*Proceedings of the Fifth IASTED International Conference*
*Computational Intelligence (CI 2010)*
*August 23-25, 2010   Maui, Hawaii, USA*

# TOWARDS A PRE-PROCESSING SYSTEM FOR CASUAL ENGLISH ANNOTATED WITH LINGUISTIC AND CULTURAL INFORMATION

Eleanor Clark, Tyson Roberts, Kenji Araki
Language Media Laboratory, Graduate School of Information Science and Technology, Hokkaido University
Kita-ku, Kita 14 Nishi 9, 060-0814 Sapporo, Japan
E-mail: {eleanor, nall, araki}@media.eng.hokudai.ac.jp

## ABSTRACT

We present a preliminary revision of a text processing system, CECS (Casual English Conversion System) the purpose of which is to normalize the casual, error-ridden English that is frequently a feature of new media such as Twitter, into regular English. CECS has two applications: as pre-processing on input for Machine Translation or Information Retrieval systems, and as a standalone system to aid non-native speakers' reading comprehension of informal written English. The educational aspect of CECS is enhanced by the provision of manually compiled annotation on each word or phrase converted by the system. The system currently runs using a manually compiled database and a fairly straightforward text-to-text replacement method, but future plans include the implementation of a web mining algorithm for wider knowledge acquisition. Preliminary experiments produced positive results, suggesting that the basic concept and implementation of the system give it considerable potential as a pre-processing tool, and that the main task hereafter lies in the expansion of the database and addition of web mining and word-sense disambiguation automatic candidate selection algorithms.

**KEY WORDS**
Text Processing, Machine Translation, Educational Technology, Twitter

## 1. Introduction

### 1.1 Background

The surge in user-generated content on the Internet characterized by social networking sites, video-sharing sites, blogs and micro-blogging services such as Twitter[1], often referred to as the "Web 2.0[1]", uses less standardized language than traditional media, and frequently presents problems to non-native speakers of the language being used. In addition to posing a problem to readers, this kind of casual language, rich in creativity and individual user idiosyncrasy, is often a barrier to automated tasks such as Machine Translation (MT) and Information Retrieval (IR). Our research aims to address both these problems through the development of a text normalization or pre-processing system on casual English input, producing standardized, syntactically correct output that is more easily handled by both a human reader - particularly non-native learners of English whose access to web 2.0 material and new media is impeded by the highly irregular language used within - and automatic text processing systems.

### 1.2 Contribution of this paper

As seen in the previous research summarized above, a fully automated method, particularly the common SMT-based approach to this problem, has not yet shown fully effective results. Our research is a combination of automatic and manual approaches, using a manually complied and edited database for high accuracy and human knowledge in a text replacement system. The obvious drawback of the manually created database is that it cannot provide huge coverage at this early stage, but the research is a three-year project carried out as a collaboration between authors of both engineering and linguistics backgrounds, and later revisions of the system are expected to increase in coverage. In addition, we plan to implement a web mining-based technique at a later stage to find candidates for out-of-database items, thus vastly increasing the knowledge pool available.

The novel contribution of this paper is primarily the potential offered by this system as an aid to MT, IR or automatic summarization tasks which use new media such as Twitter as data. The effectiveness of the system is at present limited by the size and quality of the manually compiled database, which is at an early stage, but continues to be expanded. However, whereas similar research has attempted to fully automate

An original feature of the system is the linguistic and cultural information provided as annotation for the human user. This provides an educational aspect to non-native learners of English who wish to participate in new communications media but are excluded by the irregular language used within; words and phrases which frequently cannot be found in any dictionary. This system aims to address such barriers by not only "translating" the problematic text into more accessible English, but also

[1] www.twitter.com

providing information on each converted item as an aid to casual language study.

## 2. Related Work

Despite the expansion of MT-related research in recent years, particularly in the area of Statistical Machine Translation (SMT), which has now become the dominant paradigm [2], research aimed at the specific problem of automatically normalizing casual English is relatively rare [3]. Spelling error correction is a fairly well-established area, with initial pattern matching and *n*-gram analysis techniques having improved over the last two decades [4], but 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. In [5], a categorization system of errors and irregularities in casual English was proposed by E. Clark et al., which was later included in the system presented in this paper as annotation to each item in the database.

Research on unknown vocabulary items often focuses on the recognition and translation / transliteration of proper names; although Sproat et al. [6] 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" [6]. A. Clark's work on pre-processing a large collection of Usenet 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. [8] have produced a system for normalizing Short Message Service (SMS) 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 spelling, and punctuation omission, by creating a parallel corpora of 5000 raw and normalized English SMS messages and applying a phrase-based SMT model, resulting in a significantly boosted BLEU [9] score when 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.

