Computerized Dynamic Assessment for Pragmatic Competence in Second Language Learners' English Email Requests*

John Blake$^{1[0000-0002-3150-4995]}$, Maxim Mozgovoy$^{1[0000-0003-0684-4012]}$, and Allan Nicholas$^{1[0000-0002-6178-3171]}$

University of Aizu, Aizuwakamatsu, 965-8580, Japan
{jblake,mozgovoy,anich}@u-aizu.ac.jp

Abstract. Here we report on the development of the first Computerized Dynamic Assessment (C-DA) platform for pragmatic failure in request emails written by Japanese learners of English. Dynamic language assessments are usually dyadic interactions between the tutor and the learner. In our online C-DA, we harness natural language processing algorithms to detect specific instances of pragmatic failure in learner email texts. On matching, graduated feedback in the form of increasingly more explicit hints is displayed sequentially until either learners have appropriately revised the text or the allocated number of attempts is reached. The C-DA incorporates researcher- and learner-facing interfaces, automatically generates progress reports for learners, and tracks all activities for research purposes.

Keywords: Dynamic Assessment · Computerized Dynamic Assessment · Graduated Feedback · Sociocultural Theory.

1 Introduction

1.1 Dynamic assessment

Dynamic assessment (DA) for language learning is an approach to learning and assessment that aims to simultaneously evaluate a learner's abilities and at the same time promote development in the learner [20]. While traditional non-dynamic assessments seek to assess a learner's abilities in isolation from others, DA is embedded in a sociocultural perspective on development, in which learning is fundamentally social in nature [24]. A central sociocultural concept relevant to DA is the Zone of Proximal Development (ZPD), which can be operationally defined as the space between what a learner may be able to accomplish with the aid of an expert other, and what they can accomplish independently. For Vygotsky [24], assistance will only reliably lead to learner development if

* Supported by a JSPS *Kakenhi* (Grants-in-aid for Scientific Research), Grant Number 19K00826, "Developing a diagnostic language assessment of Japanese EFL learners' English language email writing".

it falls within the learner's ZPD. To this end, assistance should be both graduated – initially minimal assistance provided – and only increased gradually, as appropriate – and contingent, being provided only when necessary [2]. Studies typically operationalise this approach by providing a range of feedback graded in terms of explicitness, with highly implicit assistance being initially employed. Assistance is then gradually made more explicit until the learner is able to successfully resolve a problem and continue with a given language task. In a typical DA session, a learner and expert mediator collaborate one-to-one on a language task. When the learner encounters some difficulty, the mediator may engage in the provision of assistance that is developmentally sensitive to the learner's individual ZPD in this way. By attending to the frequency of the assistance given, and its explicitness, insights can be gained into the learner's development within their ZPD. For example, a learner requiring only implicit assistance may be close to the ability to perform the task independently (in sociocultural terms, self-regulation [2]); the provision of explicit assistance, however, may indicate that the learner is still reliant upon the mediator to collaboratively complete the task (other-regulation [2]).

One criticism of DA is its time- and labour-intensive nature, with in-person one-to-one sessions requiring considerable resources. The creation of a computerised dynamic assessment (C-DA) offers a scalable solution, allowing multiple users to simultaneously undertake DA. To create such a system, Natural Language Processing (NLP) needs to be drawn upon to identify the target linguistic features. Once those features are identified, a prepared series of feedback hints sequenced by degree of explicitness can be initiated, which deliver feedback until the user is able to resolve the issue.

