TIARA 2.0: an interactive tool for annotating discourse structure and text improvement

Jan Wira Gotama Putra¹ · Kana Matsumura² ·
Simone Teufel^{3,4} · Takenobu Tokunaga¹

Accepted: 20 October 2021 / Published online: 24 November 2021
© The Author(s) 2021

Abstract Discourse structure annotation aims at analysing how discourse units (e.g. sentences or clauses) relate to each other and what roles they play in the overall discourse. Several annotation tools for discourse structure have been developed. However, they often only support specific annotation schemes, making their usage limited to new schemes. This article presents TIARA 2.0, an annotation tool for discourse structure and text improvement. Departing from our specific needs, we extend an existing tool to accommodate four levels of annotation: discourse structure, argumentative structure, sentence rearrangement and content alteration. The latter two are particularly unique compared to existing tools. TIARA is implemented on standard web technologies and can be easily customised. It deals

This work was partially supported by Tokyo Tech World Research Hub Initiative (WRHI), JSPS KAKENHI Grant Number 20J13239, and Support Centre for Advanced Telecommunication Technology Research. The software accompanying this article is publicly available at <https://sites.google.com/view/tokyotechcl-tiara/>.

✉ Jan Wira Gotama Putra
gotama.w.aa@m.titech.ac.jp

Kana Matsumura
kmatsumura@suou.waseda.jp

Simone Teufel
simone.teufel@cl.cam.ac.uk

Takenobu Tokunaga
take@c.titech.ac.jp

¹ School of Computing, Tokyo Institute of Technology, Tokyo, Japan

² Graduate School of Education, Waseda University, Tokyo, Japan

³ Department of Computer Science and Technology, University of Cambridge, Cambridge, UK

⁴ Tokyo Tech World Research Hub Initiative (WHRI), Tokyo Institute of Technology, Tokyo, Japan

with the visual complexity during the annotation process by systematically simplifying the layout and by offering interactive visualisation, including clutter-reducing features and dual-view display. TIARA's text-view allows annotators to focus on the analysis of logical sequencing between sentences. The tree-view allows them to review their analysis in terms of the overall discourse structure. Apart from being an annotation tool, it is also designed to be useful for educational purposes in the teaching of argumentation; this gives it an edge over other existing tools.

Keywords Annotation tool · Discourse structure · Argumentative structure · Teaching of argumentation

1 Introduction

There are many important aspects to writing, such as grammar, mechanics, rhetorical move, argumentative structure and coherence (Bacha, 2010; Grosz & Sidner, 1986; Lee and Webster, 2012). Motivated by the need for individualised feedback in learning, many studies in the computational linguistics (CL) community have developed various systems to detect and correct micro-level errors targeting factual knowledge, such as grammar and spelling (e.g., Han et al., 2006; Hirst & Budanitsky, 2005; Yuan & Briscoe, 2016). Despite the growing interest in the field, systems that support the analysis of macro-level aspects, such as rhetorical move and argumentative structure, are still infrequent (Crossley et al., 2016; Strobl et al., 2019).

Our long-term goal is to perform an automatic analysis of argumentative structure in English-as-a-foreign-language (EFL) learner essays, as well as providing suggestions on how to improve the essays by rearrangement of sentences (cf. Sect. 2). Argumentative structure shows how sentences relate to each other and what roles they play in the argumentation. On the other hand, rearrangement of sentences may improve the organisational quality of the essays (Matsumura & Sakamoto, 2021; Silva 1993; Zhang et al., 2017). However, before we can provide such a system, we start by studying two types of annotations: argumentative structure, and parallel original and reordered versions of EFL texts; a tool is needed to support such annotations. Some tools support the annotation of argumentative structure (e.g., GraPat (Sonntag & Stede, 2014) and DiGAT (Kirschner et al., 2015)) but none of the available tools support the sentence rearrangement annotation. Since most studies aimed at only analysing texts, it is natural that they did not allow text editing operation in their tools. On the other hand, there are also studies aiming at annotating text revisions, e.g., Zhang et al. (2017). However, Zhang et al., did not perform the argumentative structure analysis.

A new annotation study often has specific, so far unserved needs, and we are no exception. Our long-term goal is unique compared to past studies in that we perform both argumentative structure and text revision (by reordering) annotations. Although tools that support a wide range of annotation tasks (e.g., INCEpTION (Klie et al., 2018)) exist, the modification of an existing tool is still often not

realistic due to many real-life constraints, for example, the time involved in modification rather than fresh implementation, the availability of documentation and the entire redesign necessary when novel annotation needs diverge too much. This renders the use of existing tools limited before the ever-changing annotation needs. We previously developed an annotation tool TIARA¹ (Putra et al., 2020) to cater our specific needs: argumentative structure and sentence rearrangement annotations. In this article, we take a further step forward by extending the tool to be more “generic”, that is, supporting a wider range of annotation tasks. On top of this, we also design the extended tool to be also useful for educational purposes, particularly in the learning-to-write scenario, which constitutes the uniqueness of our extended version compared to other tools.

Our extended annotation tool TIARA 2.0 (henceforth simply referred to as “TIARA” except in ambiguous cases) can be used for four different levels of annotation as follows.

- *Discourse structure* annotation, which identifies how discourse units (e.g., sentences, clauses) function in the text and how they connect amongst each other, forming a hierarchical structure.
- *Argumentative structure* annotation, which in contrast to the general discourse annotation, uses discourse units selectively. Here, discourse units are categorised as argumentative and non-argumentative components. Non-argumentative components are not connected to the structure.
- *Sentence re-arrangement* annotation, which is one of the common feedback in education, aiming to improve text coherence and organisational qualities. Different to the previous two levels of annotation which analyse the texts as they are, sentence rearrangement modifies the textual surface. However, it does not modify the textual content.
- *Content alteration* annotation. In the learning-to-write scenario, students are often asked to modify their content in response to instructors’ feedback. Students can perform this directly in TIARA by adding, deleting, and editing discourse units.

The rest of this article is structured as follows. We outline our annotation needs (target domain, annotation scheme) in Sect. 2, and describe how these requirements translate to design considerations and features of TIARA in Sect. 3. Section 4 shows how it sits among other annotation tools. This article also describes how the argumentative structure analysis will be useful in the education domain in Sect. 5. Furthermore, it describes how TIARA can also be used to facilitate the teaching of argumentative writing as a byproduct of our design. Finally, Sect. 6 concludes this paper and describes what can be improved in the future.

¹ TIARA stands for **T**ool for **I**nteractive **A**RGument **A**nnotation. The tool and a detailed user manual are publicly available at <https://sites.google.com/view/tokyotechcl-tiara/>.

2 Annotation needs

2.1 Target domain

Our target texts are argumentative essays, which are common in classroom writing exercises. We source our texts from the ICNALE² corpus, a collection of short argumentative essays written in English by Asian college students, of 200–300 words in length (Ishikawa, 2013, 2018). Most of these students have intermediate level proficiency.