In a similar vein, Henriquez et al.[10], in their work for the CAW2.0 project[2] 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".

This is not to say, however, that a statistical approach is not useful in problems related to this area. Salvetti et al. [11], when working on the problem of filtering spam weblogs (known as *splogs*) where the words had been "glued together as one token", developed a technique which segments long tokens into the words forming the phrase, based on statistical occurrence in training data. This initial segmentation was carried out prior to weblog classification, leading their approach to achieve accuracy similar to that of human evaluators.

Finally, with the rapid expansion of new media, the irregularity of language poses a barrier to various automated tasks other than the previously mentioned MT. 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 [12]. The interest in content of this type, both from researchers and corporations, shows a pressing need for effective text normalization of casual English.

## 3. System Overview

### 3.1 Database

The starting point for our present system was the creation of a database, initially in a simple word-pair format but later expanded to include the categorization system earlier proposed in [5], and notes on word sense disambiguation (WSD) problems for entries with two or more possible translations. The latter is a temporary measure designed to address the WSD issue simply by presenting the user with alternative translations of the input word; a later version of the system would ideally present the correct corresponding regular English word or expression through use of tools such as an improved tokenizer with context tagging.

The database is constructed manually by recording casual English vocabulary found in a variety of Web 2.0 sources[3], checking each item for absence in the UNIX dictionary[4], confirming regular English meaning with a

[2] "Content Analysis for Web 2.0" workshop held in Barcelona, Spain, 2009 http://caw2.barcelonamedia.org/

[3] Data was gathered from microblogging sites including http://twitter.com, online newspaper and media comment boards such as http://www.youtube.com, http://www.mailonline.com and others.

[4] Built into Apple's OSX operating system, containing 98,570 words, including proper names and plurals. This step is to confirm that the item

variety of dictionaries[5], and including E. Clark et al.'s categorization and notes on WSD problems as introduced in [5]. In addition to the initial eight categories of Shortform (abbreviation), Shortform (acronym), Typing error/ misspelling, Punctuation error/ omission, Non-dictionary slang, Cultural reference/ in-group meme, Wordplay/ intentional misspelling, and Omission of vocabulary, three further categories have been included. The first is Named Entity, which as yet, has very limited support; this would be vastly improved by the planned web mining function. The second and third new categories are Swear-word Censor Avoidance, to identify words that have been obscured in order to fool automatic censorship, and Emoticon/Visual Representation; some support has been given to frequently occurring emoticons, converting to the intended expression in parentheses, e.g. "(smiley face)", but this is not yet a fully comprehensive feature. It is written in Comma Separated Values (CSV) format and is primarily edited and updated using Microsoft Excel. At the time of publication it contains translations, categories and notes for 646 casual English vocabulary items, both single words and phrases. Database construction is ongoing.

### 3.2 System flow

CECS is written in Python. The flow of the system is illustrated in Fig. 1. Whereas the first revision of the code had used a simple whitespace delimiter – as is common in related research [7] [10] - to tokenize input into words, stripping and later replacing punctuation [13], the second revision tokenizes input using a strictly regular grammar defined in PyParsing [6] which defines words and punctuation as separate tokens, and allows combinations.

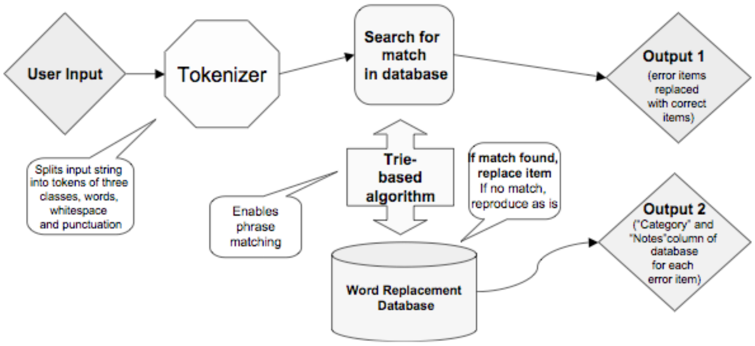

User Input
Tokenizer
Search for match in database
Output 1 (error items replaced with correct items)
Splits input string into tokens of three classes, words, whitespace and punctuation
Trie-based algorithm
Enables phrase matching
If match found, replace item If no match, reproduce as is
Output 2 ("Category"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

**Figure 1: Flow Chart of CECS**

"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. "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. 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.

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.

