Product Name Recognition and Normalization in Internet Forums

Yangjie Yao
School of Computer Engineering
Nanyang Technological University, Singapore
yyao002@e.ntu.edu.sg

Aixin Sun
School of Computer Engineering
Nanyang Technological University, Singapore
axsun@ntu.edu.sg

ABSTRACT

Collecting users' feedback on products from Internet forums is challenging because users often mention a product with informal abbreviations or nicknames. In this paper, we propose a method named GREN to recognize and normalize mobile phone names from Internet forums. Instead of directly recognizing phone names from sentences as in most named entity recognition tasks, we propose an approach to first generating candidate names. The candidate names capture short forms, spelling variations, and nicknames of products, but are not noise free. To predict whether a candidate name mention in a sentence indeed refers to a specific phone model, a CRF-based name recognizer is developed. The CRF (Conditional Random Field) model is trained by using a large set of sentences obtained in a semi-automatic manner with minimal manual labeling effort. Lastly, a rule-based name normalization component maps a recognized name to its formal form. Evaluated on more than 4000 manually labeled sentences with about 1000 phone name mentions, GREN achieves precision and recall of 0.918 and 0.875 respectively, with the best feature setting.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Linguistic processing*

Keywords

Mobile phone, name recognition and normalization, Internet forum

1. INTRODUCTION

Reading relevant discussions and reviews has become a common practice for many users before they purchase products like mobile phones and digital cameras. Users also seek for recommendations through Internet forums and social networking sites (*e.g.,* Facebook, Google+, and Twitter). Such discussions and review comments are important resources for product providers to better understand consumers' concerns so as to improve their products or marketing strategies. However, users often mention product names with a large number of name variations including informal abbreviations, misspellings and nicknames. Table 1 lists 25 name variations of *"Samsung Galaxy SIII"*, each used by at least 10 users in an Internet forum used in our study. Many short forms are not covered in knowledge bases like Wikipedia.

In this research, we focus on mobile phones and aim to *recognize and normalize phone name mentions in Internet forums*. Table 2 lists 4 example sentences, taken from a forum. Our task is to recognize the phone name mentions (highlighted in boldface) and link the recognized names to their corresponding formal names (shown in brackets). In our problem setting, we assume two sets of inputs: (i) a collection of formal/official mobile phone names, and (ii) a collection of discussion threads from an Internet forum that are relevant to mobile phones. Our proposed solution, named GREN, consists of the following three main components.

Table 1: Name variations of *"Samsung Galaxy SIII"*

Name variation	#users	Name variation	#users
1. galaxy s3	553	14. lte s3	46
2. s3 lte	343	15. galaxy s3 lte	45
3. samsung galaxy s3	284	16. s3 non lte	32
4. s iii	242	17. samsung galaxy siii	32
5. galaxy s iii	225	18. sgs 3	27
6. samsung s3	219	19. samsung galaxy s3 lte	22
7. sgs3	187	20. sg3	21
8. siii	149	21. gsiii	16
9. samsung galaxy s iii	145	22. samsung galaxy s3 i9300	15
10. i9300	120	23. samsung i9300 galaxy s iii	13
11. gs3	82	24. s3 4g	11
12. galaxy siii	61	25. 3g s3	11
13. i9305	52	–	

Candidate Name Generator. Candidate names are generated in two phases. First is to find name variations of phone *brands* including their common misspells and abbreviations (*e.g.,* bberry for BlackBerry). Next is to generate phone name variations by assuming that (i) a phone name is a noun phrase, and (ii) a phone name shall either contain a brand variation or appear after a brand variation at least once. As the result, we obtain a relatively large collection of candidate names. However, not all candidate names obtained are truly mobile phone names. For example, the word "battery" appears more than once after a brand (*e.g.,* Sony battery); then "battery" will be considered as a candidate name.