Texts written by both native and non-native speakers may contain micro-level errors, such as in terms of grammar (although the errors are less frequent in native writings); but non-native writings are particularly unique in terms of macro-level errors (Rabinovich et al., 2016; Silva, 1993). The field of *contrastive rhetoric* studies how non-native speakers, who are influenced by their mother tongues and cultures, may structure their texts differently from those of native speakers (Bacha, 2010; Connor, 2002; Johns, 1986; Kaplan, 1966; Silva, 1993). For example, in the argumentative essays written by East Asian students, it is observed that reasons for an opinion may be presented before the opinion, which is not common in Anglo-Saxon cultures (Connor, 2002; Johns, 1986; Kaplan, 1966; Silva, 1993). This makes the texts written by EFL learners may be perceived as less coherent in the eyes of native readers because the order of sentences in non-native writings violates the expectations of the native readers. One way to mitigate this problem is by reordering sentences to satisfy the argument-development strategies that are perceived coherent in the eyes of native speakers (Bamberg, 1983; Connor, 2002; Garing, 2014; Kaplan, 1966; Silva, 1993; Zhang et al., 2017). Learner essays in general may also pose content-related issues. For example, an essay is imbalanced when arguing from only a single viewpoint (Hsin & Snow, 2020; Matsumura & Sakamoto, 2021). In this case, instructors may recommend to add more sentences for a more balanced argument. In short, EFL essays pose more challenges compared to the texts normally used for discourse annotation in past studies, where the texts were written by proficient writers and all parts of the texts could be assumed to be coherently connected (Mann & Thompson, 1988).

2.2 Discourse structure annotation scheme

Discourse annotation aims to create a structured representation out of text, which explains how discourse units (e.g. sentences or clauses) relate to each other and what roles they play in the overall discourse (Mann & Thompson, 1988; Wolf & Gibson, 2005). The discourse structure can be represented as a tree (Mann & Thompson, 1988) or a graph (Wolf & Gibson, 2005). The discourse units are represented as nodes, and discourse relations as edges (links) between nodes or between nodes and edges (for example, when challenging the acceptability of the inference between nodes—Peldszus and Stede (2016)). Therefore, there are two main features of a discourse annotation tool: (1) annotating the categories (roles) of

² <http://language.sakura.ne.jp/icnale/>.

discourse units and (2) annotating the links and label the relations between discourse units.

Since we are trying to analyse argumentative essays written by EFL students, we approach the discourse structure analysis from the argumentation perspective. The annotation of argumentative structure typically consists of two main steps (Lippi & Torroni, 2016). The first is *argumentative component identification*, which differentiates argumentative components (ACs) and non-argumentative components (non-ACs) (Lippi & Torroni, 2016). This is because discourse units that do not function argumentatively may exist even in an argumentative text. The main difference between general discourse structure annotation and argumentative structure annotation lies in this treatment of non-ACs. ACs can be further classified into a more fine-grained categories; for example, the distinction between *proponent* and *opponent* (Sonntag & Stede, 2014), or the distinction between *major claim*,³ *claim*, and *premise* (Stab & Gurevych, 2017). The second step is *argumentative discourse structure prediction*, which establishes relations between ACs to form the argumentative structure representation. All ACs must be connected to the structure as nodes, while all non-ACs remain unconnected. Relations (edges) can be directed (Stab & Gurevych, 2017) or undirected (Kirschner et al., 2015).

In our annotation scheme, we annotate EFL essays at the sentence granularity level and represent the argumentative structure as a tree. We differentiate ACs and non-ACs. However, our approach differs from the previous studies on student essays by Stab and Gurevych (2017) that a further classification of ACs into *main claim*, *claim*, and *premise* is not necessary or appropriate for our analysis. In our scheme, as in that by Stab and Gurevych (2017), the major claim is already topologically distinguished as the root of the tree structure, which is recognisable as the only node with incoming but no outgoing links. We do not label ACs as *claim* and *premise* to avoid conflicts that might arise in long argumentation chains. A premise at level X can easily itself become the claim for a lower-level premise at level $X + 1$, making the AC act as both claim and premise at the same time. With a finite number of labels, this means that none of the fixed labels is applicable. We note that such ambiguous cases do happen in Stab and Gurevych's study; these cases were resolved according to topology, a treatment that is consistent with our decision not to label ACs in the first place. We feel that omitting AC labels makes our annotation scheme not only more economical but also intrinsically consistent. That being said, our annotation tool TIARA 2.0 still facilitates the annotation of such rhetorical categories to support a wide range of annotation schemes.

The second step of our annotation scheme is to establish the relationship between ACs. We observed that for argumentative essays, the tree structure is the most common and natural representation of relations, as a single higher-level statement is recursively supported or attacked by one or more lower-level statements (Carlile et al., 2018). We use both directed and undirected links in our scheme. However, in computation, we represent undirected links as directed, provided which nodes act as *source (child)* and *target (parent)*. It means, our definition of "undirected" link is

³ The concept behind a *major claim* has also variously been referred to as *central claim* and *main stance* in the literature.

just a matter of visualisation (the presence of arrow in the visualisation) and the interpretation of the link label in question, but not of computation. This strategy is adopted to eliminate circular links which is not allowed in our scheme. Readers may refer to Putra et al. (2021) for a more complete explanation of our argumentative structure annotation scheme.

2.3 Sentence rearrangement and text editing

One of our long-term goals is helping EFL learners to improve their essays to closely resemble native-level productions, particularly considering the organisation of ideas that is expected by native readers, an aspect which they often struggle with (Bacha 2010; Johns, 1986; Silva, 1993; Zhang et al., 2017). One way to do this is by *re-arrangement* of sentences. This is the second layer of our annotation scheme, on top of the argumentative discourse structure annotation. If the sentences in the essays are not already in the best order they could be, we ask annotators to arrange them into a more logically well-structured text. However, reordering may introduce errors in referring and connective expressions (Iida & Tokunaga, 2014). To remedy these negative changes, annotators are allowed to correct such kinds of expressions to retain the original semantics of the sentence. An example of this is to replace a pronoun with its referent noun phrase or to make an implicit connective explicit by the use of conjunctions, for instance, “because.”

Our scheme with two annotation layers of argumentative structure and sentence reordering is very beneficial for EFL students. These annotations can enable students to understand the differences between their writings and native (“good”) writings—*contrastive analysis* (Bacha, 2010; Kaplan, 1966; Silva, 1993). By understanding the differences, EFL students will understand the necessary material and the coherent arrangement of ideas that are expected by native readers (Bacha, 2010). While the content modification, for example, adding and deleting sentences, is also often required to improve texts, our current annotation scheme does not cover such content modification yet. Still, TIARA 2.0 facilitates content modification to support the educational and future use cases.

3 Annotation tool TIARA 2.0

3.1 Design

There are several considerations that influence TIARA’s technical and visual design.

- (a) *Intuitive interface and visualisation*—We believe an annotation tool should provide an intuitive interface and visualisation. In the context of this study, it means the annotators must be able to read sentences in linear order while viewing the (argumentative) discourse structure as well. This is to support both logical-sequencing and structural analysis. The novelty of TIARA lies in this dual-view (text view and tree view), which we believe provides an

important aspect of global overview to the annotators who operate by making local changes.