1.2 Pragmatic Features and Failure

The text type selected is request emails, this challenging genre [10], in which a learner needs to make a request to an email receiver, is particularly relevant for Japanese university students who need to communicate with non-Japanese speaking faculty [18]. The C-DA focuses on pragmatic failure specifically – text elements that do not adhere to community norms and conventions regarding appropriate levels of formality, request directness or email conventions. This may be because the sender has not adapted their language choices to varying social contextual variables of Power (akin to social status), Social distance (the level of acquaintanceship between sender and receiver) and the Rank of imposition (how potentially troublesome the email request may be for the receiver) [5]. Email pragmatic features are typically categorised in terms of framing moves and content moves which the sender can draw upon when making pragmatically appropriate language choices [7]. Framing moves refers to the opening and closing portions of the email text. An opening may include a greeting ("Dear..."), the receiver's title, if appropriate ("Prof.") and name. There may also be a self-introduction if appropriate for the situation. An email closing may comprise a pre-closing statement ("I look forward to hearing from you..."), a complementary closing ("Kind regards...") and signature [7, 19]. Content moves, on the other hand, refer to those aspects of the text that serve to realise the purpose of the email, which, in the current study, is a request. These moves primarily serve to soften the request, and may be within the request head act (the portion of text in which the actual request is produced), or outside of it, placed before or after. Within the head act, content moves may be relatively direct, such as an imperative ("Give me..."), or a *want* statement ("I want you to..."), while more indirect moves may also be employed ("could you possibly...?"). Content moves that soften the request outside of the head act may include, for instance, grounders (providing a reason for the request), an apology ("I'm sorry to bother you...") or a preparator (hinting at the request to come; [7, 19]).

In the example email text below, in which a university student emails a faculty member, requesting help, we see a number of possible instances of pragmatic failure (depending on the pragmatic norms of the relevant institution in which the email communication takes place). In the email opening, the faculty member's title ("Professor") is absent. Further, in the request head act, the student employs the use of a *want* statement as a content move, which may be perceived as insufficiently attending to the norms of politeness for a student making a request to faculty.

```
Dear Smith,
I want to come to your office today to ask for your help.
Kind regards,
Yuki
```

Email remains the most popular means of electronic communication in business and academia. Approximately 40% of the faculty in our university are non-Japanese [13] and so undergraduates may need to write in English to request permitted absences, delays in submission of assessments or additional help with homework or assignments [13].

1.3 Overview

An extensive search of the published literature revealed no existing dynamic language assessment platforms that were able to parse extended text for specific instances of pragmatic failure. Therefore, the aim of this research was to create a C-DA platform that could help Japanese learners of English draft pragmatically-appropriate request emails to non-Japanese faculty.

The C-DA is limited in scope by its remit to focus on (1) pragmatic failure in (2) request emails, written by (3) Japanese learners of English. The written English of learners may contain multiple issues, such as lexical and grammatical errors, which are not attended to in the current C-DA. Further, request-based emails are just one of the many possible types of emails that learners write. Finally, the pragmatic failures that the system is designed to identify are based on instances of failure types identified in a Japanese learner of English corpus [19], and so may not reflect the pragmatic failures that other users of English may produce.

The remainder of this chapter is organized as follows. Section 2 itemizes three intractable complications in the development of this tool. Section 3 explains how the incorporation of NLP into Computer-Assisted Language Learning (CALL) has created a new era of intelligent CALL. Related works are introduced in Section 4 including automated error detection tools and dynamic language assessment platforms. Section 5 provides the outline of the system architecture and initial design, and describes the practical implementation of the design. Section **??** reflects on the extent to which the three challenges in creating the system were overcome.

2 Challenges

There are three core challenges that need to be overcome when creating a CD-A that can provide graduated feedback on pragmatic failure: the detection of (1) the presence or (2) absence of pragmatic features in (3) learner texts, which are permeated with grammatical and lexical errors.

First, identifying the presence of pragmatic features resulting in pragmatic failure is more complex than, for example, identifying syntactic features. For instance, parts of speech in texts can be predicted relatively accurately by analyzing their context and syntactical relationships within a sentence and leveraging statistical probabilities. However, pragmatic features require understanding not only context, but also intent, and cultural nuances [19, 25], which are abstract and vary widely across different communication situations, unlike the more structured and rule-based nature of syntax.

Second, identifying pragmatic failure caused by the lack of a feature is more challenging because it involves both understanding subtle context-dependent communication cues and inferring what may be implicitly missing, requiring a sophisticated grasp of both the explicit content and the underlying expectations of a communicative act. Rule-based parsing can be used to match known strings, but cannot directly match the the intent of the string.