CRF-based Name Recognizer. The name recognizer is a classifier based on linear chain CRF. Given a sentence which contains at least one candidate name mention, the classifier predicts whether the mention indeed refers to a mobile phone. To learn the CRF classifier, we use three types of features including lexical features, grammatical features, and the features derived from candidate names and candidate name mentions. We highlight that a large number of training instances are semi-automatically labeled in our implementation with minimal human annotation effort.

Rule-based Name Normalizer. The last component of GREN maps a recognized name variation to its formal name (*e.g.,* "ssg3" is mapped to "Samsung Galaxy SIII"). The normalization is mainly based on lexical rules. The confidence of a mapping is measured by the co-occurrence of the candidate name and the formal name in selected forum threads.

SIGIR Symposium on IR in Practice (SIRIP'14), Gold Coast, Australia
ACM .

Table 2: Mobile phone name mentions (highlighted in boldface), and their formal names [in brackets] in 4 example sentences

1. True, **Desire** [HTC Desire] might be better if compared to **X10** [Sony Ericsson Xperia X10] but since I am using **HD2** [HTC HD2], it will be a little boring to use back HTC ...
2. I just wanna know what problems do users face on the **OneX** [HTC One X]... of course I know that knowing the problems on **one x** [HTC One X] doesn't mean knowing the problems on **s3** [Samsung Galaxy SIII]
3. Still prefer **ip 5** [Apple iPhone 5] then **note 2** [Samsung Galaxy Note II]...
4. oh, the mono rich recording at **920** [Nokia Lumia 920] no better than stereo rich recording at **808** [Nokia 808 PureView].

2. RELATED WORK

The task of Named Entity Recognition (NER) is to extract and classify information units like person, organization, location and other entity types. Relatively good recognition accuracy has been achieved for NER on formal text (*e.g.,* news article). NER on user-generated short text like tweets remains challenging [3]. While the language usage in Internet forum is similar to that of tweets, the approaches proposed in most work are for named entities of all types (*e.g.,* person, location, organization). Here, we aim to detect mobile phone names which follow certain naming conversions.

There is a large number of studies on Named Entity Normalization (NEN) in formal text. However, the performance of NEN is beset by the characteristics of user-generated content like tweets [4]. Wu *et al.* [7] achieve the best accuracy in the contest organized by ICDM2012.[1] The task is to detect name mentions from web pages or forum messages and link the name mentions to product catalog. Different from [7] and other submissions to the contest, instead of recognizing product names directly from forum messages, we generate candidate phone names and then use CRF-based classifier to predict whether a candidate name mention is a mobile phone name. For name normalization, we use lexical rules to directly link a candidate name to its formal name. Also focusing on a relatively narrow domain, a two-phase (candidate name identification and candidate name validation) method is proposed in [6] for extracting dish names from blog reviews in Chinese. Both phases are significantly different from ours because of the lexical rules and the semi-supervised learning method utilized in our approach.

3. OVERVIEW OF GREN

3.1 Candidate Name Generation

A mobile phone formal name usually contains two components: *brand* and *model name*. Some phones (about 22% in our dataset) have *model numbers*. That is, a phone can be identified either by its *brand and model name* or by its *brand and model number*. Figure 1 gives an overview of the candidate name generation process. A major challenge dealing with Internet forum data is the informal abbreviations and misspellings. To partially address this challenge, we adopt Brown Clustering, a hierarchical clustering algorithm which groups the words with similar meaning and syntactical function together [1]. Using the clustering results, we obtain brand variations and then generate candidate phone names, detailed next.

Compared to the number of different phone models, number of brands is much smaller. Because of their clear context in word usage, almost all variations of the same brand are grouped into the same word cluster through Brown clustering. An example word cluster containing brand "Samsung" is shown in Figure 1. The cluster has 29 words and many of them are spelling variations of "Samsung" and its nicknames (*e.g., sam* and *sammy*). If a word cluster contains a mobile phone brand, we call it a *brand word cluster*.

