Two Database Resources for Processing Social Media English Text

Eleanor Clark and Kenji Araki
Graduate School of Information Science and Technology, Hokkaido University
Kita 14, Nishi 8, Kita-ku, Sapporo, Hokkaido 060-0814 Japan
E-mail: eleanor@media.eng.hokudai.ac.jp, araki@media.eng.hokudai.ac.jp

Abstract

This research focuses on text processing in the sphere of English-language social media. We introduce two database resources.
The first, CECS (Casual English Conversion System) database, a lexicon-type resource of 1,255 entries, was constructed for use in our experimental system for the automated normalization of casual, irregularly-formed English used in communications such as Twitter. Our rule-based approach primarily aims to avoid problems caused by user creativity and individuality of language when Twitter-style text is used as input in Machine Translation, and to aid comprehension for non-native speakers of English. Although the database is still under development, we have so far carried out two evaluation experiments using our system which have shown positive results.
The second database, CEGS (Casual English Generation System) phoneme database contains sets of alternative spellings for the phonemes in the CMU Pronouncing Dictionary, designed for use in a system for generating phoneme-based casual English text from regular English input; in other words, automatically producing humanlike creative sentences as an AI task. This paper provides an overview of the necessity, method, application and evaluation of both resources.

Keywords: Natural Language Processing, Social Media, Text Normalization

1. Token-to-token Database for Text Normalization of Casual English

1.1 Necessity

Although research aimed at the specific problem of automatically normalizing casual English is relatively rare, there is a clear need to clean noisy data obtained from social media data for use in multiple NLP tasks, including machine translation, information retrieval, ontology creation, and others (Wong et al., 2007; Henriquez & Hernandez, 2009; Ritter et al., 2010). The rapid expansion of Internet use, electronic communication and user-oriented media such as social networking sites, blogs and microblogging services has led to an equally rapid increase in the need for non in-group human users – for example, non-native readers of English and older Internet users - to understand casual written English, which often does not conform to rules of spelling, grammar and punctuation. With automated normalization of noisy forms, these excluded users could enjoy more active participation in Web 2.0 communications such as chat applications, Twitter, internet comment boards and others.

1.2 Defining Casual English

Our database is organized on the premise that errors and irregular language used in casual English found in social media can be grouped into several distinct categories. We thus define "casual English" as tokens which fall into the eight categories used in CECS' database, which are as follows.

1. **Abbreviation (shortform).** Examples: *nite* ("*night*"), *sayin* ("*saying*"); may include letter/number mixes such as *gr8* ("*great*").
2. **Abbreviation (acronym).** Examples: *lol* ("laugh out loud"), *iirc* ("if I remember correctly"), etc.
3. **Typing error/ misspelling**. Examples: *wouls* ("*would*"), *rediculous* ("*ridiculous*").
4. **Punctuation omission/error.** Examples: *im* ("*I'm*"), *dont* ("*don't*").
5. **Non-dictionary slang**. This category includes word sense disambiguation (WSD) problems caused by slang uses of standard words, e.g. *that was well mint* ("*that was very good*"). It also includes specific cultural reference or in group-memes.
6. **Wordplay.** Includes phonetic spelling and intentional misspelling for verbal effect, e.g. *that was soooooo great* ("*that was so great*").
7. **Censor avoidance**. Using numbers or punctuation to disguise vulgarities, e.g. *sh1t*, *f****, etc.
8. **Emoticons.** While often recognized by a human reader, emoticons are not usually understood in NLP tasks such as Machine Translation and Information Retrieval. Examples: :) (smiling face), <3 (heart)

1.3 Approach

In our normalization system, CECS, tokenized input is passed through a database to find a match, using a trie-type data structure. The database is recursively loaded into a trie to allow easy item lookup, tokenized by the same tokenizer used for input. Database entries which are a front-anchored substring are allowed, but full matches are not. Using this data structure, multi-word phrase matching is enabled. The flow of CECS is shown schematically in Figure 1.