Third, although NLP has increased in power and sophistication over the years, algorithms that work on native-speaker texts are much less effective on learner texts, mainly due to their high perplexity, i.e. the difficulty of predicting the likelihood of subsequent words, given that the interlanguage used may vary in terms of lexical, syntactic and semantic choices. The variation is the use of language that is inappropriate, such as spelling mistakes and grammatical errors.

3 Intelligent Computer-Assisted Language Learning

The evolution of intelligent Computer-Assisted Language Learning (iCALL) marks a significant milestone in the integration of technology with language education. In the mid-20th century language labs in which tape recorders and later, computer technologies, were harnessed to provide language learners with monitored language practice. These early endeavors laid the groundwork for the emergence of Computer-Assisted Language Learning (CALL) at the turn of the century. The number of dedicated CALL language labs increased greatly in 1990s and early 2000s signifying a shift towards leveraging digital technologies in language pedagogy.

As CALL evolved, it gave rise to Mobile-Assisted Language Learning (MALL), fueled by the widespread adoption of smartphones and tablets. This transition underscored a shift towards making language learning accessible, allowing learners to engage with educational content anytime, anywhere on their own devices. Mobile and web applications played a pivotal role in this phase. Developers of language learning apps incorporated elements of gamification and personalized learning experiences to enhance the efficacy and appeal of language study.

NLP began with simple string manipulation techniques focused on pattern matching and keyword extraction. These early methods relied on hand-crafted rules. More sophisticated statistical models emerged in the late 20th century, which leveraged large text corpora. The introduction of machine learning algorithms further advanced NLP, enabling systems to automatically learn and improve from experience, and the creation of tools like the Natural Language ToolKit (NLTK) [3].

4 Related works

The earliest published works on error detection used string matching [15]. Many researchers have developed specialist tools for error detection for second language learners of English [14]. Some tools simply identify the errors, e.g. [8] while others provide suggested corrections for the errors. Automatic error detection in Japanese learner English has been a focus of research for at least two decades [4, 11, 12, 16]. The detection of errors and the detection of pragmatic failure are closely related concepts.

Pragmatic competence is an important aspect of overall communicative competence [6], and concerns the ability to adapt one's language choices appropriately to varying social contexts, with their accompanying expectations of appropriate levels of formality, politeness, and use of community conventions. Conversely, pragmatic failure refers to instances in which language use does not align with community pragmatic norms or expectations. This may lead to a learner being perceived negatively by an interlocutor, such as a university faculty member [9]. To date, instances of pragmatic failure in corpora have been identified manually; their often complex, subjective nature makes it difficult to implement automatic pragmatic annotation [19, 25]. Because of this, there have been few computerised DA focusing on pragmatics to date [22], with none that automatically identify specific instances of pragmatic failure in whole texts.

In a DA, a learner and an expert collaborate on a given task. When the learner encounters difficulty in proceeding with the task, the expert can provide support (mediation). From a sociocultural perspective, for this mediation to be effective in promoting learner development, it should fall within the ZPD of the learner. To this end, mediation should be graduated and contingent [2]. In terms of the former, initially only minimal, highly implicit assistance should be provided. If this does not help the learner, the expert gradually increases the explicitness of the assistance until the learner is able to successfully continue with the task. Mediation should also be contingent upon need, only being provided if necessary. In this way, the DA aims to promote development of the learner by attending to their ZPD. At the same time, the learner's ability can be assessed by evaluating the frequency of the mediation given, and its degree of explicitness.

Two broad categories of DA can be identified – interactionist and interventionist DA. In the former, mediation provided is unscripted, and is produced by the mediator responding to the particular needs of an individual learner at that moment in time [20]. Interventionist DA, on the other hand, employs standardised, scripted forms of feedback. While a degree of flexibility and sensitivity to a learner may be lost, interventionist DA allows for increased reliability and generalisation [21]. This latter, standardised approach is also suitable for computerised forms of DA [21].

While DA was initially developed within educational psychology, in recent years it has also been applied within the field of second language (L2) acquisition. The methodology has been applied to the assessment of L2 learner grammar [20], listening comprehension [1], French pronoun usage [23], and spoken requesting [17]. A number of studies have also begun to explore the potential for a computerised form of DA (C-DA), in which standardised graded and contingent feedback is provided automatically [21, 22]. However, with the exception of Qin and van Compernolle [22], who applied C-DA to learner implicature comprehension, there has been little research on C-DA in relation to learners' pragmatic competence, or to pragmatic competence in relation to L2 English emailing specifically.