- (b) *Ease of use, installation and deployment*—Ease of use and installation for annotators is often prioritised for annotation design, but we believe that deployment is equally important. Not every project owner is tech-savvy; for them, an annotation tool that is hard to deploy is practically unusable. In contrast, tools that are usable without deployment and may run at the client-side, such as EasyTree⁴ (Little & Tratz, 2016), are able to reach and help many potential users, including those who have no knowledge in the inner-work of computer systems. TIARA shares the same principle. Users only need a web browser⁵ and the TIARA package. This tool is written in JavaScript, HTML and CSS. We use JsPlumb⁶ and Treant-js⁷ as the visualisation libraries. We understand that the necessity of deployment (server-side) is often coupled with annotation management features (Yimam et al., 2013), and this is important in a large annotation project (Kaplan et al., 2010). Although the current version of TIARA does not actively support such annotation management yet, we plan to do so in future TIARA versions.
- (c) *Annotation scheme compliance and completeness checking*—An annotation tool ideally prevents violations of its annotation scheme, such as an illogical annotation of connecting a sentence to itself. Compliance guarantee offered by annotation tools are attractive; annotators can follow their natural workflow without having to worry about doing something wrong or having to perform separate checks. Project owners also benefit from this as they do not have to ask the annotators for a post-hoc repair of the annotations. TIARA checks in real-time whether the annotation violates any constraint of the annotation scheme, and warns the annotator when it does. There are three constraints we implement in TIARA. First, we do not allow self-loop and circular links. Second, users cannot establish relations from and to non-AC nodes. Third, the annotated structure should form a hierarchical structure. On top of compliance to the scheme, TIARA also checks whether the annotation is *complete* upon saving (incomplete annotation cannot be saved). This is to ensure that the annotators finish their assignments, particularly considering that we do not implement an annotation management feature. TIARA checks whether all discourse units have been categorised and whether all discourse units have been connected to the whole structure (except for non-ACs). This feature can be turned on and off according to project owners' preference (cf. Sect. 3.4).
- (d) *Annotation tracking*—Tracking changes and actions performed by the annotators is important because it provides information about annotation behaviour. It is also valuable for troubleshooting annotation schemes because project owners can identify the parts that often cause confusion or require

⁴ <https://github.com/alexalittle/easytree>.

⁵ Google Chrome and Safari are recommended

⁶ <https://jsplumbtoolkit.com>.

⁷ <https://fperucic.github.io/treant-js/>.

post-hoc repair. For example, we know that labels X and Y are potentially confusing when annotators often change the links labelled with X to Y (and vice versa). TIARA records the annotator actions in each annotation file.

- (e) *Customisability*—An annotation tool must be flexible in order to accommodate a wide variety of annotation tasks (Kaplan et al., 2010). This is important in the early stage of an annotation study when the project goal and annotation scheme might frequently change. We adhere to the principle that users should never have to touch the main code at all; they should be able to customise the annotation tool easily in some other way. Similar to BRAT annotation tool (Stenetorp et al., 2012), the annotation scheme of TIARA can be changed by editing a configuration file. The project owners should define this configuration script at the start of an annotation project, and keep it unchanged throughout the project. We chose this approach over the alternative, a user interface provided by the tool, for instance, as in RSTTool (O'Donnell, 2000), since JavaScript should not modify local files on-the-fly for security reason.

3.2 Annotation process and functionalities

To annotate using the tool, users have to prepare a `.txt` file where each discourse unit has been separated by a newline.⁸ Users then load this file into TIARA for annotation. The annotated text can then be exported to TIARA's internal saving format (`.html` file) or spreadsheet-friendly format (`.tsv`). Both export file types can be loaded into the tool to modify an existing annotation.

To illustrate the tool's interface and how our annotation is operated using the tool, we show a fictional argumentative essay with the prompt “*smoking should be completely banned at all the restaurants in the country*” in Fig. 1, in which each sentence (discourse unit) is separated by a newline.

The example essay can be divided into several parts. S1–S3 together form the introduction section of the essay. S1 provides a background for the discussion topic, and S3 serves as the major claim of the essay. S2, which describes a personal episode that does not have an argumentative function, is identified as a non-AC, and thus excluded from the argumentative structure. S4–S5 discuss the topic of enjoyment of eating and talking, with S5 providing the introduction of this idea, and S4 giving an opinion on the topic. Sentence S6 then presents an argument about the number of customers; it supports the opposite opinion of S3. S7 introduces a new health-related argument as a counter-argument to the opinion in S6. Finally, S8 concludes the whole argument, by restating the major claim.

Figure 2 illustrates TIARA **text view** in which the annotation is performed on the example essay. Annotators can read the discourse units sequentially while viewing the annotated discourse structure at the same time. The interface in the text view is divided into two parts, the menu navigation part at the top and the work area at the bottom. After loading a text file, its content is shown in the work area. Each

⁸ Note that users may also define discourse units at the clause level. The bottom line here is to prepare a file where each discourse unit has already been separated by newlines.

s1 Government has been trying to introduce laws to ban smoking in restaurants.
 s2 I have watched the news.
 s3 I agree with the prompt.
 s4 If somebody smokes in the restaurant, other people may not be able to enjoy their meal.
 s5 In restaurants, customers enjoy eating and talking.
 s6 However, if we ban smoking in restaurants, they might lose some customers.
 s7 But I firmly support banning smoking in restaurants since we need to prioritise health.
 s8 In conclusion, I encourage banning smoking at all restaurants.

Fig. 1 An example essay

Fig. 2 A screenshot illustrating TIARA 2.0 text view

discourse unit appears framed in a box (denoting node), numbered (“ID”) according to its original order in the input text. Coloured links (defined by user) depict the annotated relations and their labels. The upper right-hand side of the work area shows the relation label legend.

TIARA supports the differentiation between ACs and non-ACs by checking the “drop” checkbox located at the right-hand side of each sentence box. When checked, the corresponding box is blacked-out and annotators cannot establish a relation to or from the dropped unit. For example, S2 is dropped in Fig. 2. Annotators may un-check to revert back. This feature can be used to simulate deleting sentences as well in the educational use case. Annotators link discourse units by dragging an arrow from the rectangular endpoint of the source unit to the circular endpoint of the target unit (left-hand side of the boxes). Annotators can then choose the link labels. Figure 2 shows three directed links, att (*attack*), det (*detail*)

Fig. 3 A screenshot illustrating TIARA 2.0 tree view for the annotation in Fig. 2

and sup (*support*), and an undirected link “=” (*restatement*). Annotators may delete or change the link label by clicking the established link in question.

TIARA supports indentation of discourse units by clicking the indentation button at the right-hand side of boxes (under the “drop” checkbox) to quickly visualise the hierarchical structure of the discourse (De Kuthy et al., 2018) and reduce cluttering. However, the indentation does not alter the discourse structure interpretation. This feature is only for readability purpose.

For the sentence rearrangement annotation, annotators may move the position of discourse unit boxes by drag and drop operations. For example, S4 and S5 are swapped in position in Fig. 2, introducing an idea before the main opinion. TIARA also allows annotators to edit the text inside boxes to correct connective and referring expressions due to rearrangement (cf. Sect. 2.3). To track changes, some notation can be employed. For example, using “[*original expression* | *revised expression*]” notation as illustrated in box (3). This feature can also be used to mark grammatical-error correction in the educational use case. TIARA can adjust the size of boxes by the “resize” button at the bottom of the text view. Should discourse units become longer after the editing operation, annotators may click the “resize” button to make the boxes bigger.