When a match is found, the normalized English equivalent is displayed in the user interface in the "Output" pane, and the replaced item's category and notes, where present, are displayed in the "Notes" pane. Tokens not found in the database are passed through unchanged.

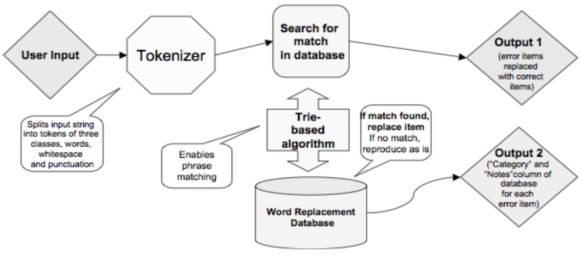
User Input
Tokenizer
Search for match in database
Output 1 (error items replaced with correct items)
Splits input string into tokens of three classes, words, whitespace and punctuation
Enables phrase matching
Trie-based algorithm
If match found, replace item If no match, reproduce as is
Output 2 ("Category" and "Notes"column of database for each error item)
Word Replacement Database

Figure 1: System flow of CECS

An example sentence successfully normalized by CECS is as follows.

> **Raw input:**
> B4 u run, u need 2 walk, b4 walking u need 2 crawl
> **System output:**
> before you run, you need to walk, before walking you need to crawl

1.4 Database Construction and Rules

CECS uses a manually compiled and verified database, currently of a total of 1,255 entries. These entries are either single words or phrases; the trie-type data structure theoretically allows for phrases of unlimited word length, but at present the majority of phrase entries are sets of two or three words. Phrase matching in CECS is an important feature. Firstly, slang phrases constituting more than one word can be matched in the database; secondly, problems regarding word sense disambiguation (WSD) problems can be tackled to some extent.

Each database entry has been taken from training data which is rich in casual English occurrence, including Twitter[1] entries and YouTube[2] comment boards, and meanings have been verified through collaborative user-compiled, user-evaluated resources such as Wiktionary[3] and Urban Dictionary[4]. Database entries comprise of four classes: "error word" (the casual English item), "regular word" (the corresponding dictionary English item), "category" (the item's category as defined in Section 1.2) and "notes" (cultural or linguistic information about the item's origin, intended for CECS' human users). Database construction is an ongoing project, and we intend to improve its coverage and quality further.

A number of the phrase entries attempt to handle word sense disambiguation (WSD) problems within casual English using context-based token sets. For example, the token *ur* can either mean the possessive "your" or the contraction "you're". The database contains various token sets including *ur* which can be normalized with high probability of accuracy e.g. *ur so* (you're) *ur wrong* (you're), *ur friend* (your), *what's ur* (your).

1.5 Evaluation

Two evaluation experiments were conducted in order to assess CECS' effectiveness as a preprocessing system for Machine Translation (MT) input, and also as a reading comprehension aid for non-native readers of English.

In testing CECS as a preprocessor for MT input, 100 sentences from Twitter, taken from Choudry et al.'s Twitter corpus (2010), were run through two MT applications, Google Translate[5] and Systran[6]. The same sentences were then pre-processed with CECS and run through Google Translate and Systran a second time. The quality of the resulting translations was compared by measuring error incidence. The working language pair used was English to Japanese. MT errors were counted manually in two separate categories, "non-translated word" ("NTW") and "wrongly translated word" ("WTW"). An NTW was defined as cases when the MT application simply outputs a token in the original English, thus not converting to Japanese at all. A WTW was defined as a Japanese word that is completely semantic different from the English meaning. In the Twitter data, with an average sentence length of 15.35 words, there was a decrease in NTW occurrence from 3.34 to 0.86 words per sentence (average of both MT applications, see Table 1). This significant drop showed that CECS' database coverage already gives a reasonable performance.

