社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report

A Rule-based Method of Text Normalization for Casual English

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, araki}@media.eng.hokudai.ac.jp

英語におけるスラング表現のテキスト標準化のためのルールに基づく手法

エレナ・クラーク† 荒木 健治†

†北海道大学大学院情報科学研究科 〒060-0814 北海道札幌市北区北14条西9丁目

E-mail: †{eleanor, araki}@media.eng.hokudai.ac.jp

Abstract This research introduces an experimental system for the automated normalization of casual, irregularly-formed English used in communications such as Twitter. Our rule-based approach 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. We describe the results of two evaluation experiments using our system. Finally, we explore how to effectively utilize the same rule-based approach to generate casual English; in other words, automatically producing humanlike creative sentences as an AI task.

Keywords Natural Language Processing, Text Normalization, Noisy Text, Twitter, Machine Translation

1. Introduction

The rapid expansion of Internet use, electronic communication and user-oriented media such as social networking sites, blogs and microblogging services has led to a rapid increase in the need to understand casual written English, which often does not conform to rules of spelling, grammar and punctuation. Despite this, text normalization is commonly seen as cumbersome [1], and remains a somewhat niche topic of research. Studies which attempt to tackle this problem generally use a fully automated, statistical approach [2,3]; however, we propose that a combination of automated and manual techniques is a potentially more useful approach to this problem. Accordingly, our aim is to develop a method which uses automated tokenization, word matching and replacement techniques in combination with a high-quality, large scale, manually compiled database. We present recent progress on this system, CECS (Casual English Conversion System).

CECS has two applications: as pre-processing on noisy input for automated Natural Language Processing tasks such as Machine Translation or Information Retrieval; and as a standalone system for human users, to aid non-native speakers' reading comprehension of informal written English, the irregularity of which may pose a barrier to their positive participation in 21st Century international communications.

This user-oriented educational aspect of CECS is complemented by the inclusion of annotation on linguistic and/or cultural aspects of each word or phrase converted by the system. At present, the system's knowledge base for text replacement is a manually compiled database of 1,043 items, although expansion of the database is constant and regular.

2. Related Work

Research aimed at the specific problem of automatically normalizing casual English is relatively rare [4]. While spelling error correction is a well-established area, with initial pattern matching and *n*-gram analysis techniques having improved over the last two decades [5], the range of problems presented by user-generated content in online sources go beyond simple spelling correction; other problems include rapidly changing out-of-dictionary slang, short-forms and acronyms, punctuation errors or omissions, phonetic spelling, misspelling for verbal effect and other intentional misspelling, and recognition of out-of-dictionary named entities [6].

Research on unknown vocabulary items often focuses on the recognition and translation/transliteration of proper names; although Sproat *et al.* [1] included some attempts at automatic expansion of acronyms and abbreviations, slang and casual language were not specifically featured. Sproat et al. note that "text normalization is not a problem that has received a great deal of attention, and it (…) seems to be commonly viewed as a messy chore" [1]. Alexander Clark's work on pre-processing a large collection of the Internet discussion system Usenet's posts, through a straightforward machine learning methodology using generative models and a noisy channel method, made some progress towards handling the type of input discussed here, but faced problems with the quality of the corpus and did not reach the evaluation stage [7]. Aw *et al.* [2] have produced a system for normalizing Short Message Service[1] mobile phone texts, which share many of the characteristics of the casual English focused on in this paper, such as non-standard short-forms of words, creative phonetic or stylistic

[1] Short Message Service, or SMS texts are limited to 160 characters in length, which gives necessity for the creative use of new shortened forms of language.

spelling, and punctuation omission, by creating a parallel corpora of 5,000 raw and normalized English SMS messages and applying a phrase-based SMT model, resulting in significantly more accurate translations when the system's output was passed through commercially available MT systems. The use of a phrase-based model rather than a word-based one incorporates logical contextual information to the translation model and thus improves lexical affinity and word alignment. However, their model is essentially a fairly straightforward SMT system, and was limited by the unavailability of parallel corpora suitable for automated constructing of such a system.