TIARA allows user to add new sentences into the text. This feature is specifically designed to support the potential use case in learning-to-write (mode detailed description in Sect. 5). For example, students do not always provide enough reasons to support their claims. In this case, instructors may recommend adding new reasons or elaborating existing content (Cho & MacArthur, 2010; Crossley & McNamara, 2016); the ‘add new sentence’ button serves this purpose. This feature can also be useful where students are asked to add more counter-arguments to produce a more balanced or comprehensive argument, considering multiple viewpoints (Hsin & Snow, 2020; Matsumura & Sakamoto, 2021).

Fig. 4 A screenshot illustrating TIARA 2.0 text view with discourse unit categorisation functionality

Fig. 5 A screenshot illustrating TIARA 2.0 tree view for the annotation in Fig. 4

Table 1 The association between annotation functionalities in TIARA 2.0 and various annotation levels

Functionalities\Level	Discourse	Argumentative	Rearrangement	Content
(a) Dropping discourse units		✓		✓
(b) Discourse unit categorisation	✓	✓		
(c) Linking and link labelling	✓	✓		
(d) Reordering			✓	
(e) Text editing			✓	✓
(f) Adding discourse units				✓

While the text view can be used to illustrate the local hierarchical structure of the discourse by using indentation, we think that it is not enough for the analysis of the whole discourse structure. Another view offered by TIARA is a **tree view**, which illustrates the shape of the structure as a whole. Figure 3 shows the tree view of the annotation in Fig. 2. The tree view emphasises the analysis of the overall discourse structure, while the text view emphasises the text analysis on logical sequencing and local connections. Annotators annotate in the text view and then verify their annotation in the tree view, and they can freely switch between both views while annotating. We believe that providing the tree view enhances the annotation experience, and therefore, the annotation quality. Some visual operations can be performed in the tree view to reduce cluttering. Annotators may also fold/unfold subtrees in the tree view by clicking the top-right button of unit boxes. This feature is useful for analysing longer texts as it prevents annotators from being overwhelmed by too much content at once. It is also possible to adjust the box and text sizes by clicking “shrink” and “enlarge” buttons; this feature is dedicated for the readability purpose. Annotators may capture and download the tree view visualisation (analogous to screenshot). The captured image can be printed and shared among annotators to facilitate discussion. In the educational use case, instructors may write comments on the (printed or digital) image to provide feedback to students.

We have shown an annotated-essay example using our annotation scheme without AC categorisation. However, as we have previously mentioned, TIARA also facilitate AC categorisation (discourse unit categorisation in general). This functionality can be turned on and off depending on the project needs. Figures 4 and 5 illustrate an annotation with AC categorisation as *proponent* and *opponent*. The difference between Figs. 2 and 4 lies in the dropdown option under each box, which is used to select the corresponding AC category. Figures 3 and 5 differ in terms of the box colouring; the box colouring scheme in the tree view is the same as the dropdown colouring scheme in the text view.

```

var enableDropping = true; // flag for “dropping discourse units” function
var enableSentenceCategorisation = true; // flag for “discourse unit categorisation”
    function
var enableLinking = true; // flag for “linking” function
var enableReordering = true; // flag for “reordering” function
var enableEditing = true; // flag for “text editing” function
var enableAddNewSentence = true; // flag for “adding discourse units” function
var enableIntermediarySave = false; // flag for “completeness checking” function;
    set true if you allow annotators to save incomplete annotation;
    false if only allowing complete annotation

var sentenceCategories = [“proponent”, “opponent”]; // discourse unit categories
var sentenceCatColours = [“lightseagreen”, “violet”]; // the visualisation colour for the
    corresponding unit categories

var relLabels = [“att”, “sup”, “det”, “=”]; // relation labels
var relColours = [“lightpink”, “lightgreen”, “lightblue”, “lightgray”];
    // visualisation colours for the corresponding relation labels
var relDirections = [true, true, true, false]; // relation types, true if directed (arrow presents
    in the visualisation) and false if undirected (no arrow)

```

Fig. 6 Example of TIARA 2.0 configuration script (written in JavaScript)

3.3 Functionalities and annotation levels

Table 1 summarises the association between the tool’s functionalities and various levels of annotation we have introduced in Sect. 1: discourse structure, argumentative structure, sentence rearrangement followed by text editing and content alteration annotations. All these functions will be useful for the educational use case as it contains all levels of annotation.

The preceding version of TIARA 1.0 described in our previous paper (Putra et al., 2020) did not support discourse unit categorisation and did not allow adding new discourse unit. It only supported argumentative structure and sentence rearrangement annotation levels which were specific to our scheme (cf. Sect. 2.2). In contrast, TIARA 2.0 described in this article supports a wider range of annotation levels as above. Concerning the visual operations, TIARA 2.0 additionally provide the box-resize functionality in the text view and the shrink/enlarge functionality in the tree view. While seemingly “small”, these additional visual features are helpful to reduce clutter in the display, and therefore enhance the annotation experience.

3.4 Customisation

Project owners may customise the sentence categories, relation types, relation labels and their colours by modifying an external configuration script. They can also disable or enable each annotation function provided. For example, a project owner may want to disable the dropping, reordering, text editing and sentence addition functions during a discourse structure annotation project. However, they should enable the dropping function for an argumentative structure annotation project. Figure 6 shows a configuration script example. During the preliminary trial of the

tool, we found that users can modify the configuration script as fast as 5 mins on their first try.

4 Related work

We have described the functionalities of TIARA 2.0 in the previous section. Now, we compare how the tool sits among other existing tools.

In the CL community, many annotation tools have been developed. Among them, BRAT⁹ (Stenetorp et al., 2012) is relatively popular as it supports a wide range of tasks. It offers annotation visualisation and collaboration features. BRAT also has been used for argumentative discourse structure annotation in the study by Stab and Gurevych (2017). Built in the same spirit as BRAT, WebAnno¹⁰ (Yimam et al., 2013) offers additional management and monitoring features. These tools are easy to customise, offering the flexibility to accommodate a wide range of annotation tasks. WebAnno also provided an automation mode in which the system can learn from annotations made by the user and provide suggestions. However, the automation has to be retrained and has to be triggered manually by an administrator. INCEpTION¹¹ (Klie et al., 2018) also provided such an automation called the “active learning” mode where in contrast to WebAnno, it does not have to be triggered manually by an administrator. However, BRAT, WebAnno and INCEpTION were originally designed for morphological, syntactic and semantic annotations, that is, rather local word or phrase-level annotation. While they support link display and could thus theoretically be used for discourse annotation, the visual display of links appears as drawn directly on top of text. This style of display has already been identified by others as a source of confusion for argumentation and discourse annotation projects (Kirschner et al., 2015). PDTB annotator¹² (Prasad et al., 2008) also falls into the class of annotation tools designed for local relations. When it comes to the display of larger-scale hierarchical or graphical structure of discourse, this falls entirely outside the purview of these tools.