In evaluating CECS for human users, ten non-native learners of English between the ages of 23 and 64 completed two questionnaires, in which they were asked to assess their understanding of 20 sentences, also taken from Twitter. The first questionnaire used raw input for the sentences, and the second questionnaire used the same sentences after processing by CECS. No participants were allowed to see the corrected sentences until they had submitted the first questionnaire. Rankings were made on a five-point semantic differential scale, as follows:

Question: *How much of the sentence can you understand?*
1. *None at all* ***2.*** *A little* ***3.*** *Some* ***4.*** *Most* ***5.*** *All*

Overall, average understanding of the 20 sentences increased by exactly one semantic differential point: evaluator comprehension of the sentences averaged at 2.89 for raw input, on the low side of "Some" on the semantic scale, and 3.89 for system output, or slightly lower than "Most" on the semantic scale (Table 2). The evaluators were asked to self-assess their English ability prior to completing the questionnaire; when grouped by English level, the largest improvement in comprehension was seen in the lowest level participants.

	Raw Input		CECS Ouput	
	NTWs*	WTWs	NTWs	WTWs
Google MT	2.78	1.55	0.83	0.86
Systran MT	3.83	0.84	0.77	0.56
Average of both MT systems	**3.31**	**1.2**	**0.8**	**0.71**

*All NTW (non-translated word) and WTW (wrongly-translated word) counts are given as an average per sentence.

Table 1: Error counts in all sentences

[1] http://twitter.com
[2] www.youtube.com
[3] www.wiktionary.org
[4] www.urbandictionary.com
[5] http://translate.google.com
[6] http://www.systranet.com

	Level 2 English (Basic)	Level 3 English (Fair)	Level 4 English (Good)
Reader understanding: **Raw input**	1.53*	3.26	3.88
Reader understanding: **System output**	2.93	4.21	4.53

*Reader understanding is given as an average of answers made on a semantic differential scale of 1-5, where 5 is full comprehension.

Table 2: Reader understanding of 20 Twitter sentences before and after using CECS

2. Phoneme-to-Phoneme Database for Automated Creation of Casual English

2.1 Necessity

During this research, we became interested in the creation of a reverse version of CECS, in other words a casual English generation system, primarily as an AI task. Automatic generation of slang-type English from regular input text would be useful in areas such as social media marketing, targeting teenage consumers, or making chatbots seem more humanlike.

2.2 Approach

Although the CECS database could simply be used in reverse, a system which could turn any input word into slang, not only commonly-used existing slang forms such as those collected in CECS, would be more interesting. We instead chose a phoneme-by-phoneme approach, which attempts to mimic SMS (short message service) or Twitter-type phonetic spellings by selecting replacement candidates at the phonemic level. Selected tokens are split into phonemes using the CMU Pronouncing Dictionary, and these phonemes are then converted into the multiple alternative phonemes in our database. As this method can produce highly creative phonetic slang, it is necessary to strike a balance between "interesting" and "difficult to understand".

One important point in casual English sentence design is that, usually, not all tokens (words) in a given sentence are irregular. Even if only a small proportion of tokens per sentence consists of casual English items, this is often enough to render the sentence incomprehensible to a non-native speaker or to a machine translation application, as shown in the experiments in Section 1.5. Thus, frequency of casual English tokens per sentence was selected based on prior linguistic analysis of 320 tweets, in which casual English items and their POS were manually tagged, in order for the method to reflect the human creation of casual English sentences in a more natural way. Our analysis found an average of 21.67% occurrence of casual English tokens per sentence. In the code of CEGS ("Casual English Generation System"), this is rounded up to 22% selection of input tokens to be processed after the initial filtering stages. Although the secondary goal of the analysis experiment was to determine distribution of casual English tokens across POS categories, we found that POS categories were not in fact shown to be a significant factor in the placing of casual English tokens overall. However, we found that certain words, particularly pronouns and some contractions, were very often written in the same way, e.g. "*u*" for "you", "*im*", for "I'm"; so these tokens are incorporated into CEGS using a filter consisting of a small section of the CECS database (with input and output reversed). An overview of the system is shown schematically in Figure 2.