### 3.3 User interface

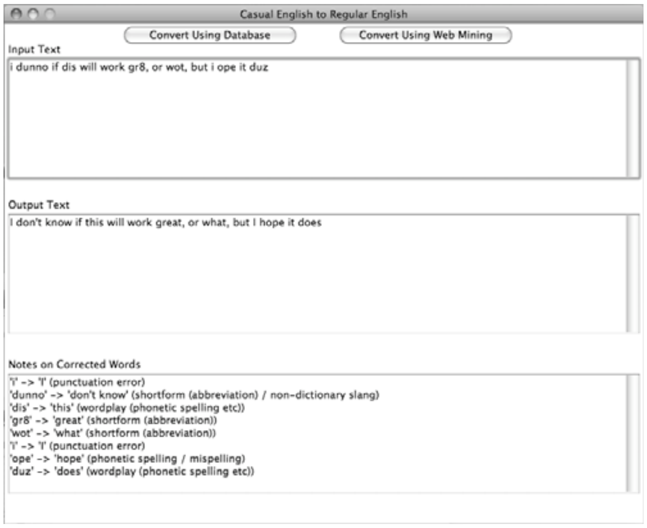
CECS uses a simple graphical user interface (GUI), which is shown in Fig. 2 in a Macintosh OSX-nativized appearance. The GUI was written using Pythoncard, a GUI construction kit using wxPython. The early development of a GUI was motivated by a wish for immediate ease-of-use in future user-based evaluations.

The GUI consists of three text areas: the input box in which the user writes a word, sentence or paragraph including casual English items (copy-paste is enabled); the output box, which displays a regular English "translation" of the input; and the notes box, which breaks down each casual English item into input word, translation, categorization, and provides notes on WSD problems if they are present in the database. The two buttons at the top of the GUI are currently labeled "Convert Using Database" and "Convert Using Web Mining". The first button tokenizes the input, searches for matches, and replaces the casual English items with the corresponding regular English in the database in the output box, as well as simply reproducing the database entry as one line in the notes box. The "Convert Using Web Mining" button is currently not linked to any functions but is planned to initialize a web mining process in future revisions.

is a reasonably "unknown" word and thus would pose a problem to non-natives.

[5] As many slang vocabulary items do not feature in standard dictionaries, we used user-edited and user-evaluated large-scale dictionary websites such as http://www.wiktionary.org and http://www.urbandictionary.com to confirm slang meanings.

[6] http://pyparsing.wikispaces.com/



i dunno if dis will work gr8, or wot, but i ope it duz

I don't know if this will work great, or what, but I hope it does

'i' -> 'I' (punctuation error)
'dunno' -> 'don't know' (shortform (abbreviation) / non-dictionary slang)
'dis' -> 'this' (wordplay (phonetic spelling etc))
'gr8' -> 'great' (shortform (abbreviation))
'wot' -> 'what' (shortform (abbreviation))
'i' -> 'I' (punctuation error)
'ope' -> 'hope' (phonetic spelling / mispelling)
'duz' -> 'does' (wordplay (phonetic spelling etc))

**Figure 2: CECS GUI Appearance**

# 4. Preliminary Evaluation Experiment

## 4.1 Preliminary experiment: System as pre-processing form machine translation

A small-scale preliminary evaluation experiment was carried out on CECS' potential as a pre-processing method for MT input. Sixty input sentences, of a total of 863 words, were passed through CECS, and the pre-processed output, as well as the original raw input for comparison, was tested in two popular free MT applications, Google Translate and Systran[7]. Of the 60 test sentences, 30 were 'known sentences' which had been used as training data for CECS (irregular vocabulary was manually entered into the database prior to evaluation), and 30 test sentences were 'unknown' sentences, where database coverage could be tested.

Input data was taken from the comments boards of the three most popular music videos on Youtube[8], as Youtube tends to attract highly non-standard error-ridden language and have a fairly international range of posters; input data featured a wide range English writing styles, not only American English or British English but also many examples of non-native English errors. Sentences were limited to between 5 and 20 words, and were selected from the most recent comments, although grammatically perfect sentences – rare in occurrence – and non-English comments were discarded.

The method of evaluation was set as counting the number of Non-Translatable Words (NTWs) when using both Google Translate and Systran's English to Japanese translation. This language pair was selected for two reasons: a) as a considerably problematic language pair, it provides a challenging test situation for CECS and b) it is the working language environment for the authors, and as such translation accuracy can be easily checked. Automatic evaluation metrics such as BLEU [9] were not used for this preliminary experiment, as these require a reference translation text for the MT output to be measured against. In CECS' case this would entail manual normalization of raw input, which may be subject to individual bias.