5 System Architecture and Design

5.1 General Design Goals and Technological Stack

The design of the system and the choice of technologies were dictated by a number of basic business requirements, such as:

- **Portability.** The system has to be accessible with a variety of desktop and mobile devices.
- **Teacher-centered functionality.** The system has to provide a specialized interface for the teacher, allowing to modify exercises and to evaluate student progress.
- **Student-centered functionality.** Likewise, the system has to provide student-centered functions, such as completing assignments and tracking progress.
- **Persistency.** The system has to be able to save student progress between the sessions and let the students to resume work, possibly using another device.
- **Group management.** The system should provide an option to offer different challenges for different groups of students.

These requirements made the implementation of the system as a Web application a natural choice. The user accesses the C-DA, input texts and views machine-generated feedback via a browser, while the bulk of the work takes place on a backend server. We use the Django framework, which provides much of the required functionality out of the box. In addition, Django is based on Python, having excellent support for NLP libraries and tools. Fig. 1 shows the interactions within the model-template-view architecture and its connection to the database. The user-facing frontend is powered with Bootstrap JavaScript library, enabling us to build cross-platform responsive interface. The current C-DA functionality requires only relatively basic NLP processing capabilities, supplied by the NLTK package [3]. User access rights management and group management are performed with built-in Django functions.

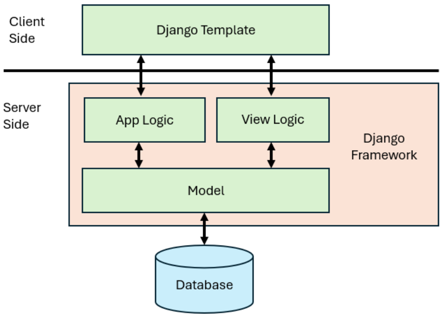


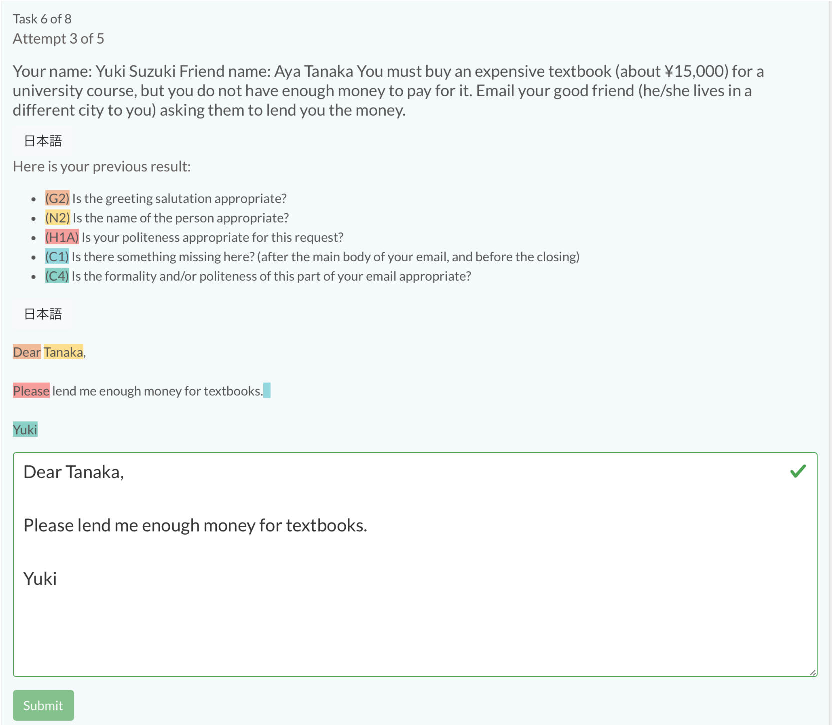
Fig. 1. System architecture

The system is deliberately designed for easy rollout and upgrade. In production mode, the web application is served with a combination of an application server *Waitress* and static file server *WhiteNoise*, and persistent storage is established by a built-in *SQLite* database. Alternatively, the system can be deployed with a container manager, simplifying upgrades and dependency management.