Henriquez *et al.* [3], in their work for the CAW 2.0 project introduced an approach using a *n*-gram based SMT system and were able to produce syntactically correct sentences from input with a high frequency of misspelled words and Internet slang, but again found that their system's effectiveness had "a strong dependency on the dictionary quality and size" and that their "small dictionary is not able to handle all possible abbreviations and terms".

With the rapid expansion of new media, the irregularity of language poses a barrier to automated tasks. Ritter *et al.,* in their modeling of Twitter dialogue acts, found that posts were "often highly ungrammatical, and filled with spelling errors", and resorted to selecting clusters of spelling variations manually [8]. The interest in content of this type, both from researchers and corporations, shows a pressing need for effective text normalization of casual English.

3. Casual English Classification System and Database

3.1. Casual English Classification System

Our Casual English Conversion System (CECS), is designed on the basis that errors and irregular language used in casual English found in social media can be grouped into several distinct categories, and accordingly, a multi-faceted approach will be the most effective way to deal with the problem. The categories used in CECS' database 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)

3.2. Database Construction and Rules

CECS uses a manually compiled and verified database, currently of a total of 1,043 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. Each entry has been taken from training data which is rich in casual English, including Twitter[2] entries and YouTube[3] comment boards, and meanings have been verified through collaborative user-compiled, user-evaluated resources such as Wiktionary[4] and Urban Dictionary[5]. Database entries comprise of four columns: "error word" (the casual English item), "regular word" (the corresponding dictionary English item), "category" (the item's category as defined in Section 3.1) 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. Careful manual editing of the database includes checking to avoid rule conflicts, a common problem in rule-based systems.

3.3. Phrase Matching Rules

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. When a word exists as a regular English word but is often used in casual English to mean something else, it is not detected by conventional spellcheckers. As an example, the regular English word "rite" is commonly used as a shortened form of "right". However, it may also be used in its original meaning as "ritual", as the example sentences below show.

> **Regular usage:** Going to high school is tough, but it is a necessary **rite** of passage.
> **Casual usage**: seein that ad makes me wanna listen to dat song **rite** now. (*Seeing that advertisement makes me want to listen to that song right now*)

As well as being confusing to non-native readers, this word causes problems to MT applications, which tend to translate it as "ritual", rendering many casual English sentences difficult to understand after translation. With phrase matching in CECS, common combinations of "rite" which can *only* be used in the sense of "right" can be added into the database. Thus, pre-processing casual English with CECS can improve MT handling of such vocabulary items. Table 1 shows a section of the database entries containing "rite":

Table 1. Section of database entries containing "rite"

Input	Normalization
is rite	is right
it rite	it right
iz rite	is right
so rite	so right
r rite	are right
rite now	right now
rite away	right away

This approach also proves useful for normalizing numbers which have been used as phonetic substitutions, e.g. "4" for "for", "2" for "to" or "too", etc. Whereas it would be obviously inaccurate to automatically convert all instances of the number "4" to "for", with phrase matching it is possible to convert a high number of occurrences correctly using carefully designed combinations. Thus, we can define the rules for the usage of these items manually, and automatically convert appropriately with CECS. So far, the number of necessary phrase matching rules per vocabulary item differs widely.

[2] http://twitter.com
[3] www.youtube.com
[4] www.wiktionary.org
[5] www.urbandictionary.com

While this strategy of addressing WSD cannot yet cover every potential possibility and usage, it is logical that the combinations used are finite and thus can be entered in the database. As more data is collected, analyzed and more examples gathered, the quality and coverage of the database further increases.