Annotation tools which are specifically aimed at visualising larger-scale and more global discourse structure have also been developed, for example, rstWeb¹³ (Zeldes, 2016), TreeAnno¹⁴ (De Kuthy et al., 2018), OVA¹⁵ (Janier et al., 2014), DiGAT¹⁶ (Kirschner et al., 2015), and GraPat¹⁷ (Sonntag & Stede, 2014). Table 2 shows in detail how TIARA 2.0 is situated in terms of its annotation features amongst other annotation tools, in particular with respect to its support of

⁹ <https://brat.nlplab.org/>.

¹⁰ <https://webanno.github.io/webanno/>.

¹¹ <https://inception-project.github.io/>.

¹² <https://www.cis.upenn.edu/~pdtb/pdtb3-tools.shtml>.

¹³ <https://corpling.uis.georgetown.edu/rstweb/info/>.

¹⁴ <https://github.com/nilsreiter/treeanno>.

¹⁵ <http://www.arg-tech.org/index.php/ova/>.

¹⁶ <https://github.com/UKPLab/argmin2015-DiGAT>.

¹⁷ <https://github.com/discourse-lab/GraPat>.

Table 2 Comparison of features in TIARA 2.0 and discourse annotation tools in terms of argument mining tasks (1–7) and our additional needs (9–11)

Feature	GraPat	DiGat	OVA	TreeAnno	rstWeb	TIARA 2.0
1. Structure	Graph	Graph	Graph	Tree	Tree	Tree
2. Segmentation			✓		✓	
3. AC vs. non-AC categorisation	✓	✓	✓			✓
4. Discourse unit categorisation	✓					✓
5. Linking	✓	✓	✓	✓	✓	✓
6. Link labelling/polarity	✓	✓	✓		✓	✓
7. Structure visualisation	✓	✓	✓	✓	✓	✓
8. Scheme customisation			✓		✓	✓
9. Discourse unit reordering						✓
10. Text editing			✓			✓
11. Adding new sentence			✓			✓

Fig. 7 A screenshot of GraPAT annotation tool (adapted from Fig. 2 in Sonntag and Stede (2014))

argumentative structure tasks (1–7) and our additional needs (8–11), which of course it is designed to fulfil. Despite supporting a wide range of tasks, the reordering annotation is the problem for other tools (cf. Sect. 2.3). None of the existing tools support such annotation, which is indispensable in our project. Most annotation tools also do not allow changing the textual content, something that is needed for the educational purpose.

Fig. 8 A screenshot of DiGAT annotation tool

Fig. 9 A screenshot of OVA annotation tool

RstWeb is a strong competitor of TIARA in terms of features implemented and visual elegance. However, it only allows RST-style annotation in which all units have to be connected to the structure, while argumentative structure annotation excludes non-AC units in the global structure. Similar to TIARA, TreeAnno allows a general tree structure of discourse. TreeAnno is easy to use, but falls short in the number of features implemented. While the visualisation of hierarchy via node indentation in TreeAnno illustrates the discourse structure to some extent, it does not show the links between discourse units.

GraPat, DiGAT and OVA offer features that support discourse annotation tasks, and they assume a graph structure of texts. However, GraPAT and DiGAT require

considerable effort to customise the annotation scheme. While any tree structure is by definition also a graph, these tools cannot ensure annotation compliance to the specific tree structure we assumed in our scheme (cf. Sect. 2).

GraPat is the only tool among the surveyed tools that supports AC categorisation, into *proponent* and *opponent*; the distinction is represented by different shapes of nodes (circular for *proponent* and rectangular for *opponent*). Thus, it is at an advantage over other tools for argumentative discourse annotation. It also allows establishing relations between nodes and edges, e.g., in the case of *undercut*. The undercut relation is used to challenge the acceptability of inference between two nodes (Peldszus & Stede, 2016). For example, the textual span ‘*except for, they would be indestructible*’ in Fig 7 suggests that it is rather hard to forbid energy-saving lamps since they would be indestructible. GraPat draws the discourse structure annotation on top of the text. However, Kirschner et al. (2015) argued that the visual in GraPat might be confusing for texts with multiple long sentences. Their solution to the problem, DiGAT, splits the display into a text and its structure view; this design have been followed by both OVA and TIARA.

DiGAT (Fig. 8) and OVA (Fig. 9) present both text and structural views simultaneously, but in DiGAT’s structure view, the text corresponding to a node is not shown, and text and nodes are associated by IDs instead. We think it is essential to see both text and structure in the same screen as OVA and TIARA do, because it is cognitively expensive to synthesise two views in one’s mind by switching between the left and right side of the screen as in DiGAT. Furthermore, we believe it is more advantageous to switch between text and tree views instead of presenting them simultaneously. Human brains are unable to simultaneously process all visual information. Visual attention that is focused on a small area (single task) enables performance benefits while distributing attention over a large area (multiple tasks in parallel) incurs penalties (Evans et al., 2011; Sun et al., 2015). In our case, annotators have to analyse both the logical sequencing of sentences and the overall discourse structure, which are complex and cognitively demanding. Thus, the implementation of TIARA’s dual-view allows annotators to focus on one type of analysis at one time, that is, either logical sequencing or the overall discourse structure.

Among the annotation tools that assume graph structure, we consider OVA as our strongest competitor. It offers almost all features needed for our project, with the exception of the discourse unit reordering feature. However, it has to be noted that the modification of OVA to support this feature is not straightforward. Concerning the visual elegance, TIARA is more advantageous for annotating longer texts compared other tools since it offers features to reduce clutter on the display, for instance, box-resize and indentation features in the text view, and fold/unfold and shrink/enlarge features in the tree view. These features are crucial for the annotators since confusing visual display hinders quality annotation.

Overall, there is no one-for-all discourse annotation tool, but TIARA with its middle-ground visual solution is efficient for annotation and is a strong general tool for relation-focused discourse annotation. In particular, because it provides versatile visualisation for representing structure (the dual-view, clutter-reducing features), annotators can choose the method that works best for them. Despite its advantages,

TIARA does not provide a text segmentation feature. Segmentation is an activity of splitting text into discourse units. Our annotation scheme operates at the sentence level, that is, one sentence automatically corresponds to a single discourse unit. Other annotation studies typically used the idiosyncratic definition of discourse units, which requires manual text segmentation (Lippi & Torrioni, 2016). It certainly would be helpful if TIARA supports the manual text segmentation, but there is no consensus on the definition of a clause that corresponds to a discourse unit; that is, we cannot ensure the "annotation compliance" aspect. At the current stage, we treat the text segmentation as a separate "pre-processing" step, that is, the input text has to be already pre-segmented beforehand.

TIARA was originally designed for the annotation of relations between discourse units and sentence rearrangement in monologue text, as motivated in Sect. 2.2. Four link labels were used (three directed and one undirected). Two annotators used an earlier version of TIARA to annotate a total of around 434 short (200–300 words) argumentative essays from the ICNALE corpus (Putra et al., 2021). Our preliminary annotation study in Putra et al. (2019) showed that the annotation time compared to using a spreadsheet (representing a "general-purpose" tool) was reduced from 40 to 25 mins.¹⁸ Since then, TIARA has found another use case, which is described in the next section.