NTWs were defined as either: the reproduction of the input word in English (no translation into Japanese; most NTWs fell into this category), or the production of semantically completely unrelated Japanese words. MT output considered not to be NTWs were: correct Japanese translations, and partially incorrect or inappropriate Japanese translations that were semantically related to the input word. The latter rule's leniency was determined in accordance with the current quality of both Google Translate and Systran's free Japanese to English MT systems, which remains relatively low.

## 4.2 Preliminary evaluation results

Table 1 summarizes the results of the preliminary evaluation experiment. NTW counts were calculated for each sentence in four permutations: in raw input and CECS pre-processed forms, passed through both Google and Systran MT systems. NTW counts were then calculated as an average for all 60 sentences together, then for known and unknown sentences separately.

**Table 1: Preliminary Evaluation Results**

| Input type (number of sentences) | NTW count (average) | |
|---|---|---|
| | Google | Systran |
| All test sentences (60) | | |
| Raw input | 2.75 | 2.98 |
| CECS output | 0.3 | 0.38 |
| Known sentences (30) | | |
| Raw input | 2.7 | 3 |
| CECS output | 0.13 | 0.16 |
| Unknown sentences (30) | | |
| Raw input | 2.8 | 2.96 |
| CECS output | 0.46 | 0.6 |

On average, raw input sentences had an average of 2.75 NTWs with Google Translate and 2.98 with Systran, which was reduced to 0.13 (Google) and 0.16 (Systran) in the known sentences and 0.46 (Google) and 0.6 (Systran) in the unknown sentences.

The results show an overall reduction of 2.52, or 88.2%, in NTW occurrence after pre-processing with CECS, (89.1% when using Google, 87.3% with Systran). Known sentences performed considerably better, for obvious reasons of having high to full database to coverage, with a 95.2% reduction in NTWs on average in Google, and a 94.7% reduction when using Systran. However, unknown sentences also showed a marked reduction of 83.6% with Google and 79.8% with Systran, suggesting that CECS'

[7] http://translate.google.com/, http://www.systranet.com/

[8] http://www.youtube.com. At the time of the evaluation experiment, these were videos by American recording artists Lady Gaga, Justin Bieber and Rihanna.

database gives reasonable coverage even at this early stage.

One reason for the difference in Google Translate and Systran's NTW count results are that the two systems appear to have some limited recognition, though widely different in coverage, of common errors and slang forms; the phonetic spelling for emphasis 'sooo' (up to 4 'o's were recognized), the slang acronyms 'lol' (*laugh out loud*) and 'btw' (*by the way*) were all correctly translated by Google, whereas Systran was not able to recognise these items. Conversely, Systran recognized the punctuation omissions 'dont' and 'doesnt' and the misspelling "becuse" whereas Google did not, although neither system was able to identify the punctuation omission "shes".

### 4.3 Problems and potential solutions

Table 2 shows an example of NTW problems in MT output. NTWs are obvious even to a non-Japanese reader, as they have mostly been reproduced simply in the original English input form.

**Table 2: Example of NTW and problem identification**

| | |
|---|---|
| Raw input: | i dont kno why but after seein the vid i wanna buy a g shock |
| Google MT: | ため息 i dont はカチン理由が後に私の VID ショックを ag は買いたいと思う (5 NTWs) |
| Systran MT: | 私は seein 私が g の衝撃を買いたいと思う vid 後 kno なぜが (4 NTWs) |
| CECS output: | I don't know why but after seeing the video I want to buy a g shock |
| Google MT: | 私はなぜかわからないが私はショックを ag は購入したいビデオを見た後 (1 NTW) |
| Systran MT: | はビデオを見た後私が g の衝撃をなぜ買いたいと思うが、かか知りません (1 NTW) |

This sentence, part of the known sentences category (as a result, CECS coverage of irregular words is high), produces most of its NTWs due to abbreviated slang shortforms (*kno, seein, vid*) which are all in the database. However, the named entity 'g shock' (a brand digital watch model) is not included, and as such remains a NTW after CECS pre-processing.