Because of the relatively good quality of word grouping for mobile phone brands, we apply lexical rules[2] as *brand filter* to identify brand variations from a brand word cluster. Recall that a mobile phone name contains brand and model name. A word cluster containing a model word may have higher chance containing model variations. Based on the assumption that a mobile phone name is a noun phrase, we filter noun phrases to obtain candidate phone names using lexical rules. Shown in Figure 1, we capture most phone name variations such as "sumsang galaxy s3". However, due to the noisy word usage contexts of model names like "one" and "active", the collection of candidate names contains a lot of noise. For example "service center", "samsung updates" are listed as candidate names because they satisfy both conditions.

Table 3: Original and rewritten sentence with labels for "ip_5"

Original sentence:	Still	prefer	ip 5	then	note 2
Rewritten sentence:	Still	prefer	ip_5	then	note_2
"ip_5" (w_i):	w_{i-2}	w_{i-1}	w_i	w_{i+1}	w_{i+2}

3.2 CRF-Based Name Recognition

We now have a set of candidate names that might be used to refer to mobile phones in forum. Given a sentence mentioning at least one candidate name, our task is to predict whether this mention indeed refers to a specific phone model, *i.e.,* a binary classification task. We adopt linear-chain CRF model [2].

Training Examples. Obtaining a reasonable number of high quality training examples is always a major challenge in training a classifier. We propose to generate training examples for our CRF model in a semi-automatic manner with minimal manual labeling effort.

Before we create labeled sentences for CRF training, we create two sets of mobile phone names: a set of positive names ($\mathcal{P}$) and a set of negative names ($\mathcal{N}$). Recall that a set of formal names is given as one input. Each formal name (brand and model name, without model number) is inserted into $\mathcal{P}$. To further enlarge the list of positive names, if a model name has more than one word (*e.g.,* "galaxy note"), then the model name is inserted into $\mathcal{P}$ as a separate entry. Populating the set of negative names $\mathcal{N}$ is through manual annotation. Many entries in the set of candidate names are not truly mobile phone names. A number of entries that can be easily manually identified not referring to phone names are selected to form $\mathcal{N}$. Examples are "debug", "service center", "retail prices", "toolbar", "firmware", "update". We argue that annotating words/phrases that are not phone names is much easier and less time-consuming compared to annotating sentences.

With sets $\mathcal{P}$ and $\mathcal{N}$, we select sentences satisfying two conditions to be training examples: (i) The sentence contains at least one entity in either set $\mathcal{P}$ or set $\mathcal{N}$; and (ii) The sentence does not contain any entry appears in $\mathcal{C} \setminus (\mathcal{P} \cup \mathcal{N})$. The second condition is set because the entries in $\mathcal{C} \setminus (\mathcal{P} \cup \mathcal{N})$ are not manually annotated.

Features. Because our task is to predict whether a candidate name mention is truly a mobile phone name, we consider each candidate name "*a single token*" in the sentence. More specifically, we rewrite the sentence by adding underscores to the contained words in a candidate name, as shown in Table 3. The rewriting enables us to take surrounding words of a candidate name to be its context in a more natural manner.

[1] http://icdm2012.ua.ac.be/content/contest

[2] The lexical rules are not detailed here because of page limit.

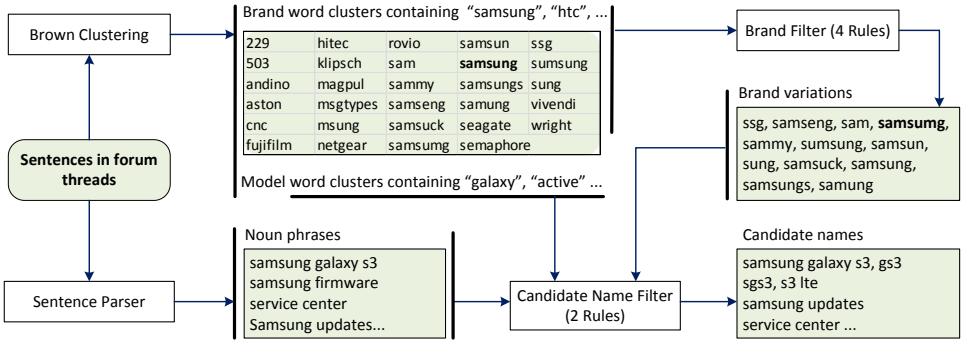

Figure 1: Overview of candidate name generation using brand "Samsung" as an example