4. System Overview

The flow of CECS is shown schematically in Figure 1. CECS is written in the Python programming language. Firstly, user input is tokenized using a strictly regular grammar defined in PyParsing[6], which defines words and punctuation as separate tokens, and allows combinations. "Main characters" are defined as the letters from a-z and A-Z, numbers 0-9 (in case of spellings which incorporate numbers such as "*gr8*" for "*great*"), and selected punctuation marks which may appear mid-word such as apostrophe ("*don't*"), hyphen ("*mid-word*"), and asterisk for censor avoidance spellings ("*s****"), etc.

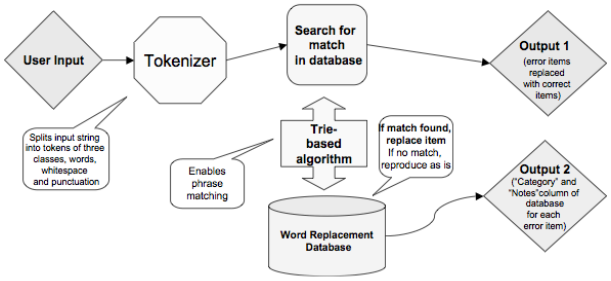

Figure 1. System flow of CECS

"Other characters" are defined as all other ASCII characters, and whitespace and carriage returns are defined separately. A token is thus defined here as either a word composed of main characters ("English word") or composed of other characters ("punctuation token").

Tokenized input is then passed through the 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.

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.

5. Overview of Evaluation Experiments: CECS on MT Input, CECS for Human Evaluators

Evaluation experiments were conduction in order to assess CECS' effectiveness as a preprocessing system for Machine Translation (MT) input, and also as a reading aid for non-native readers of English [6].

5.1 Evaluation Experiment A: CECS Output for MT Use

In testing CECS' as a preprocessor for MT input, 100 sentences from the popular microblogging service Twitter[7] were run through two well-known free MT applications, Google Translate[8] and Systran[9]. The sentences used in Experiment A were gathered from a 10.5 million "tweet" (Twitter posting) Twitter corpus as compiled and publicly released by Choudhury [9]. The corpus contains tweets from 200,000 unique users collected between 2006 and 2009; the 100 sentences used in our experiment were taken from the September 2009 section of the corpus. The user tweets used as data for this experiment were essentially selected at random, but with the following criteria: a) the sentence is written entirely in English b) the sentence contains at least two "errors" or non-dictionary words for CECS to be tested on.

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. Of the 100 Twitter sentences, 20 were "known" sentences, i.e., they had been analyzed for error words and those items were pre-entered into the database. The remaining 80 were "unknown" sentences. MT errors were counted manually in two separate categories, "non-translated word" ("NTW") and "wrongly translated word" ("WTW"). An NTW is defined here as the MT application simply reproducing an item in Roman alphabet letters or numbers, and not converting to Japanese at all. A WTW was defined as a Japanese word that is completely semantic different from the English meaning.

An example of a successfully normalized sentence from this experiment is shown below. NTWs are immediately obvious even to a non-Japanese reader.

Raw input: 4 yr old went 2 her first funeral. Asked me "Grandma when U die, will U invite me 2 UR funeral & can I sit on the front seat?" I said "yes"

Google Translate: 4 yrは、古い2彼女の最初の葬儀 を行った。私質問"おばあちゃんは、Uが死ぬ、Uは2 ウルの葬儀&Iは、前の座席に座ってすることができ ます私を招待のだろうか?"私は言った"はい"
NTWs: 7 (yr, 2, U, U, 2, &, I) WTWs: 1 (ウル from input "UR")

Systran : 4 yr oldは2彼女の最初葬式行きました。 U が死ぬ場合私に「祖母頼まれて、Uは私を2 URの葬 式誘いましたり及び前の座席で置かれることができ ます私か」。を 私は「はい」言いました
NTWs: 7 (yr, old, 2, U, U, 2, UR) WTWs: 0

System output: 4 year old went to her first funeral. Asked me "Grandma when you die, will you invite me to your funeral and can I sit on the front seat?" I said "yes"

