An Example-based Japanese Proofreading System for Offshore Development

Yuchang CHENG Tomoki NAGASE
Speech & Language Technologies Laboratory of Media Processing Systems Laboratories
FUJITSU LABORATORIES LTD.
4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki, Kanagawa 211-8588, Japan
cheng.yuchang@jp.fujitsu.com, **nagase.tomoki**@jp.fujitsu.com

ABSTRACT

More than 70% of Japanese IT companies are engaged in the offshore development of their products in China. However, a decrease in the quality of the accompanying Japanese engineering documentation has become a serious problem due to errors in Japanese grammar. A proofreading system is therefore required for offshore development cases. The goal of this research is to construct an automatic proofreading system for the Japanese language that can be used in offshore development. We considered an example-based proofreading approach that can effectively use our proofreading corpus and simultaneously process multiple types of error. There are three main steps in the proofreading system. They are the search step, the check step and the replace step. We will make a demostration for the proofreading system and simulated the use of our example-based approach. The results show that using the entire corpus can reduce the errors by over 66%.

Title and abstract in another language (Japanese)

オフショア開発向けの日本語自動校正システム

概要

近年，日本企業からのオフショア開発が増加傾向にある．オフショア開発では，外国人の執筆する日本語文書の品質確保が課題であり，校正作業がコストを押し上げている．そのため，自動校正システムの実用化が期待されている．本研究ではオフショア開発向けの日本語自動校正システムの開発を目的として，校正履歴コーパスを効果的に利用して，1文中で複数の誤用を同時に検出できる事例ベースの校正手法を提案した．事例ベースの校正システムは校正履歴コーパスを持つ．校正処理は，まず，処理対象文の単語依存構造をキーとし，校正履歴コーパスに同じ単語依存構造を持つ事例を検索する．その後，検索結果の適用候補事例に対し，単語の表記と意味概念情報を用いて事例が処理対象文の校正に適用できるかどうかを確認する．最後は，チェックされた適用事例候補を用いて，校正事例の修正方法と同様に処理対象文を校正する．提案手法の効果を検証するため，本デモはは事例ベースの自動校正手法を展示し、その効果をシミュレーションした．その結果，校正履歴コーパス全体を事例として本手法を適用することにより，誤り全体の66%を校正できることを示す．

KEYWORDS : Error correct system, proofread system, offshore development, detect the misuse in Japanese
KEYWORDS IN L_2 : 文章校正，日本語誤用の検出，オフショア開発

Proceedings of COLING 2012: Demonstration Papers, pages 67–76,
COLING 2012, Mumbai, December 2012.

1 Condensed Version in L2－オフショア開発向けの日本語自動校正システム

本稿では，オフショア開発向けの日本語自動校正システムの開発を目的として，校正履歴コーパスを効果的に利用して，1文中で複数の誤用を同時に検出できる事例ベースの校正手法を提案した．

我々は中国人技術者の執筆した日本語技術文書の校正履歴をもとに，外国語母語話者による日本語誤用パターンの分析を行った(Cheng, 2012)．TABLE 2に誤り分類の種類と定義をまとめる．技術文書の校正において，最も誤りの頻度が高い助詞変更の校正が最重要課題である．しかし，助詞の校正が文の他の部分に影響を受ける場合や語彙誤用など校正方法が文脈に依存する場合，人手で誤りのパターンを一般化して校正ルールを作成することが困難である．そこで，我々は校正履歴をそのままシステムが読み込んで文書校正が動作する事例ベースの校正手法を考案した．事例ベースの校正手法では大量の校正例が必要であるが，実際のオフショア開発会社において業務の中で蓄積された大量の校正履歴を入手することにより，この校正手法を実現できる．

デモを行う事例ベースの校正システム（FIGURE **1**参照）はTABLE 1のような事例からなる校正履歴コーパスを持つ．校正処理は，1) 事例の検索，2) 事例のチェック，および 3) 対象文の書き直しのステップからなる．

まず，ステップ 1) では，処理対象文の単語依存構造をキーとし，校正履歴コーパスに同じ単語依存構造を持つ事例を検索する．検索条件によって複数の事例が適用候補になることがある．