Table 4: Feature description for lexical features (L_1, L_2), grammatical features (G_1, G_2), and name features (N_1, N_2)

L_1	The current word and its surrounding two words $w_{i-2}w_{i-1}w_iw_{i+1}w_{i+2}$, and their lower-cased forms.
L_2	Word surface feature of the current word: Initial capitalization, all capitalization, containing capitalized letters, all digits, containing digits and letters.
G_1	POS tagger of the current word and its surrounding two words.
G_2	Path prefixes of length 4, 6, 10, 20 (*i.e.,* maximum length) of the current word by Brown clustering.
N_1	Flags to indicate whether the current word and its surrounding two words are candidate phone names
N_2	The brand entropy of the current word and its surrounding two words.

We propose to use 6 features in our CRF model (listed in Table 4), mostly considering the current word w_i, and its surrounding two words on both sides: $w_{i-2}w_{i-1}$ to its left-hand side and $w_{i+1}w_{i+2}$ to its right-hand side. We use "YNO" (Yes-No-Out) scheme. Given a rewritten sentence as a training example, if a word is rewritten from a positive name in $\mathcal{P}$, then the word is labeled "Y"; if a word is rewritten from a negative name in $\mathcal{N}$, the word is labeled "N". The remaining words in the sentence are labeled "O".

3.3 Rule-based Name Normalization

If a candidate name is predicted to be a phone name variation, our next task is to map this name to its formal name. For example, all the 25 variations listed in Table 1 shall be mapped to "Samsung Galaxy SIII". We adopt a rule-based approach to creating the mapping. Because the way we train our CRF classifier, most phone name variations detected are originated from the candidate name set C. That means we can pre-normalize candidate names in C to their corresponding formal names with some degree of noise because not all candidate names are truly mobile phone names. For each formal name f, we build a list of its possible variations from C with confidence scores, denoted by $\mathcal{L}^f$. A candidate name in list $\mathcal{L}^f$, if detected to be true by the CRF classifier, is normalized to formal name f.

Given a formal name f containing brand f_b, model name f_m, and model number f_r, we select its name variations from C in two steps. First, given a candidate name c, if all its characters are contained in the brand and model name of the formal name and are arranged in the same sequence, then we consider c is a name variation of f. For example, all characters in candidate name "SGS III" are contained in "Samsung Galaxy SIII" and are in the same sequence. The match is case-insensitive and Roman numbers match Arabic numbers. As the result, this rule returns true for "s3", "galaxy s3", "sgs3", and "siii" etc. We now get the initial set of name variations in $\mathcal{L}^f$. Next, we try to learn from this set of name variations on how users name this phone. For this purpose, we tokenize all names currently in $\mathcal{L}^f$ to get all words people use to refer to this phone. The brand variations and single-character words are ignored. Example words obtained include "s3", "galaxy", and "sgs". Then, if any word in a candidate name that is currently not in $\mathcal{L}^f$ matches one of these words, then the candidate name is added to $\mathcal{L}^f$. With this step, we get candidate names like "s3 lte", "s3 phone", "s3 4g" and "s3 pebble blue". Through the above two steps, each formal name f is now associated with a list of candidate names stored in $\mathcal{L}^f$. Then a confidence score is computed based on the co-occurrence of a candidate name and its formal name in the forum posts. If a candidate name appears in multiple lists, only the instance with the highest confidence score is kept in the corresponding list.