5.2 Software Architecture Overview

Django applications are typically designed by extending the initial pre-generated project template. It features a built-in database, and an administration page for creation and management users and user groups. User-facing functionality is typically accessible with different internet addresses (URLs), mapped to specific actions.

In our case, the administration page was modified to support the creation of student assignments. In addition to a free-form textual description, each assignment has to fall under one of four categories, reflecting the required speech register. If the currently authorized user is identified as a teacher, teacher-targeted interface elements are made visible. They allow browsing student submissions, exporting the summary of results in the Excel-compatible CSV format, and the printout of certain statistical data. The student-facing interface as shown in Fig. 2 allows students to resume making the assignments. Each submission attempt is saved in the system, so that the work can be continued from the previous save point. This functionality is also used to generate graduated feedback: subsequent submissions of the same (failed) task trigger more elaborate responses of the system.



Task 6 of 8
Attempt 3 of 5

Your name: Yuki Suzuki Friend name: Aya Tanaka You must buy an expensive textbook (about ¥15,000) for a university course, but you do not have enough money to pay for it. Email your good friend (he/she lives in a different city to you) asking them to lend you the money.

日本語

Here is your previous result:

- (G2) Is the greeting salutation appropriate?
- (N2) Is the name of the person appropriate?
- (H1A) Is your politeness appropriate for this request?
- (C1) Is there something missing here? (after the main body of your email, and before the closing)
- (C4) Is the formality and/or politeness of this part of your email appropriate?

日本語

Dear Tanaka,

Please lend me enough money for textbooks.

Yuki

Dear Tanaka,

Please lend me enough money for textbooks.

Yuki

✓

Fig. 2. Learner-facing user interface

5.3 Feedback Generation

The most advanced functionality of the system is student feedback generation. C-DA has to analyze the submitted attempt (the next revision of the required email message) and provide feedback on the basis of the target speech register and the current attempt number. The first attempt might yield a highly implicit feedback message like *"Is there anything that could be changed in this part of your email?"*, while the subsequent responses increase in explicitness (e.g. *"For this email, the greeting salutation 'Dear' is appropriate."*)

As shown in Table 1, fifteen categories of pragmatic failure were identified by expert annotators in Japanese learner corpus of request emails [19]. The current system recognizes 24 types of student pragmatic failure. For each failure type, we specify a list of failure types that must be reported instead if any of them are found. The rationale is to avoid feedback duplication: for example, we do not report the "inappropriate greeting" failure type (G2 failure type) if the greeting is missing (G1 failure type). As a result, the student only receives feedback for the G1 failure type.

Table 1. Categories of pragmatic failure

Category	**Tag Code**	**Tags**
Greeting	G1	Greeting absent
	G2	Greeting inappropriate
Opening	T1	Title absent
	T2	Title inappropriate
	N1	Name absent
	N2	Name inappropriate
Body	B1	Inappropriate use of spacing after opening
	B2	Lack of self-introduction (if appropriate)
	B3	Inappropriate lack of external modifiers
Head Request	H1	Overly direct head request act
	H2	Overly indirect head request act
Closing	C1	Pre-closing absent
	C2	Pre-closing inappropriate
	C3	Closing absent
	C4	Closing inappropriate

The complexity of identifying individual failure types varies significantly. As a general rule, we presume that one of the goals of the system is to train the use of standard set expressions. The target email messages tend to be highly formulaic, so we require the students to follow the suggested patterns closely. For this reason, we normally rely on simple pattern matching to reveal instances of pragmatic failure.

As an example of easy-to-identify failure types, consider again the types G1 ("greeting missing") and G2 ("greeting inappropriate"). To test for G1, we take the first sentence of the email, and check whether it contains the required greeting headword (such as "dear", "hello", etc.) To test for G2, we make sure that G1 is not triggered, and the greeting headword corresponds to the target politeness register.