ステップ 2) では，ステップ1の検索結果の適用候補事例に対し，事例が処理対象文の校正に適用できるかどうかを確認する．この場合は，処理対象文と校正事例（修正前の文）との共通部分が事例中の校正部分に似ているかどうかを確認する．適用可否を判定する際，単語の表記と意味概念情報を用いて対応部分の一致度を計算する．一致度の計算には単語の表記と意味概念情報に関する係数があり，係数は会社や業種に依存する．文の重複部分に修正が含まれない候補事例は処理対象文の校正に適用できないと判断され，適用事例候補から削除される．

ステップ 3) では，ステップ 2でチェックされた適用事例候補を用いて，校正事例の修正方法と同様に処理対象文を校正する．

事例ベースの校正手法の最大効果を調べるため，事例ベース手法の再現率に関するシミュレーションを行った（TABLE 3，TABLE 4参照）．異なる事例数をランダムで選択し，校正ステップに従いテスト事例に対する再現率を測った結果，校正コーパス全体を使用すると，誤用の66%が校正できることが判明した．

Before proofreading sentence:	引数***の***エンコード***転換は***されていない (There is no code-converting of the argument.)
After proofreading sentence:	引数***が***エンコード***変換[DELETE]***されていない (The argument is not code-converted.)

TABLE 1 – a proofreading example of the corpus.(校正履歴の例)

Category	Definitions of the proofreading types(校正の分類)	Count
Category 1 Proofreading the errors that are interfere with understanding context (文脈の理解に支障が出る誤りの校正) (Total count: 1060 / 8404 = 11%)	Erratum and omission of a word (誤字，脱字)	316
	Alphabet misspelling (スペルミス)	49
	The ambiguity between Chinese and Japanese (日中混同)	142
	Voiced sound, long vowel, and mis-pronunciation (濁音，長音，誤発音)	239
	Semantic mistake of words (意味誤り)	255
	Kana-kanji conversion mistake (かな変換誤り)	59
Category 2 Proofreading the errors that Chinese native speakers usually commit (中国語母語話者が犯しやすい誤りの校正) (Total count: 5096 / 8404 = 53%)	Particle addition (助詞追加)	720
	Particle deletion (助詞削除)	401
	Particle change (助詞変更)	2907
	Verb tense and aspect (動詞時制とアスペクト)	205
	Active and passive of verbs (能動と受動)	290
	Confusion of noun phrase and phrasal verb (名詞句と動詞句の混同)	573
Category 3 Proofreading inappropriate expressions in engineering documents (技術文書として不適切な表現の校正) (Total count: 2223 / 8404 = 23%)	Chinese character, hiragana, and declensional kana ending (漢字，ひらがな，送り仮名)	674
	Colloquialism (口語)	187
	Figure and unit (数字と単位)	123
	Formal speech/Casual speech (敬体常体)	76
	Technical terms (専門語)	267
	Vocabulary meaning (語彙意味)	896
Category 4 Proofreading the incomprehensible sentence structure and logic (文構造と論理の校正)(Total count: 1265 / 8404 = 13%)	Shortening of verbose text (冗長短縮)	350
	Sentence structure correction (文構造修正)	809
	Information addition (情報追加)	106

TABLE 2 – proofreading types and the count of correction history. (誤り分類の定義と頻度)

Category	distribution
Category 1	11%
Category 2	51%
Category 3	21%
Category 4	17%

TABLE 3 – the distribution of the testing data (テストデータの分布)

Corpus size	Sim 1	Sim 2	Sim 3	Sim 4	Sim 5
0	0.0%	0.0%	0.0%	0.0%	0.0%
1000	33.3%	32.7%	30.1%	34.2%	36.3%
3000	49.7%	53.2%	47.1%	55.8%	52.9%
5000	57.6%	58.5%	57.3%	62.3%	59.9%
8082	65.8%	65.8%	65.8%	65.8%	65.8%

TABLE 4 – the result of the simulation (シミュレーション結果)