4. EXPERIMENT

Dataset. We collected forum data from the discussion board titled "Mobile Communication Technology" in HardwareZone forum, probably the most popular forum in Singapore.[3] Most discussions in this board are about mobile phones and a few user interest groups are formed for a few brands: BlackBerry, Nokia, Samsung, Motorola, Sony, LG, HTC, and Apple's iPhone. In total, 25,251 discussion threads were collected, containing 1,026,190 post messages. The time duration of the discussion is from March 15, 2002 to May 2, 2013 in our data collection.

The formal names of mobile phones were crawled from GSMArena.com.[4] We only consider the mobile phone models from the 8 brands where user interest groups are found in HarewareZone: BlackBerry, Nokia, Samsung, Motorola, Sony (Sony Ericsson), LG, HTC, and Apple's iPhone. For these 8 brands, 2,623 formal mobile phone names were obtained from GSMArena. Only a subset of these 2,623 phone models were released in Singapore and the old models released before March 2002 are not covered by our dataset.

To evaluate the accuracy of name recognition and normalization, we select 20 mobile phones from these 8 brands which are relatively popular in the forum. For each phone, we randomly selected a thread with about 100 post messages, then manually labeled the mobile phone name mentions in these posts and mapped to their formal names. In total, we labeled 4,121 sentences within which there were 946 phone name mentions. The 946 name mentions include name variations for the 20 selected mobile phones as well as name variations of other phone models, because users often compare phones with many different models in their discussion.

Methods. We compare the proposed GREN method with other 3 baselines: GREN-NC, StanfordNER, and LexicalLookup. In GREN, every candidate name mention is identified and rewritten to a single

[3] http://forums.hardwarezone.com.sg/
[4] http://www.gsmarena.com/

Table 5: Precision, recall, and F_1 of the 4 methods with best results in boldface

Method	Name recognition			Name normalization		
	Pr	Re	F_1	Pr	Re	F_1
LexicalLookup	**0.983**	0.297	0.456	**0.983**	0.297	0.456
StanfordNER	0.904	0.373	0.528	0.943	0.362	0.523
GREN-NC	0.931	0.524	0.671	0.950	0.497	0.652
GREN	0.827	**0.875**	**0.850**	0.920	**0.841**	**0.879**

word. In GREN-NC method, no candidate name mention is identified and the sentences are kept in their original form. StanfordNER is of one of the most widely used package for named entity recognition. We adopted the default features provided by the package[5] and trained the CRF classifier using the same set of labeled sentences as in GREN and GREN-NC. We stored the 412 formal names that were used as positive names for sentence labeling in a dictionary. Phone name mentions were recognized by lexical lookup in this dictionary.

Experimental Results. The *Precision* (Pr), *Recall* (Re) and F_1 of the four methods for name recognition and name normalization are reported in Table 5. All the 4 methods achieve very high precision in name recognition. LexicalLookup gets nearly perfect precision as expected as all matched name mentions are formal names. Although the proposed GREN method has the lowest precision value of 0.827 among all methods, GREN significantly outperforms the other methods in recall. Because of the much higher recall, GREN achieves F_1 of 0.850, followed by GREN-NC of 0.671 and StanfordNER of 0.528. From this set of results, we argue that both candidate name generation and sentence rewriting are main factors contributing to the significant improvement of F_1 for GREN. Recall that the positive names used to generate training sentences are either formal phone names or model names (without brand) that contain at least two words. As the result, the name mentions seen by StanfordNER and GREN-NC in the training data do not contain any informal abbreviations or misspells. It becomes reasonable that StanfordNER and GREN-NC achieves better precision than GREN but much poorer recall. The better recall by GREN-NC over StanfordNER attributes to the Brown clustering feature (G_2 in Table 4) which has been reported effective in NER from user-generated content like tweets [5].