Google Translate: 4歳の彼女の最初の葬儀に行きま した。私"おばあちゃんが死んで、あなたの葬式に、 私を招待し、私は前の座席に座ることができるか?" 私は言った"はい"
NTWs: 0 WTWs: 0

Systran: 4歳児は彼女の最初葬式に行きました。 死 ぬ場合私に「祖母頼まれて、あなたの葬式に私を誘 い、前の座席で置かれることができます私か」。を 私は「はい」言いました
NTWs: 0 WTWs: 0

As this sentence used only casual vocabulary items already in the database, NTW occurrence was reduced significantly. WTW incidence was originally low or non-existent. Note that the numbers were normalized correctly by CECS: *4* was unchanged, as it referred to the age of four, but "2" was changed to "to" using the database entry *2 ur* = "to your".

[6] http://pyparsing.wikispaces.com/
[7] www.twitter.com
[8] http://translate.google.com
[9] http://www.systranet.com

The results for Experiment A are shown in Tables 2, 3, and 4. The results for all sentences are summarized in Table 2. Table 3 shows results only for "known" sentences (training data). Table 4 shows results only for "unknown" sentences (test data).

Table 2: Error counts in all sentences (100)

	Raw Input		CECS Output	
	NTWs*	WTWs	NTWs	WTWs
Google MT	2.78	1.55	0.83	0.86
Systran MT	3.83	0.84	0.77	0.56
Avg. 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 3: Error counts in known sentences (20)

	Raw Input		CECS Output	
	NTWs*	WTWs	NTWs	WTWs
Google MT	2.65	1.5	0.5	0.55
Systran MT	3.65	0.8	0.65	0.7
Avg. of both MT systems:	**3.15**	**1.15**	**0.58**	**0.63**

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

Table 4: Error counts in unknown sentences (80)

	Raw Input		CECS Output	
	NTWs*	WTWs	NTWs	WTWs
Google MT	2.81	1.56	0.91	0.93
Systran MT	3.87	0.85	0.8	0.52
Avg. of both MT systems:	**3.34**	**1.21**	**0.86**	**0.76**

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

Comparison between Table 3 and Table 4 reveals that prior entry in the database dramatically increases accuracy, as would be expected. However, the significant decrease in NTWs in the "unknown" data seen in Table 4, from 3.34 to 0.86 words per sentence (average of both MT applications), shows that CECS' current level of database coverage gives reasonable performance. As the database is constantly updated, this is expected to increase.

5.1 Evaluation Experiment B: CECS Output for English Learners

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 the Twitter corpus. 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*

Evaluators were also asked to give a reason for why they could not understand part or all of each sentence. They were given three choices: vocabulary, grammar and context. Attributing more than one reason to failing to understand a sentence was possible. 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. There was no dramatic change in the relative proportion of reasons for non-comprehension (roughly equal before and after using CECS), but "vocabulary" was reduced slightly after pre-processing.

Several sentences were not completely normalized, as the sample came from "unknown" data; many error items were not in CECS database. An example of a partially corrected sentence from Experiment B is as follows:

> **Raw input:** Gr8 ldrs surround themselves w/others who compensate 4 their weeknesses. Who r u surrounded by?
> **System output:** Great ldrs surround themselves with others who compensate 4 their weeknesses. Who are you surrounded by?

Due to the fact that some vocabulary items, particularly *ldrs* (leaders) which is the subject of the first sentence, were not converted, several evaluators assigned a low score to this sentence even after pre-processing with CECS. An example of a more successful conversion 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

This sentence, which received low scores in raw input form – mostly attributed to vocabulary by participants, probably due to the heavy use of numerical substitutions – gained a high proportion of "4" and "5" scores after pre-processing with CECS.

