aw, A., Zhang, M., Xiao, J., & Su, J. 2006. A phrase-based statistical model for SMS text normalization. In *Proceedings of the COLING/ACL on Main conference poster sessions*. Association for Computational Linguistics. pp. 33-40.

Christopher, D. M., Prabhakar, R., & Hinrich, S. C. H. Ü. T. Z. E. 2008. Introduction to information retrieval. *An Introduction To Information Retrieval*, p.58.

Cook, P., & Stevenson, S. 2009. An unsupervised model for text message normalization. In *Proceedings of the workshop on computational approaches to linguistic creativity*. Association for Computational Linguistics. pp. 71-78.

Han, B., & Baldwin, T. 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*. Association for Computational Linguistics.*Volume 1*, pp. 368-378.

Kondrak, G. 2005. N-gram similarity and distance. In *String processing and information retrieval*. Springer Berlin/Heidelberg. pp. 115-126.

Kumar, S., Morstatter, F., & Liu, H. 2014. *Twitter data analytics*. New York: Springer.

Liu, F., Weng, F., & Jiang, X. 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*, Association for Computational Linguistics. *Volume 1*, pp. 1035-1044.

Sproat, R., Black, A. W., Chen, S., Kumar, S., Ostendorf, M., & Richards, C. 2001. Normalization of non-standard words. *Computer speech & language*, *15*(3), 287-333.

Text Message - SMS - E-Mail - Chat. 2017.
Retrieved from
http://www.smart-words.org/abbreviations/text.html.

Xue, Z., Yin, D., Davison, B. D., & Davison, B. 2011. Normalizing Microtext. *Analyzing Microtext*, *11*, 05.