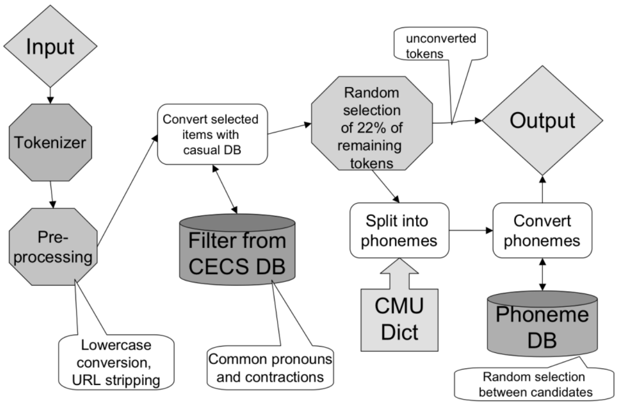
Input
Tokenizer
Pre-processing
Lowercase conversion, URL stripping
Convert selected items with casual DB
Filter from CECS DB
Common pronouns and contractions
Random selection of 22% of remaining tokens
unconverted tokens
Output
Split into phonemes
CMU Dict
Convert phonemes
Phoneme DB
Random selection between candidates

Figure 2: System flow of CEGS.

The process of CEGS is as follows. First, input (which is assumed to be regular English with no misspellings) is tokenized using a simple whitespace delimiter and removal of punctuation. Next, a single character string array of same length as the tokens in the input is created, in order to assign Boolean-type values of true ("process") or false ("do not process") to each token; this is because CEGS requires that only a minority of tokens are processed, as explained above. We then conduct some minor preprocessing on the input such as assigning "do not process" to certain tokens including URL or email indicators ("www", "http", "@", etc.) A second layer of preprocessing converts a fixed set of common standard tokens using the CECS database.

After this stage, 22% of the remaining tokens are selected randomly using Python's random module, which employs the Mersenne Twister as its generator[7]. These tokens are assigned "process", while the rest are assigned otherwise. The processable tokens are split into their constituent phonemes using the CMU Pronouncing Dictionary, through the interface built into the Natural Language Toolkit (Bird et al., 2009). Numbers signifying lexical stress and multiple outputs from the CMU Dict. are removed, and the resulting phonemes are converted using our original phoneme database. Since many of these phonemes have multiple conversion candidates in the CEGS database, random selection between candidates is performed, giving different output each time in many cases. The output then consists of the sentence composed of filtered, processed and unprocessed tokens.

2.3 Database Construction and Rules

The alternative phoneme representation is constructed based on analysis of the large volume of casual English examples collected during this research.

[7] http://docs.python.org/library/random.html

The database consists of the 39 phonemes of the CMU Pronouncing Dictionary, with our alternative original phonemes as replacements. These phonemes were selected based on their occurrence in casual English words in the Twitter corpus used in our research. Most of the phonemes, although not all, have multiple replacement candidates, which our method selects between randomly. For example, the word "everything" is split into the phonemes EH V R IY TH IH NG by CMU Dict. In our database, the phoneme IY has multiple candidates of *ee, y,* and *i,* TH has multiple candidates of *t', th, f,* and *ff,* and NG has the multiple candidates *n, n', nng* and *ngg*. Thus, "everything" could be converted as various combinations, such as *evryffin', evreet'in, evrithingg,* etc.

2.4 Generated Text

Creating convincing colloquial language can be seen as a highly difficult task, as it can be considered to fall into the sphere of the Turing test. Future work on CEGS includes human evaluation on the casual English output. We intend to determine the human-likeness of the output by asking evaluators to identify generated Tweets among human-authored ones. An example of the kind of output currently generated by the CEGS database is shown below: the opening paragraph from a Wikipedia[8] article on backgammon, which is part of the dataset for the LREC Language Library. Note that the distribution and positions of converted tokens and their phoneme combinations will be different each time due to the random selection of candidates.