5 Possible application of TIARA 2.0 for teaching argumentation

5.1 Mind mapping tools for analysing argumentation

Mind mapping tools have previously been used in teaching how to visualise arguments (Cullen et al., 2018).¹⁹ Mind maps are commonly radiantly structured, emerging outwards from a central node; the centre represents a key idea, while the second-level nodes represent secondary thoughts or supporting ideas, and so on (Buzan & Buzan, 1993). Thus, mind maps may illustrate the relationship between ideas. Not only the radiant structure, existing mind mapping tools such as XMind²⁰ or MindMup²¹ can also visualise the hierarchical structure, which is a natural representation for argumentation.

There are three basic functions in mind mapping tools: (1) adding nodes representing new ideas, (2) establishing links between nodes and (3) visualising the structure. However, mind mapping tools cannot show how structures (relations between concepts) can be serialised into coherent texts, which is important in

¹⁸ We did not measure the difference in agreement. In our trial annotations, there were some modifications in the annotation guideline. Although the annotation tasks were the same, the explanation of the relation labels were changed. Therefore, measuring the difference in the agreement was not possible under the same condition.

¹⁹ Other than argumentation, mind mapping tools can also be useful in vocabulary acquisition, that is, to illustrate synonym or antonym networks (Betancur & King, 2014; Masoud & Ibrahim, 2017). It can also be used to facilitate brainstorming (Buran & Filyukov, 2015).

²⁰ <https://www.xmind.net/>.

²¹ <https://www.mindmup.com/tutorials/argument-visualization.html>.

learning-to-write. TIARA offers those three basic functions, and additionally, show how the hierarchical structure of ideas can be serialised into a coherent text (via the text view and reordering functionality). Therefore, TIARA can be a good alternative for mind-mapping tools, particularly for teaching how to analyse or write an argument.

5.2 Teaching argumentation

Teaching students to argue well is difficult, because so many constraints need to be satisfied for an argument to be convincing; the text has to contain the desired argumentative elements. By means, the ideas should be clearly stated, connected to each other, and supported by reasons. They should also be logically developed in a particular sequencing such as time or importance, and accompanied by appropriate discourse connectives. Only then can the writing ultimately communicate the desired ideas as a whole (Jacobs et al., 1981; Matsumura & Sakamoto, 2021; Peldszus & Stede, 2013; Reed & Wells, 2007; Toulmin, 2003). Teaching EFL students is even more difficult since they require a different instruction compared to native speakers, particularly on the organisation of ideas that would be perceived coherent in the eyes of native speakers (Bacha, 2010; Connor, 2002; Kaplan, 1966; Silva, 1993).

Existing teaching pedagogy relies on verbal and written expressions as forms of feedback to students, based on teachers' evaluation of classroom writing exercises. Corrective feedback such as marginal-comments, end-comments and editing codes (e.g., circles, underlines) are often used to provide corrective feedback for errors in language use (Biber et al., 2011). These types of feedback are useful for a binary judgement, that is, whether the use of language (commonly grammar and vocabulary) is correct or wrong (Cumming et al., 2001, 2002). However, logical sequencing and argumentation judgements involve analysing the degree of appropriateness, which is hard to express using corrective feedback (Cumming et al., 2001, 2002). For example, it is not effective when explaining why an argument is one-sided or imbalanced.

Teaching how to argue can be supported by the construction of mental models of the argumentative structure, which require checks for completeness (*are all the parts there?*) and for coherence (*do relations among parts make sense?*). Teaching such skills with purely symbolic means (the use of words) is less efficient than using visual explanations, for instance, in the form of diagrams. Visual information can also act as an intuitive platform for inference (Bobek & Tversky, 2016). For these reasons, visual information has been widely used to promote effective communication. But despite its benefits, the analysis of the implicit logical structure of argumentative texts is rarely taught explicitly (Cullen et al., 2018).

5.2.1 Learning-to-read and diagnostic assessment

Cullen et al. (2018) assessed the effect that argument visualisation has on analytical reasoning and argument understanding. They performed a controlled study where

one group of students were taught how to visualise arguments using MindMup, whereas the control group was taught traditionally. The targeted texts were contemporary academic texts. When measuring the improvement of both groups in logical reasoning test before and after the teaching sessions, they found a larger increase in the visually-taught group than in the control group, suggesting that learning how to visualise arguments led to improvements in students' analytical-reasoning skills.

Beyond the benefits of checking for completeness, coherence and inference, argument visualisation also helps in conveying what students understand about the texts. The graphical visualisation of an argument can be shared with instructors, allowing students to easily discuss their interpretations with the instructors. This enables instructors to quickly identify gaps in students' understanding of the reading material, and hence suggest ways for improving their work. This feedback, in turns, should enable students to produce more accurate and effectively structured essays (Cullen et al., 2018). In the EFL context, the argument visualisation can also help students to understand the differences between the structure of their writings and native ('good') writings, i.e., *contrastive analysis* (Bacha, 2010; Kaplan, 1966; Silva, 1993).

On the instructors' side, Matsumura and Sakamoto (2021) studied in detail how the analysis of argument visualisation is helpful for diagnostic assessment. They used an earlier version of TIARA 1.0 to investigate organisation problems in texts written by Japanese EFL learners. They defined six types of directed link: five inspired by Toulmin's (2003) argumentation model, and a special link labelled "?" for feedback to students. Four annotators²² annotated 50 short (~140 words) argumentative essays written by 50 Japanese EFL learners (11th grade) in a classroom setting.²³ Additionally, they also assigned scores to these essays.

Their analysis aimed to assess coherence and organisational problems in EFL texts. For example, whether textual segments written by their EFL students were logically connected and relevant to each other, whether material presented as if it was supporting some claim was indeed relevant to the target claim (marked "?" when it was not), and whether ideas are properly arranged. Instructors can analyse the texts from multiple viewpoints using the text and tree views of TIARA: logical sequencing, sentence-to-sentence relationships, and the overall structure.

The annotations by instructors were then used to provide evidence-based feedback to the students.²⁴ Sentences marked with the questionable label (i.e., "?") provide the students with the specific errors that should be addressed in their writings. Students are then asked to review the annotations on their own and read through the whole texts for planning revisions. Matsumura and Sakamoto (2021) also collected students' comments on their experience with learning using argument visualisation. In general, students mentioned that the visualisation enabled them to grasp the overall balance of ideas in their writings and review their structures. Furthermore, the annotation of questionable relations indeed enabled them to spot

²² Two English teachers and two college students

²³ Their approximated proficiency is CEFR A2.

²⁴ Other types of corrective comments such as end-comments or editing codes were not provided.

the problems in their writings. In turn, the visualisation inspires them to perform revisions, particularly to clarify the problematic sentences.

From the instructors' viewpoint, the diagnostic assessment enables them to make inference about learners' strengths and weaknesses in the skills being taught (Jang & Wagner, 2013); in this case, Matsumura and Sakamoto (2021) found a remarkable difference between the discourse structures in the high-scored essays where they typically form a balanced tree and those in low-scored ones where the overall structure tends to be flat and linear, and isolated elements occur. The visual feedback enables students to comprehend and accept why certain ways of writing are considered logically weak, and thus, achieve poor scores. The coherent organisation of ideas is one of the most difficult textual aspects to assess because it is highly subjective (Todd et al., 2004). Here, the visualisation in TIARA can facilitate discussions among instructors as they can share each other's annotations and discuss their interpretations of the students' writings. In the long run, observations gained by annotating student texts, such as by Matsumura and Sakamoto (2021), can be used to formulate hypotheses about better assessment and teaching pedagogy of argumentative writing.