2 Introduction

With the advancement of corporate globalization, the outsourcing of system development to foreign countries (i.e., offshore development) has increased in IT companies. More than 70% of Japanese IT companies currently have the offshore development of their products in China. With respect to offshore development in China, there is an increase in cases where native Chinese engineers are employed by the offshore vendor in both the software development phase and the design phase. This means that a large proportion of the engineering documentation, such as the specifications and technical reports, are created in Japanese by Chinese native engineers. Generally, the engineers who prepare the documentation are very proficient in Japanese. However, this has been accompanied by a decrease in the quality of the engineering documentation due to misuse of the language used by the purchaser, and the purchaser is required to manually proofread the engineering documentation. To reduce the cost of manually proofreading, there is a need for the development of a "document proofreading system" that automatically proofreads the documentation in the language of the purchaser. The goal of this research is to construct an automatic proofreading system for Japanese which can be utilized for offshore development.

Recently, proofreading technologies (or error detection, correction) have been considered as applied technologies for the machine translation and the language education. Many recent studies have focused on proofreading for English as a Second Language (ESL) learners (Izumi et al., 2003; Han et al., 2006; Gamon et al., 2008; Gamon, 2010). Other researches (Oyama and Matsumoto, 2010; Imaeda et al., 2003; Nampo et al., 2007; Suzuki and Toutanova, 2006; Mizumoto et. al, 2011) focus on errors that are more common to Japanese learners, such as case particles. The error correcting corpora used in previous works (regarding Japanese as a Second Language (JSL)) was acquired from essays, examinations, and social network services. These corpora include all types of error made by all levels of Japanese "learners." It is impractical to cover all such in the construction of a proofreading system. We assume that there are limited types of error made by the native Chinese engineers, and concentrate on some specific categories (because the engineers are not Japanese "learners").

We had analyzed a Japanese proofreading corpus that provides a history of proofreading for offshore development in China (Cheng, 2012). According to our findings, most types of errors mentioned in the proofreading corpus relate to the misuse of particles. However, the misuse of particles usually occurs together with other types of errors in the same sentence (see TABLE 1), and it is difficult to define general rules for the proofreading of these multiple types of errors. In this demo, we will make a demostration of an example-based proofreading approach for the multiple types of errors. This example-based approach requires a sample collection, and our proofreading corpus can be directly used for the example-based approach. We can adopt the example-based approach in English or any other language, as long as there is a proofreading corpus in the language.

3 An introduction of Japanese proofreading corpus in offshore development

We had analyzed the Chinese native engineers' misuse tendency of Japanese in the proofreading corpus (Cheng, 2012). The corpus is a history of proofreading written by a native Japanese proofreader who has experience in the correction of engineering documents prepared by native Chinese engineers in offshore development. Our proofreading corpus includes 8404 examples, which were collected from 519 documents. These documents were prepared by 20 engineers who have successfully passed the N1 level of the Japanese-Language Proficiency Test (JLPT:

http://www.jlpt.jp/e/guideline/testsections.html). We assume that the error tendencies noted in these documents normally occur in all of the engineering documentation in the offshore development industry.

A proofreading example contained in the proofreading corpus is shown in TABLE 1. The proofreading example includes the before proofreading sentence and the result after manual proofreading. Many of the proofreading examples involved multiple types of error like this example in the proofreading corpus. We classified the proofreading examples and investigated the distribution of the proofreading types in the corpus.

TABLE 2 shows the distribution of the proofreading corpus. The largest category of the proofreading is Category 2 that occupies about 53% (5096/9644) more than the entire half. Category 2 includes the proofreading of particles and the verb. This observation is similar to previous work (Oyama, 2010), but we found that the ratio of this type of error in the proofreading corpus is more than in Japanese learner's error data. The next largest is Category 3 that occupies the entire 23% (2223/9644). Category 4 accounts for 13% of the whole, and Category 1 accounts for 11% of the whole. Because Category 2 errors occur most frequently, we know that although the engineers have high Japanese proficiency, it is difficult to become proficient in the usage of particles and verbs.