6. Generation of Casual English

6.1 Major Issues for Casual English Generation

Although our main research is a straightforward normalization task for the purposes of "cleaning up" noisy natural language, we are also interested in the creation of a reverse version of CECS, in other words a generation system, as an AI task. We plan to evaluate the finished system by a variant of the Turing test, in which human evaluators are asked whether they think the style of the messages has been created by humans or machine.

In this section, we explore the problems faced in the creation of a system, including the comparative advantages of a word-to-word database and a phoneme-to-phoneme database for converting regular English to casual English, and analyze the optimum proportion and distribution per sentence of casual English vocabulary for automatically producing humanlike creative sentences. Based on this analysis, we will propose the details of our forthcoming method.

6.1.2 Rule-based Approaches

In our casual English normalization research, we use a token-to-token (broadly speaking, word-to-word, although phrase-to-phrase of any number is also possible) database for accuracy. However, it is debatable whether a token-to-token database would be most appropriate for a generation system. The goal here is *creativity* rather than *accuracy*; if all words are converted in the same way in each sentence, the humanlike creativity aspect may be weakened. In humans, five different people may write the same word in five

different ways (e.g. "this" could be written as *dis, diss, diz, thiz, viss*); thus, it may not be interesting in AI terms to use a dictionary-lookup style which states that, for example, "this" must always be converted to *diss*. However, a database which has multiple candidates for common words and a random selection algorithm would negate this problem somewhat.

Another problem with any kind of approach which relies solely on token-to-token database lookup is that out-of-vocabulary (OOV) items could not be converted. A learning method for new words using web-based searches would need to be added in order to keep the database relevant in the face of quickly-evolving language and new slang coinage.

An alternative rule-based approach would be a phoneme-by-phoneme approach, which would mimic SMS (short message service) or Twitter-type phonetic spellings by selecting replacement candidates at the phonemic level. This would be useful for two reasons: by using phoneme-based rules it removes the problem of OOV completely; and as a method which attempts to mimic the process of casual English token creation from scratch rather than use a pre-created database, it may be regarded as a more interesting approach from an AI standpoint. However, it will face similar problems to text-to-speech applications when heteronyms appear: e.g. should the word "read" be converted to "reed" or "redd"? Depending on context, both are possible:

Did you read that book*? > Did u reed dat buk?*
Yes, I read that book*. > Yeah i redd dat buk.*

Another disadvantage of a phoneme-to-phoneme approach is that it will lose the variety of non-phonetic casual English: acronyms, slang etc., as these are not usually based on pronunciation.

6.1.2 Statistical Approaches

Statistical approaches have been utilized in normalizing slang to regular English [2,3], so it seems logical to assume that a statistical method would also be useful in a reverse system of regular English to slang. However, a major disadvantage of this approach is that large-scale parallel corpora of casual English sentences with manually normalized regular English counterparts need to be built as the base for the SMT-like (statistical machine translation) system, which is a non-trivial task. Aw et al. [2] manually normalized a substantial data set of 5,000 raw SMS messages, yet still found that OOV posed a considerable problem when words appeared which did not occur as casual English with their manually normalized equivalents in the parallel corpora. Thus, despite requiring a labor-intensive creation of parallel corpora, such an approach would remain limited in the face of completely new coinages.

6.1.3 Frequency and Distribution of Casual English

One important point in casual English sentence design is that, usually, not all tokens (words) in a given sentence are irregular. As a very broad generalization, it appears that only a small proportion of tokens per sentence tend to be casual English items (this may, however, often be enough to render the sentence incomprehensible to a non-native speaker or to a machine translation application, as shown in the experiments in Section 5).

The ultimate goal for this system is to be a natural recreation of human "slangification" of regular English input sentences. There are several questions that arise regarding this aim. How should we select which words to convert in a sentence? Would an extremely simple rule such as "convert 25% of all tokens at random incidence" or even "convert every fourth token" create an impression of humanlike creativity? Or are there particular parts of speech (POS) which are more likely to be converted, e.g. are nouns more commonly written in slang than verbs? Before making the first steps in designing a method of casual English generation, these rules must be clarified. We propose to devise these rules based on empirical data; as such, we have conducted a preliminary experiment on 320 tweets from Choudhury's Twitter corpus [9].