5.2.2 *Learning-to-write*

So far, we have established the benefits of argument analysis in education, particularly in the learning-to-read scenario. The next step in education is to verify whether students indeed have learnt and understood the teaching material; this is usually done by asking them to produce texts. Learning-to-write is important on the road to mastery of a language, moving from comprehension to production of well-thought piece of writing (Cole & Feng, 2016). Production is also often a process of discovery, as writers have to learn new knowledge to become even better writers (Suleiman, 2000). In the pragmatic viewpoint, the ability to write good argumentation is critical for getting awards and research funding; not to mention, the success of teachers and teaching methods are also evaluated based on students' ability to write (Hosseini et al., 2013).

One of the advantages of TIARA 2.0 compared to other tools is that can be used in the learning-to-write. In a classroom setting, students could write argumentative essays and simultaneously draw the intended structures on TIARA in parallel, allowing instructors to interactively and quickly point out and address those student mistakes that are visible in TIARA's visualisation. Instructors can then suggest improvements in the overall discourse flow (e.g., by reordering sentences), in the textual realisation (e.g., by editing discourse connectives) and argumentation (e.g., by adding more sentences for a stronger or more balanced argument) (Cho & MacArthur, 2010; Crossley & McNamara, 2016; Matsumura & Sakamoto, 2021). All of these can be performed directly in TIARA 2.0, while other tools typically do not support such an activity. Thereby, our tool should enhance the process of student-instructor communication and feedback during writing and revising stages. However, the classroom trial of the learning-to-write scenario has not been conducted yet, and we leave it for the future.

6 Conclusion

This article presents TIARA 2.0, a new web-based annotation tool for annotating argumentative discourse structure. On top of this, it is also designed to be useful for educational purposes. To this end, TIARA supports a sentence *re*-arrangement annotation followed by textual editing (of connectives and pronouns), and a content alteration annotation by adding, deleting or modifying sentences. These features are particularly unique compared to existing tools.

TIARA provides versatile visualisation to enhance structural annotation. Particularly, in the dual-view display, annotators can analyse texts from both the logical-sequencing and the overall structure viewpoints. The visual-related functions, such as indentation, box-resize, fold/unfold, and shrink/enlarge features can also help to reduce clutter on the display when annotating long texts. The tool is easily customisable via a configuration script. TIARA has been used to annotate hundreds of texts in discourse annotation studies and has also proved its usefulness for education in the analysis and construction of arguments.

Future versions of TIARA will improve the visualisation by easy comparisons of original and edited text. In addition, we plan to allow relations between nodes and edges, for example, the *undercut* relation. It would also be useful to enhance and speed up the annotation process by providing an automatic mode where the tool recommends the annotation on the fly; annotators only need to evaluate the accuracy of the automatically provided annotation and provide changes when needed. On the purely technical side, the current version of TIARA is appropriate for relatively small-scale projects, while for bigger and more complex projects, an additional management feature would improve the experience substantially. We therefore consider the provision of two parallel versions of TIARA: a light-weight client-side TIARA versus one with more extensive management, collaboration and monitoring features.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Bacha, N. N. (2010). Teaching the academic argument in a university EFL environment. *Journal of English for Academic Purposes*, 9(3), 229–241.
- Bamberg, B. (1983). What makes a text coherent. *College Composition and Communication*, 34(4), 417–429.
- Betancur, M. I. G., & King, G. (2014). Using mind mapping as a method to help ESL/EFL students connect vocabulary and concepts in different contexts. *Trilogia*, 10, 69–85.

- Biber, D., Nekrasova, T., & Horn, B. (2011). The effectiveness of feedback for L1-English and L2-writing development: A meta-analysis. *ETS Research Report Series*, TOEFL iBT™ Research Report.
- Bobek, E., & Tversky, B. (2016). Creating visual explanations improve learning. *Cognitive Research: Principles and Implications*, 1(1), 1–14.
- Buran, A., & Filyukov, A. (2015). Mind mapping technique in language learning. *Procedia - Social and Behavioral Sciences*. XVth International Conference "Linguistic and Cultural Studies: Traditions and Innovations", 206, 215–218. ISSN 1877-0428.
- Buzan, T., & Buzan, B. (1993). *The mind map book: How to use radiant thinking to maximize your Brans untapped potential*. Plume.
- Carlile, W., Gurrapadi, N., Ke, Z., & Ng, V. (2018). Give me more feedback: Annotating argument persuasiveness and related attributes in student essays. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 621–631, July.
- Cho, K., & MacArthur, C. (2010). Student revision with peer and expert reviewing. *Learning and Instruction*, 20(4), 328–338.
- Cole, J., & Feng, J. (2016). Effective strategies for improving writing skills of elementary English language learners. In *Chinese American Educational Research and Development Association Annual Conference*.
- Connor, U. (2002). New directions in contrastive rhetoric. *TESOL Quarterly*, 36(4), 493–510.
- Crossley, S. A., & McNamara, D. S. (2016). Say more and be more coherent: How text elaboration and cohesion increase writing quality. *Written Research*, 7(3), 351–370.
- Crossley, S. A., Kyle, K., & McNamara, D. S. (2016). The tool for the automatic analysis of text cohesion (taaco): Automatic assessment of local, global, and text cohesion. *Behavior Research Methods*, 48, 1227–1237.
- Cullen, S., Fan, J., van der Brugge, E., & Elga, A. (2018). Improving analytical reasoning and argument understanding: A quasi-experimental field study of argument visualization. *NPJ Science of Learning*, 3(21), 1–6.
- Cumming, A. H., Kantor, R., Powers, D. E. (2000). prototype writing tasks: An investigation into raters decision making and development of a preliminary analytic framework. *TOEFL Monograph Series*, 22, 94.
- Cumming, A. H., Kantor, R., & Powers, D. E. (2002). Decision making while rating ESL/EFL writing tasks: Descriptive framework. *The Modern Language Journal*, 87, 67–96.
- De Kuthy, K., Reiter, N., & Riester, A. (2018). QUD-based annotation of discourse structure and information structure: Tool and evaluation. In *Proceedings of the International Conference on Language Resources and Evaluation*, pp. 1932–1938.
- Evans, K. K., Horowitz, T. S., Howe, P., Pedersini, R., Reijnen, E., Pinto, Y., et al., (2011). Visual attention. *WIREs. Cognitive Science*, 2(5), 503–514.
- Garing, A. G. (2014). Coherence in argumentative essays of first year college of liberal arts students at de la salle university. *DLSU Research Congress*.
- Grosz, B. J., & Sidner, C. L. (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3), 175–204.
- Han, N.-R., Chodorow, M., & Leacock, C. (2006). Detecting errors in English article usage by non-native speakers. *Natural Language Engineering*, 12(2), 115–129.
- Hirst, G., & Budanitsky, A. (2005). Correcting real-word spelling errors by restoring lexical cohesion. *Natural Language Engineering*, 11(1), 87–111.
- Hosseini, M., Taghizadeh, M. E., Abedin, M. J., & Naseri, E. (2013). In the importance of EFL learners writing skill: Is there any relation between writing skill and content score of English essay test. *International Letters of Social and Humanistic Sciences*, 6, 1–12.
- Hsin, L. B., & Snow, C. E. (2020). Arguing for teachers and for friends: Eighth-graders sensitivity to argumentation features when judging and revising persuasive essays. *Discourse Processes*, 57(10), 823–843.
- Iida, R., & Tokunaga, T. (2014). Building a corpus of manually revised texts from discourse perspective. In *Proceedings of the International Conference on Language Resources and Evaluation*, pp. 936–941.
- Ishikawa, S. (2013). The icnale and sophisticated contrastive interlanguage analysis of Asian learners of English. *Learner Corpus Studies in Asia and the World*, 1, 91–118.
- Ishikawa, S. (2018). The icnale edited essays: A dataset for analysis of L2 English learner essays based on a new integrative viewpoint. *English Corpus Linguistics*, 25, 1–14.