4 The Demostrating System – An Example-based proofreading approach

Many examples of our proofreading corpus include multiple types of errors in a single sentence. It is difficult to introduce rules for proofreading multiple types of errors. By contrast, our corpus is not large enough for normal machine learners, because some proofreading examples occur only once and it causes the data sparse problem. To effectively use our proofreading corpus, we considered an example-based proofreading approach instead of using the machine-learning approach.

4.1 The system flowchart

FIGURE **1** shows the system flowchart and the process of proofreading. This system includes a proofreading corpus, which includes the original sentence (it includes error or misuse) and the proofreading result. The system proofreads wide types of errors and misuse by searching the corpus to find the useful examples. There are three main steps in the proofreading system. They are the search step (the part ③ in FIGURE **1**), the check step (the part ④ in FIGURE **1**) and the replace step (the part ⑤ in FIGURE **1**). The flow of the proofreading approach is described as the following paragraph.

The target proofreading documents are inputted into the system, and then the system divides the document to sentences and processes the sentences respectively (the part ① in FIGURE **1**). Then the system will do several processes to require information for proofreading (the part ② in FIGURE **1**).

These processes include morphological analysis, dependency parsing and the Semantic analysis. In the part ③ (search step), the system searches the proofreading corpus to find the useful examples for the proofreading, here the system will use the morphological and dependency information to search. The search results possibly have more than one example.

In the part ④, the system checks the search example in search step by estimating the similarity of the words in the target sentence and the proofreading examples. If there is no similar example

in part ③ and ④ (after check step, it is possible that the searching results are rejected), the system will back to the part ② to process the next target sentence. If the search results pass the check step, the proofreading example can be used to proofread the target sentence.

In part ⑤ (replace step), the system will refer to the before sentence and the after sentence of the proofreading example to proofreading the target sentence. This means that the system will do "similar" proofreading to the target as the proof-reader was done in the example. Then the system outputs the proofreading results of the target sentence. In next section, we describe more detail about the main steps of the proofreading system.

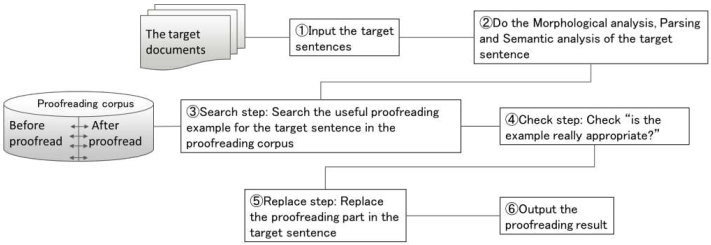

FIGURE **1** – The flowchart of our example-based proofreading system.

4.2 The main steps of the example-based approach

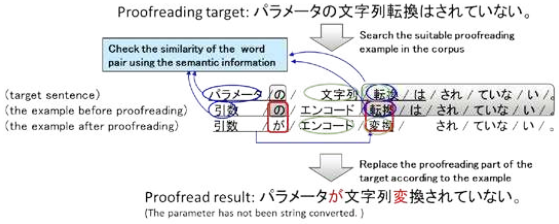

FIGURE **2** – an example of the example-based proofreading approach

FIGURE **2** shows an example of the example-based proofreading approach. The input sentence "パラメータの文字列転換はされていない. (The parameter has not been string converted.)" is the proofreading target. The system analysed the target sentence, than searched the corpus and found a possibly useful example "Before: 引数のエンコード転換はされていない → After: 引数がエンコード変換されていない(The argument is not converted the encoding.)". Then the system proofread the target sentence using the similar replacement in the example. That is, changing the particle "の(no)" in the target to the particle "が(ga)", changing the word "転換(convert)" to the word "変換(convert)", and deleting the particle "は(ha)". Therefore, the target sentence became to "パラメータが文字列変換されていない(The parameter has not been string converted.)". In our approach, if the proofreading example occurs once in the corpus, the

system can use the example to proofread a new “similar” sentence. Therefore this approach can use the proofreading corpus efficiency.

Search Step: “Is there any useful proofreading example for the target sentence in the proofreading corpus?”