6.2 Analyzing Casual English

6.2.1 Experiment: Analyzing Tweets

In order to answer some of the questions raised above, we conducted a preliminary experiment on 4,716 words (320 tweets) from Choudhury's Twitter corpus [9]. There were two factors for analysis: first, we attempted to extrapolate the average occurrence per sentence (AOpS) of casual English items. Second, we attempted to establish which, if any, parts of speech (POS) were particularly likely to be written in casual English. The aim of determining these two points was for our proposed system to be capable of mimicking human casual English creation as naturally as possible, by recreating the most commonly seen frequency and distribution trends.

6.2.2 Experiment method

For the purpose of this experiment, casual English items were defined as a) tokens (words) which were flagged by the open source spellchecker Hunspell[10], *and* b) were not named entities or obvious unintentional typing/spelling errors (e.g. a: *Bieber,* b: *appology).* As some clearly intentional spelling errors are typical casual English items used for brevity or style reasons (e.g. *whateva, nuffing),* any spelling error which was deemed to have a likelihood of being intentional was included in the casual English category.

The tweets were taken from the most recent section (Fall 2009) of Choudhury's Twitter corpus. Although selection of tweets for data was again essentially random, tweets written in languages other than English and tweets written in entirely standard English (while mostly English, the corpus also contains tweets in German and Spanish, and others) were excluded from the experiment data. The latter was due to the fact that the aim of this experiment was to analyze the construction of sentences which feature casual English items, not to analyze the incidence of casual English in Twitter sentences as a whole.

POS were recorded by manual annotation, as a conventional POS tagger cannot function effectively on such noisy text. The POS categories were Noun, Verb, Pronoun, Adverb, Preposition, Conjunction, Interjection, and Contraction. Although the first eight represent traditional English POS categories, Contraction was added for the purpose of this experiment due to its frequent occurrence. Contraction refers to contractions of any pair or greater number of tokens which have been written as one token. For example, *imma* (I'm going to). Emoticons, though occurring with relative frequency, were not included as a POS category and thus were not counted as casual English in this experiment. Other tokens manually stripped from the data were URLs and usernames. An example sentence with manual POS tagging is shown below.

Welcome 2 Valencia, Spain! once the weather settles dn, U're gonna luv it hre

The total words in the tweet are 14, with an AoPS of 6, or 43%. These are broken down into: 1 preposition: *2* (to); 2 adverbs: *dn* and *hre* (down); 1 verb: *luv* (love); and 2 contractions: *U're* and *gonna* (you are, going to).

6.2.3 Experiment: Results and Discussion

Table 5 shows the AOpS of casual English in the twitter data. Figure 2 show a breakdown of distribution of casual English in

[10] http://hunspell.sourceforge.net

terms of POS.

Table 5. Average words per sentence and occurrence per sentence (AOpS) of casual English

Avg. words per sentence	Avg. casual English words per sentence
14.63	3.019

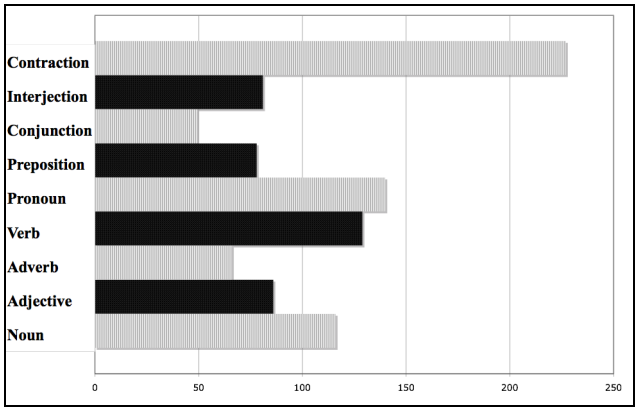

Fig 2. Casual English distribution by POS

As indicated in Table 2, the AOpS of casual English items is, at 20.66%, perhaps surprisingly low. It can be suggested that while this may be reasonably representative of Twitter users during late 2009, a different corpus like the compilation of SMS messages such as that used in [2] may yield a much higher percentage of casual English items.