The family of H failure types represents a more challenging case. These instances of pragmatic failure are identified when the student uses a request construction that is considered too direct for the target politeness register. For example, we require the use of "possibly" in a formal setting ("could you *possibly* help us"). To identify such failure instances, we first isolate the head act, which can be defined as the body of the message between the opening and the closing parts in our case of a short email. Next, we apply part of speech tagging, and search for the required combinations, such as “polite word” + verb. A separate set of handcrafted rules is applied to the head act to check for different H-type failure types.

In addition to the usual challenge of minimizing both false-positive and false-negative triggers, we have to ensure that pragmatic failure conditions do not overlap. The identified failure type is passed to the user feedback system that selects the final message on the basis of the current submission attempt.

5.4 Feature Testing

The above-described system of handcrafted cross-referenced rules is inherently brittle: a change in failure detection code might cause unwanted consequences in related instances of failure. To ensure reliability of C-DA and consistent failure detection, we have devised a list of feature tests that are run after every code change. Basic feature tests work as a sanity check in simple cases:

```
# Greeting absent
def test_G1(self):
    self.assertError(TC.B, 'G1', "John, here is my request.")
    self.assertNoError(TC.B, 'G1', "Dear John, here is my request.")

# Greeting inappropriate
def test_G2(self):
    self.assertError(TC.B, 'G2', "Hi, this is my test answer.")
    self.assertNoError(TC.R, 'G2', "Dear, this is my test answer.")
    self.assertError(TC.G, 'G2', "Dear, this is my test answer.")
    self.assertNoError(TC.Y, 'G2', "Hi, this is my test answer.")
```

For instance, this code checks that the text *“John, here is my request.”* should trigger the G1 failure type, while *“Dear John, here is my request.”* should not. Task categories, such as `TC.B`, `TC.R`, etc. correspond to different target politeness. In particular, `TC.B` is polite register, and thus the greeting ”Hi” is considered inappropriate for any `TC.B`-labeled submission. Advanced feature tests run failure detection code for the complete email messages and make sure the system identifies the correct set of failure types.

6 Conclusion

To conclude, we examine the extent to which our system address three key challenges identified in Section 2, namely the (1) detection of the presence of pragmatic features resulting in pragmatic failure; (2) detection of the absence of pragmatic features, resulting in pragmatic failure; and (3) parsing texts written by learners of English, which contain a plethora of language errors.

For the first challenge, the C-DA successfully matches pragmatic failure when the failure can be detected through simple string-matching algorithms. This was achieved by identifying the typical pragmatically inappropriate forms contained in the Japanese learner corpus. Thus, the program is limited to detecting prototypical pragmatic errors of Japanese learners. As the system is designed for use in a Japanese context, this limitation is not problematic. However, although the corpus is sufficiently large at approximately 1300 email texts, all potential errors do not occur within the corpus; thus, future users may make errors that will not be detected.

Detecting pragmatic failure caused by the lack of presence of a pragmatic feature is a simple task, when the feature is well-defined. For example, the lack of a standard salutation in a formal email can be discovered by failing to match a set of salutations. However, searching for external modifiers, such as grounders that prepare the reader for the forthcoming request was more problematic. Although the function of the modifiers can be defined, the functional exponents that can be harnessed to realize the function are innumerable. The current version searches for lack of a defined set of frequent exponents providing coverage for most instances, but failing to detect the acceptable yet less common forms. The system varies greatly in its ability to pinpoint the absence of such features, and so this remains a work in progress.

Dealing with L2 texts rather than native-speaker texts raised multiple issues. One such issue was the unexpected variety of forms that learners used. Based on analysis of the corpus, we could identify prototypical errors and incorporate their discovery into the system. However, idiosyncratic language usage typical of learner interlanguage that contains a combination of grammatical, vocabulary and spelling mistakes remains challenging to parse. As the database of user submissions increases, we aim to more finely-tune the algorithms by tailoring the system to match more errors.