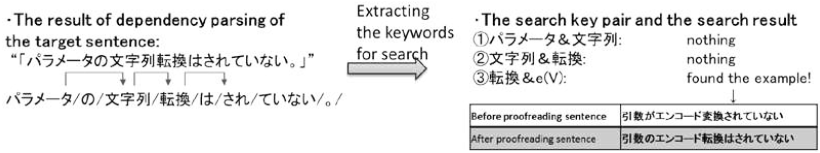

FIGURE **3** – The search key word of the target sentence and its rearch results

In this step, the system uses the dependency analysis results of the target sentence. FIGURE **3** shows the search keywords and the search result. The system used the substantives and the declinable words that have dependency relations in the target sentence to search the corpus. The proofreading examples in the corpus should also be analyzed with respect to the dependency structure and semantic structure. It should be noted that the system does not only search the string of the keywords, but also searches the morphological information and semantic information of words, such as the keyword pair ③. If the before proofreading sentence has a dependency relation that is similar to the keyword pair, the example is selected as a candidate for proofreading the target sentences. FIGURE **3** has only one search result, which is the example in TABLE 1. If there is no search result, the system reverts to part ② in FIGURE **1** to process the next target sentence.

Check Step: “Is the example really appropriate?”

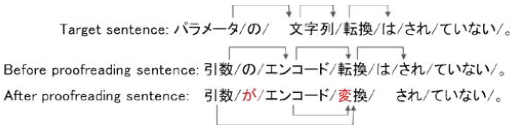

FIGURE **4** – The dependency structure of target, and before / after sentences

After searching the corpus, some (possibly) useful examples for proofreading were found. However, not all of these examples are useful for proofreading. In this step, the system checks two conditions regarding the example. The conditions are “Is the example similar to the target sentence?” and “Can the target replace the example?” Considering the example in FIGURE **4**, the target sentence should be similar to the before proofreading sentence. Also, there should be parts of the target sentence that can be replaced to change the before proofreading sentence to the after proofreading sentence.

For checking the first condition, the system considers the similarity of the corresponding words in the dependency structure between the target sentence and the before proofreading sentence. In this case, the system checked the word pair “パラメータ (parameter) / 引数 (argument)” and “文字列 (string) / エンコード (encoding)”. The similarity of the word pair is calculated according to the following equation:

- $\text{Similarty} = \alpha \times \text{Txt} \div \text{WordLen} + \beta \times \text{Syn} + \gamma \times \text{Sem}$

Where:

- α, β, γ : The coefficients that can be changed for different proofreading corpus and manual tuning
- Txt: The edit distance between the words
- WordLen: The count of the character of the words
- Syn: The distance between the words in the dependency structure.
- Sem: The distance of the semantic class between the words, it can be tuning manually

If the similarity is smaller than a threshold value, the proofreading example will be excluded. The threshold value can be set manually for different situation, such as the different industry, company or project. In this case, the word pair "パラメータ (parameter) / 引数 (argument)" and "文字列 (string) / エンコード (encoding)" have similar usage in this offshore vendor.

To check the second condition, the system compares the morphological sequences of the before proofreading sentence and the after proofreading sentence. In FIGURE **4**, the sub-sequence "引数 (argument) / の(no) /" is changed to "引数(argument) / が(ga)/", and the sub-sequence "転換 (convert) / は(ha) /" is changed to "変換(convert)". Then, the system can use the sub-sequence in before proofreading sentence to rewrite the sub-sequence in the target sentence. If there is no need for the rewritten sub-sequence in the proofreading example, this example will be excluded.

Replace Step: Replace the proofreading part in the target sentence

After the checking step, the remaining examples can be used to proofread the target sentence. The sub-sequence that is rewritten in the proofreading example can be used for proofreading. Considering the case in FIGURE **2**, the system can proofread the words "パラメータ/の/" to "パラメータ/が/", and the sub-sequence "引数 (argument)/の(no)/" is changed to "引数/が(ga)/". Then, the system replaces all replaceable sub-sequences and outputs the proofreading result "パラメータ***が***文字列***変換***されていない. (The parameter has not been string converted)."

5 System performance – a simulation

As described in section 4.2, the system requires several coefficients for the check step. However, the coefficients and threshold value need to be tuned, but this is currently difficult, as more examples are required for tuning. In this paper, we made a simulation that can estimate the upper limit of the recall. This simulation followed the approach that we described in section 4, but the check step is performed manually. That is, when the system checked the similarity between words, we judge the word pair manually.