Below the most frequently occurring reasons for NTWs remaining after CECS pre-processing are given, with suggestions on how to solve these problems in future revisions of CECS.

a. Named entity: The most logical solution to identification of named entities is with the addition of a web mining algorithm to CECS. Compiling a manual database of named entities would be an extremely laborious task, whereas online sources such as Wikipedia could be mined for information. If a match is found through web mining for an NTW not in the database, hyperlinks could be given in the "Notes" output of the GUI for the human user.

b. Out-of–database (OOD) item: All OOD slang and wordplay items that occurred during this experiment were later added to the database. Web mining would help solve this problem in the case of OOD slang, but most occurrences were creative spelling or typing errors. The integration of a spellchecking tool such as GNU Aspell[9] into CECS would help identify OOD spelling errors and produce automatic candidates for correction.

c. Grammatical error: Lack of punctuation in the input used in this experiement was the biggest general cause of low translation quality. Automatic punctuation insertion and grammatical correction is a non-trivial task and possibly outside the scope of this project; an SMT-based approach may be useful in this case. Again, the integration of other available tools would be a desirable solution.

d. Not possible to normalize: In some cases, it is impossible even to manually normalize the item with confidence, as the writer's intended meaning is unclear. The best solution would seem to be a "closest likely candidate" approach. These cases appear to have spelling or typing errors as a major cause, so the previously-mentioned integration of Aspell may have some effect. However, it seems likely that there will always be some input that is simply indecipherable to anyone but the original author.

## 5. Conclusion

In this paper we have presented a method for normalizing casual, error-ridden English frequently found in online communications, with linguistic and cultural information provided as annotation. Our system, CECS, is in a very early stage, with a limited database of around 650 items, but has shown positive results in our preliminary evaluation experiment, reducing Non-Translatable Words (NTWs) occurring in the English-to-Japanese versions of Google Translate and Systran's online MT applications by 89.1 and 87.3% respectively. This indicates that CECS' combination of manual and automated approaches has potential for becoming a useful pre-processing method for automated tasks that require the cleaning of noisy text such as MT and IR. Further expansion of the database and improvements to the code, as well as large-scale evaluation (both user and automatic) experiments are necessary.

## 6. Future Work

As the scale of CECS' first evaluation experiment was somewhat limited, future work will need to include a

[9] http://aspell.net/

wider range of evaluation methods, for example human user-evaluated experiments, on a much larger sample of data. Ritter et al. have announced the forthcoming availability of their Twitter corpus (a collection of 1.3 million Twitter conversations), which we hope to utilize as input in future large-scale experiments.

In the near future, we plan to make improvements in code, specifically the implementation of a web mining algorithm for identification of named entities and OOD items, the integration of an open-source spellchecker such as Aspell, and the capability for automatic selection of WSD candidates, as well as the continuous task of the expansion of the database.

## References

[1] T. O'Reilly, "What is Web 2.0", http://oreilly.com/web2/ 2005, last accessed May 17 2010

[2] A. Lopez, Statistical machine translation. In *ACM Computing Surveys,* 40, 3, Article 8, 2008, pp. 1-49

[3] W. Wong, W. Liu and M. Bennamoun, Enhanced Integrated Scoring for Cleaning Dirty Texts, *Proceedings of IJCAI-2007 Workshop on Analytics for Noisy Unstructured Text*, India, 2007 pp. 55-62

[4] K. Kukich, Techniques for Automatically Correcting Words in Text, *ACM Computing Surveys, Vol. 24, No. 4,* 1992, pp. 377-439

[5] E. Clark and K. Araki. A Proposal for an Automatic Linguistic Pre-processing System of Casual English for MT Use, *Language Acquisition and Understanding Research Group (LAU) Technical Reports,* Sapporo, 2009, pp. 1-5

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

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

[8] A. Aw, M. Zhang, Z.Z Fan, P.K. Yeo and J. Su, A Phrase-based Statistical Model for SMS Text Normalization, *Proceedings of the COLING/ACL 2006 Main Conference Poster Session,* Sydney, 2006, pp.33-40

[9] K Papineni, S. Roukos, T. Ward, and W.J. Zhu, BLEU: a method for automatic evaluation of machine translation, *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics,* 2002, pp. 311–318

[10] C. A Henriquez and A. Hernandez, A Ngram-based Statistical Machine Translation Approach for Text Normalization on Chat-speak Style Communications, *Proceedings of CAW2.0 2009* Madrid, 2009 pp. 1-5

[11] F. Salvetti and N. Nicolov, Weblog classification for fast splog filtering: □a URL language model segmentation approach, *Proceedings of the Human Language Technology Conference of NAACL*, *Companion Volume*, New York, 2006, pp. 137-140

[12] A. Ritter, C. Cherry and B. Dolan, Unsupervised Modeling of Twitter Conversations, *Proceedings of HLT-NAACL 2010,* Los Angeles, CA, 2010, in print

[13] E. Clark and K. Araki, A Basic Annotated Linguistic Pre-processing System for Casual English with Ideas for Expansion, *Proceedings of GCOE-NGIT 2010*, Sapporo, 2010, pp. 184-185