References

1. Ableeva, R.: Dynamic assessment of listening comprehension in second language learning (2010)
2. Aljaafreh, A., Lantolf, J.P.: Negative feedback as regulation and second language learning in the zone of proximal development. The Modern Language Journal **78**(4), 465–483 (1994). https://doi.org/10.2307/328585
3. Bird, S.: Nltk: the natural language toolkit. In: Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions. pp. 69–72 (2006)
4. Blake, J.: Genre-specific error detection with multimodal feedback. RELC Journal **51**(1), 179–187 (2020). https://doi.org/10.1177/0033688219898282
5. Brown, P., Levinson, S.C.: Politeness: Some universals in language usage. Cambridge University Press, Cambridge (1987)
6. Celce-Murcia, M.: Rethinking the role of communicative competence in language teaching. Intercultural language use and language learning pp. 41–57 (2007)
7. Chen, Y.s.: Developing chinese efl learners' email literacy through requests to faculty. Journal of Pragmatics **75**, 131–149 (2015)

8. De Felice, R.: Automatic error detection in non-native English. Ph.D. thesis, Oxford University, Oxford (2008)
9. Economidou-Kogetsidis, M.: Variation in evaluations of the (im) politeness of emails from l2 learners and perceptions of the personality of their senders. Journal of Pragmatics **106**, 1–19 (2016)
10. Ishihara, N.: Teacher-based assessment for foreign language pragmatics. TESOL Quarterly **43**(3), 445–470 (2009)
11. Izumi, E., Uchimoto, K., Isahara, H.: The overview of the SST speech corpus of Japanese learner English and evaluation through the experiment on automatic detection of learners' errors. International Conference on Language Resources and Evaluation (2004)
12. Izumi, E., Uchimoto, K., Saiga, T., Supnithi, T., Isahara, H.: Automatic error detection in the Japanese learners' English spoken data. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics (2003)
13. Kaneko, E., Park, M., Wilson, I., Heo, Y., Roy, D., Yasuta, T., Nicholas, A., Blake, J.: English curriculum innovation for computer science majors in the japanese efl context: From needs to tasks. In: 2018 IEEE International Professional Communication Conference (ProComm). pp. 84–89. IEEE (2018)
14. Leacock, C., Gamon, M., Mejia, J.A., Chodorow, M.: Automated grammatical error detection for language learners. Springer Nature (2022)
15. Macdonald, N., Frase, L., Gingrich, P., Keenan, S.: The writer's workbench: Computer aids for text analysis. IEEE Transactions on Communications **30**(1), 105–110 (1982). https://doi.org/10.1109/TCOM.1982.1095380
16. Nagata, R., Iguchi, T., Wakidera, K., Masui, F., Kawai, A.: Recognizing article errors in the writing of Japanese learners of English. Systems and Computers in Japan **36**(7), 54–63 (2005)
17. Nicholas, A.: Dynamic assessment and requesting: Assessing the development of Japanese EFL learners' oral requesting performance interactively. Intercultural Pragmatics **17**(5), 545–575 (2020)
18. Nicholas, A., Blake, J., Mozgovoy, M.: Efl learners and english email writing: developing a computerised diagnostic language assessment. CALL and professionalisation: short papers from EUROCALL 2021 p. 238 (2021). https://doi.org/10.14705/rpnet.2021.54.1339
19. Nicholas, A., Blake, J., Mozgovoy, M., Perkins, J.: Investigating pragmatic failure in l2 english email writing among japanese university efl learners: A learner corpus approach. Register Studies **5**(1), 23–51 (2023)
20. Poehner, M.E.: Dynamic assessment: A Vygotskian approach to understanding and promoting L2 development, vol. 9. Springer Science & Business Media (2008)
21. Poehner, M.E., Zhang, J., Lu, X.: Computerized dynamic assessment (C-DA): Diagnosing l2 development according to learner responsiveness to mediation. Language Testing **32**(3), 337–357 (2015)
22. Qin, T., van Compernolle, R.A.: Computerized dynamic assessment of implicature comprehension in l2 chinese (2021)
23. Van Compernolle, R.A.: Sociocultural theory and L2 instructional pragmatics, vol. 74. Multilingual Matters (2014)
24. Vygotsky, L.S., Cole, M.: Mind in society: Development of higher psychological processes. Harvard University Press (1978)
25. Weisser, M.: Speech act annotation. Corpus pragmatics: A handbook pp. 84–113 (2014)