We now discuss the results on name normalization. There is no change in results of LexicalLookup because the names recognized are formal names by the method definition. For the rest three methods (*i.e.,* StanfordNER, GREN-NC, and GREN), improvement in precision and degradation in recall are observed compared to their corresponding results on name recognition. In our evaluation for name normalization, a positive instance refers to a name mention that is correctly recognized and correctly normalized to its formal name. Many of the mentions that are wrongly recognized as phone names by the name recognizer cannot be normalized to any formal names using the normalization rules are now considered negative. On the one hand, the normalization process removes errors from the results of name recognition, leading to increase in precision. On the other hand, due to the incompleteness of the rules, some of the correctly recognized names cannot be normalized to their corresponding formal names, *e.g.,* "Nozomi" is not normalized to "Sony Xperia S", leading to degradation in recall. Considering both precision and recall, GREN increases its F_1 from 0.850 to 0.879 after name normalization. Slightly worse F_1's are observed for both GREN-NC and StanfordNER. Nevertheless, in real applications, more rules may be added to address the special cases in name normalization.

Table 6 reports the precision, recall and F_1 of name normalization using all features in GREN and after removing one or two features (*e.g.,*-L_1, -N_1, N_2). Surprisingly, we observe that much better F_1 is achieved by removing the word surface feature L_2 from GREN (*i.e.,* GREN-L_2). Removing L_2 feature leads to improvement of F_1 from 0.879 to 0.896, about 2%. For comparison, we include results of GREN-NC and GREN-NC-L_2 in the last two rows in the table. Observe that removing L_2 from GREN-NC adversely affects F_1 with a big drop in recall. While word surface features are effective in most NER tasks, our results suggest that such features derived from the artificially created words (*i.e.,* the single words rewritten from candidate names in GREN) confuse the CRF model. Among the other features, features from Brown clustering are the most effective features. The two features derived from the rewritten candidate names (N_1 and N_2) are the second most effective features.

Table 6: Name normalization accuracy after removing one or two features. Best results are highlighted in boldface

Method/Feature	Pr	Re	F_1
GREN	0.920	0.841	0.879
GREN-L_1	0.916	0.836	0.874
GREN-L_2	0.918	**0.875**	**0.896**
GREN-G_1	0.920	0.839	0.877
GREN-G_2	0.899	0.728	0.804
GREN-N_1, N_2	**0.961**	0.791	0.868
GREN-NC	0.950	0.497	0.652
GREN-NC-L_2	0.953	0.447	0.609

5. CONCLUSION

In this paper, we propose a novel and practical method for mobile phone name recognition and normalization in Internet forums. Different from most NER approaches where named entities are recognized from sentences directly, we generate possible name variations based on word usage patterns and mobile phone naming conventions. The pre-generation of candidate names also enables us to obtain a large number of training examples at very low cost for manual annotation. Through extensive experiments, we show that our proposed GREN method significantly outperforms baseline methods, particularly in recall measure for name recognition and normalization. The GREN method is flexible in the sense that both the candidate names and the name normalization rules can be easily modified to accommodate special cases in practical applications. We believe that the overall concepts of candidate name generation and candidate name mention prediction can be adopted in product name detection and normalization from user-generated content.

6. REFERENCES

[1] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479, 1992.

[2] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289. Morgan Kaufmann Publishers, 2001.

[3] C. Li, A. Sun, J. Weng, and Q. He. Exploiting hybrid contexts for tweet segmentation. In *SIGIR*, pages 523–532. ACM, 2013.

[4] X. Liu, Y. Li, H. Wu, M. Zhou, F. Wei, and Y. Lu. Entity linking for tweets. In *ACL*, pages 1304–1311. ACL, 2013.

[5] A. Ritter, S. Clark, Mausam, and O. Etzioni. Named entity recognition in tweets: An experimental study. In *EMNLP*, pages 1524–1534. ACL, 2011.

[6] R. T.-H. Tsai and C.-H. Chou. Extracting dish names from chinese blog reviews using suffix arrays and a multi-modal crf model. In *The 1st Int'l Workshop on Entity-Oriented Search (EOS)*, 2011.

[7] S. Wu, Z. Fang, and J. Tang. Accurate product name recognition from user generated content. In *IEEE ICDM Workshops*, pages 874–877, 2012.

[5] `http://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/ie/crf/CRFClassifier.html`