- Jacobs, H. L., Zinkgraf, S. A., Wormuth, D. R., Harfiel, V. F., & Hughey, J. B. (1981). *Testing ESL composition: A practical approach*. Rowley: Newbury House.
- Jang, E. E., & Wagner, M. (2013). Diagnostic feedback in the classroom. In K. Antony (Ed.), *Companion to language assessment, vol 2, chapter 6*. Oxford: Wiley-Blackwell.
- Janier, M., Lawrence, J., & Reed, C. (2014). Ova+: An argument analysis interface. In *International Conference on Computational Models of Argument*, pp. 463–464.
- Johns, A. M. (1986). The ESL student and the revision process: Some insights from schema theory. *Journal of Basic Writing*, 5(2), 70–80.
- Kaplan, D., Iida, R., & Tokunaga, T. (2010). Annotation process management revisited. In *Proceedings of the International Conference on Language Resources and Evaluation*, pp. 3654–3661.
- Kaplan, R. B. (1966). Cultural thought patterns in inter-cultural education. *Language Learning*, 16(1–2), 1–20.
- Kirschner, C., Eckle-Kohler, J., & Gurevych, I. (2015). Linking the thoughts: Analysis of argumentation structures in scientific publications. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pp. 1–11.
- Klie, J.-C., Bugert, M., Boullosa, B., Castilho, R. E. de, & Gurevych, I. (2018). The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pp. 5–9, August.
- Lee, J., & Webster, J. (2012). A corpus of textual revisions in second language writing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 248–252.
- Lippi, M., & Torroni, P. (2016). Argumentation mining: State of the art and emerging trends. *ACM Transaction of Internet Technology*, 16(2), 10:1–10:25.
- Little, A., & Tratz, S. (2016). EasyTree: A graphical tool for dependency tree annotation. In *Proceedings of the International Conference on Language Resources and Evaluation*, pp. 2343–2347.
- Mann, W. C., & Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3), 243–281.
- Masoud, H. M. A., & Ibrahim, O. L. (2017). The effectiveness of using an e-mind mapping software based program in developing faculty of education 2nd year English majors vocabulary acquisition and use. *Journal of Research in Curriculum, Instruction and Educational Technology*, 3, 177–224.
- Matsumura, K., & Sakamoto, K. (2021). A structure analysis of Japanese EFL students' argumentative paragraph writing with a tool for annotation discourse relation. *The Bulletin of the Writing Research Group, JACET Kansai Chapter*, 14.
- O'Donnell, M. (2000). RSTTool 2.4: A markup tool for Rhetorical Structure Theory. In *INLG'2000 Proceedings of the First International Conference on Natural Language Generation*, pp. 253–256, Mitzpe Ramon, Israel, June 2000.
- Peldszus, A., & Stede, M. (2013). From argument diagrams to argumentation mining in texts: A survey. *International Journal of Cognitive Informatics and Natural Intelligence*, 7(1), 1–31.
- Peldszus, A., & Stede, M. (2016). An annotated corpus of argumentative microtexts. In *Argumentation and Reasoned Action - Proceedings of the 1st European Conference on Argumentation*.
- Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., & Webber, B. (2008). The Penn discourse TreeBank 2.0. In *Proceedings of the International Conference on Language Resources and Evaluation*.
- Putra, J. W., Teufel, S., & Tokunaga, T. (2019). An argument annotation scheme for the repair of student essays by sentence reordering. In *Proceedings of Annual Meeting of the Association for Natural Language Processing Japan*, pp. 546–549.
- Putra, J. W., Teufel, S., Matsumura, K., & Tokunaga, T. (2020). TIARA: A tool for annotating discourse relations and sentence reordering. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 6912–6920, May.
- Putra, J. W., Teufel, S., & Tokunaga, T. (2021). Annotating argumentative structure in English-as-foreign-language learner essays. *Natural Language Engineering*. <https://doi.org/10.1017/S1351324921000218>.
- Rabinovich, E., Nisioi, S., Ordan, N., & Wintner, S. (2016). On the similarities between native, non-native and translated texts. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1870–1881. Association for Computational Linguistics, August.

- Reed, C., & Wells, S. (2007). Dialogical argument as an interface to complex debates. *IEEE Intelligent Systems*, 22(6), 60–65.
- Silva, T. (1993). Toward an understanding of the distinct nature of l2 writing: The ESL research and its implications. *TESOL Quarterly*, 27(4), 657–677.
- Sonntag, J., & Stede, M. (2014). GraPAT: A tool for graph annotations. In *Proceedings of the International Conference on Language Resources and Evaluation*, pp. 4147–4151.
- Stab, C., & Gurevych, I. (2017). Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3), 619–659.
- Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., & Tsujii, J. (2012). BRAT: A web-based tool for NLP-assisted text annotation. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pp. 102–107.
- Strobl, C., Ailhaud, E., Benetos, K., Devitt, A., Kruse, O., Proske, A., & Rapp, C. (2019). Digital support for academic writing: A review of technologies and pedagogies. *Computers & Education*, 131, 33–48.
- Suleiman, M. F. (2000). The process and product of writing: Implications for elementary school teachers. Educational Resources Information Center.
- Sun, Y., Wu, S., & Spence, I. (2015). The commingled division of visual attention. *PLOS ONE*, 10(6), 1–18.
- Todd, R. W., Thienpermpool, P., & Keyuravong, S. (2004). Measuring the coherence of writing using topic-based analysis. *Assessing Writing*, 9(2), 85–104.
- Toulmin, S. E. (2003). *The uses of argument* (2nd ed.). Cambridge University Press.
- Wolf, F., & Gibson, E. (2005). Representing discourse coherence: A corpus-based study. *Computational Linguistics*, 31(2), 249–288.
- Yimam, S. M., Gurevych, I., Castilho, R. E. de, & Biemann, C. (2013). WebAnno: A flexible, web-based and visually supported system for distributed annotations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 1–6.
- Yuan, Z., & Briscoe, T. (2016). Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 380–386. Association for Computational Linguistics.
- Zeldes, A. (2016). rstWeb: A browser-based annotation interface for Rhetorical Structure Theory and discourse relations. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 1–5.
- Zhang, F., Hashemi, H. B., Hwa, R., & Litman, D. (2017). A corpus of annotated revisions for studying argumentative writing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1568–1578.