Original text:

Backgammon is one of the oldest board games for two players. The playing pieces are moved according to the roll of dice, and players win by removing all of their pieces from the board. There are many variants of backgammon, most of which share common traits. Backgammon is a member of the tables family, one of the oldest classes of board games in the world.

CEGS Output:

backgammon is wunn of da oldest board geymz ffawr two ppleyurz . da playing pee$uz r moved according 2 ddo roall uv dday$, and players win by removing all of their pieces from da board . der r many variants uv backgammon , mow$t of which share common traits . backgammon is a member of da tables family , wunn av da oldust classes of board games in da world .

It can be observed that in this instance CEGS output includes broadly legible forms, e.g. "*wunn*" for "one", "*mow$t*" for "most", and more difficult coinages such as "*ffawr*" for "for", "*dday$"* for "dice". We are interested in determining to what extent reader cognition is tolerant of diverse forms, and how strict limits should be set on "creativity" from such a system.

3. Conclusions

We have presented the CECS database, used in a rule-based text normalization system for casual English, and the results of two evaluation experiments. Both the Machine Translation-based experiment and human evaluation-based experiment showed positive results, with a significant reduction in non-translated words in the former, and a notable improvement in reader comprehension in the latter after pre- processing Twitter sentences with our system. Human evaluation feedback emphasized both the usefulness and need for this system, and gave us ideas for future improvements. We consider that the main tasks hereafter will be the ongoing expansion of the database, and developing the system with additional techniques such as the integration of an open-source spellchecking tool for dealing with a wider range of spelling errors, and the implementation of a Web mining algorithm for access to a wider knowledge base.

In addition to this, we have described CEGS, which employs a phoneme-based database for automated generation of casual, irregularly-formed English used in communications such as Twitter. Based on investigation of the optimum distribution per sentence of casual English vocabulary for automatically producing humanlike creative sentences, we developed a system for converting regular English into casual English as an AI task. Future work on this system includes human user-based evaluation, as a variant of the Turing test.

4. Acknowledgements

The authors gratefully acknowledge financial support for this research provided by the Hokkaido University Global Center of Excellence Program, and the KDDI Foundation.

5. References

Bird, S; Klein, E. and Loper, E. (2009), *Natural Language Processing with Python*. O'Reilly Media.

Choudhury, M. D.; Lin, Y.R.; Sundaram, H.; Candan, K, S.; Xie, L. and A. Kelliher (2010). How does the sampling strategy impact the discovery of information diffusion in social media? In *Proceedings of the 4th International Conference on Weblogs and Social Media*, Washington DC, USA, May 2010.

Clark, E and Araki, K. (2011) Text Normalization in Social Media: Progress, Problems and Applications for a Pre-Processing System of Casual English. In *Proceedings of the Twelfth Conference of the Pacific Association of Computational Linguistics*. Kuala Lumpur, Malaysia, July 2011.

Henriquez, C. A. and Hernandez, A. (2009) A ngram-based statistical machine translation approach for text normalization on chat-speak style communications. In *Proceedings of CAW2.0*, Madrid, Spain, August 2009, pp.1–5.

Ritter, A; Cherry, C. and Dolan, B. Unsupervised modeling of Twitter Conversations. In *Proceedings of HLT-NAACL 2010*, Los Angeles, California, June 2010, pp. 172–180.

Wong, W; Liu, W; and Bennamoun, M. (2007) Enhanced integrated scoring for cleaning dirty texts. In *Proceedings of IJCAI 2007 Workshop on Analytics for Noisy Unstructured Text Data*, Hyderabad, India, January 2007, pp. 55–6.

[8] www.wikipedia.org