The testing data, which includes 324 examples, is a part of our proofreading corpus. The distribution of the testing data is shown in TABLE 3 and is similar to the distribution of the whole corpus. The remaining part of the corpus (8080 examples) is used to proofread the testing data.

We repeated the simulation five times, and the results are shown in TABLE 4 (from the column "Sim 1" to "Sim 5"). To investigate the relationship between the scope of the proofreading and the size of the proofreading corpus, we randomly selected sentences in several sizes from the proofreading corpus in each simulation. The sizes are shown in the first column in TABLE 4. TABLE 4 shows that use of the entire corpus can reduce 66% of the errors. The proofreading result obtained by using a random part of the corpus is homogeneous. We can consider that the distribution of the entire corpus is also homogeneous.

References

Yuchang, Chcng and Tomoki, nagase. (2012). Analysis of proofreading history intended for Japanese engineering document that foreign language speaker written (In Japanese). In *Proceedings of the 18rd Annual Meeting of the Association for Natural Language Processing*, pages 34–37, The Association for Natural Language Processing (Japan)

Elghafari, A., Meurers, D., & Wunsch, H. (2010). Exploring the data-driven prediction of prepositions in English. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters, (COLING'10)*, pages 267–275, Stroudsburg, PA, USA, Association for Computational Linguistics.

De Felice, R., & Pulman, S. G. (2008). A Classifier-Based Approach to Preposition and Determiner Error Correction in L2 English. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 169–176, Manchester, UK: Coling 2008 Organizing Committee.

Gamon, M., Gao, J., Brockett, C., & Klementiev, R. (2008). Using contextual speller techniques and language modeling for ESL error correction. In *Proceedings of 4th International Joint Conference on Natural Language Processing (IJCNLP 2008)*, pages 449-456, Hyderabad, India, Asian Federation of Natural Language Processing.

Han, N.-R., Chodorow, M., & Leacock, C. (2006). Detecting errors in English article usage by non-native speakers. *In Journal Natural Language Engineering archive, Vol. 12, Issue 2*, pages 115-129, Cambridge University Press New York, NY, USA

Koji Imaeda, Atsuo Kawai, Yuji Ishikawa, Ryo Nagata, and Fumito Masui. (2003). *Error Detection and Correction of Case particles in Japanese Learner's Composition (in japanese).* In Proceedings of the Information Processing Society of Japan SIG, pages 39–46.

Hiromi Oyama and Yuji Matsumoto. (2010). *Automatic Error Detection Method for Japanese Case Particles in Japanese Language Learners.* In Corpus, ICT, and Language Education, page 235–245.

Hiromi Oyama. (2010). Automatic Error Detection Method for Japanese Particles. Polyglossia Volume 18, (p 55-63).

IPA (2012). *the Annual report of IT Human Resources Development in 2012 (in Japanese)* , http://www.ipa.go.jp/jinzai/jigyou/docs/ITjinzai2012_Hires.pdf

Izumi, E., Uchimoto, K., Saiga, T., Supnithi, T., & Isahara, H. (2003). Automatic error detection in the Japanese learners' English spoken data. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL'03)*, pages 145–148, Stroudsburg, PA, USA.

Mizumoto, T., Komachi, M., Nagata, M., & Matsumoto, Y. (2011). Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners. In *Proceedings of 5th International Joint Conference on Natural Language Processing (IJCNLP 2011)*, pages 147–155, Chiang Mai, Thailand: Asian Federation of Natural Language Processing.

Suzuki, H., & Toutanova, K. (2006). Learning to predict case markers in Japanese. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (ACL'06)*, pages 1049–1056,

Stroudsburg, PA, USA, Association for Computational Linguistics.

Tetreault, J., Foster, J., & Chodorow, M. (2010). Using parse features for preposition selection and error detection. In *Proceedings of the ACL 2010 Conference Short Papers (ACLShort'10)*, pages 353–358, Stroudsburg, PA, USA, Association for Computational Linguistics.