Regarding the POS findings, as shown in Fig. 2, contractions were particularly common as this category included both acronyms (*LOL, OMG* etc.) and non-standard contractions such as *imma;* however, we also included standard contractions where the writer had omitted the apostrophe, which were extremely frequent (*im, dont, wouldnt,* etc.)

Prepositions commonly included *4* or *2* (for and to). A large number of the counts for pronouns were for *u* (you), with a majority of conjunctions being variants of "and" and "because" (*an, n, coz, cuz* etc). Counts for verbs were often *B* for "be", with a large number of gerunds (*in* or *in'* instead of writing the full *ing,* e.g. *dancin'). Frequently occurring adverbs were variants of *sooo, nw, hre* (so, now, here) and adjectives were varied, although vowel lengthening for emphasis was common, e.g. *goooood.* Nouns were also varied, but a particularly common token was *ppl* (people). Interjections very frequently used vowel lengthening for emphasis, e.g. *aaaarrrgh.*

Based on the discussion of various approaches in Section 6.1 and the results of the twitter data analysis, we propose to design a casual English generation system as follows. First, some superficial pre-processing such as lowercase conversion and URL detection/stripping will be conducted. As POS has shown to be somewhat influential in our analysis experiment, we will first use a POS tagger (using a parser such as Enju[11]) on the input sentences and select pronouns, verbs and nouns for conversion. Frequency per sentence will be set at 20%, in line with the experiment results.

Next, dictionary look-up using the previous system, CECS, will be used on a small set of common tokens with standard "slangifications" e.g. common contractions and interjections. Next, tokens selected according to the method described above will be split into phonemes using the CMU pronunciation dictionary[12]. Finally, these phonemes will then be converted using a manually compiled phoneme-to-phoneme database. The phonemic representation will be constructed based on analysis of casual English sentences and the large volume of examples collected during our research.

[11] http://www-tsujii.is.s.u-tokyo.ac.jp/enju/

[12] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

7. Conclusions

We have presented CECS, a 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 evaluator 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 proposed a method for automated generation of casual, irregularly-formed English used in communications such as Twitter. We explored the comparative advantages of a word-to-word database and a phoneme-to-phoneme database for converting regular English to casual English, and investigated the optimum proportion and distribution per sentence of casual English vocabulary for automatically producing humanlike creative sentences as an AI task.

References

[1] R. Sproat, A. W. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards. Normalization of non-standard words. *Computer Speech and Language*, 15(3) July 2001, pp..287–333.

[2] A. Aw, M. Zhang, J. Xiao and J. Su. A phrase-based statistical model for SMS text normalization. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, Sydney, Australia, July 2006, pp.33–40.

[3] C. A. Henriquez and A. Hernandez. 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.

[4] W. Wong, W. Liu and M. Bennamoun. 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. 55–62.

[5] K. Kukich. Techniques for automatically correcting words in text. *ACM Computing Surveys,* 24(4):377–439, December 1992.

[6] E. Clark, T Roberts and K. Araki. Towards a Pre- processing System for Casual English Annotated with Linguistic and Cultural Information. In *Proceedings of Computational Intelligence 2010*, Hawaii, August 2010.

[7] A. Clark. Pre-processing very noisy text. In *Proceedings of Workshop on Shallow Processing of Large Corpora,* Lancaster, UK, March 2003, pp. 12–22.

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

[9] M. D. Choudhury, Y.R. Lin, H. Sundaram, K. S.Candan, L. Xie and A. Kelliher. 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.