TOWARDS BUILDING AN INTELLIGENT REVISION ASSISTANT FOR ARGUMENTATIVE WRITINGS

by

Fan Zhang

B.Eng. in Software Engineering, Wuhan University, 2009

M.S. in Software Engineering, Wuhan University, 2011

M.S. in Computer Science, University of Pittsburgh, 2016

Submitted to the Graduate Faculty of

the Kenneth P. Dietrich School of Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2017

UNIVERSITY OF PITTSBURGH

DEPARTMENT OF COMPUTER SCIENCE

This dissertation was presented

by

Fan Zhang

Dr. Diane Litman, Dietrich School of Arts and Sciences, University of Pittsburgh

Dr. Rebecca Hwa, Dietrich School of Arts and Sciences, University of Pittsburgh

Dr. Adriana Kovashka, Dietrich School of Arts and Sciences, University of Pittsburgh

Dr. Christian Schunn, Dietrich School of Arts and Sciences, University of Pittsburgh

Dissertation Director: Dr. Diane Litman, Dietrich School of Arts and Sciences, University of Pittsburgh

TOWARDS BUILDING AN INTELLIGENT REVISION ASSISTANT FOR ARGUMENTATIVE WRITINGS

Fan Zhang, PhD

University of Pittsburgh, 2017

Current intelligent writing assistance tools (e.g. Grammarly, Turnitin, etc.) typically work by locating the problems of essays for users (grammar, spelling, argument, etc.) and providing possible solutions. These tools focus on providing feedback on a single draft, while ignoring feedback on an author's changes between drafts (revision). This thesis argues that it is also important to provide feedback on authors' revision, as such information can not only improve the quality of the writing but also improve the rewriting skill of the authors. Thus, it is desirable to build an intelligent assistant that focuses on providing feedback to revisions.

This thesis presents work from two perspectives towards the building of such an assistant: 1) a study of the revision's impact on writings, which includes the development of a sentence-level revision schema, the annotation of corpora based on the schema and data analysis on the created corpora; a prototype revision assistant was built to provide revision feedback based on the schema and a user study was conducted to investigate whether the assistant could influence the users' rewriting behaviors. 2) the development of algorithms for automatic revision identification, which includes the automatic extraction of the revised content and the automatic classification of revision types; we first investigated the two problems separately in a pipeline manner and then explored a joint approach that solves the two problems at the same time.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

I believe I will always view my years at PITT as a precious, pleasant and challenging experience. Starting from a novice who barely knows any basic research skills, I am glad that now I am able to contribute to the NLP research community after all these years' training. Life at Pittsburgh is like a box of chocolate. It is is mixed with feelings of bitterness, happiness and many other kinds of emotions. While the thought of quitting PhD came out from time to time in my first few years here, I am glad that I choose to continue working. Looking back to all the experiences throughout the years, I feel that the whole PhD experience is not only teaching me the lesson of how to be a researcher but also the lesson of how to be a responsible unit of the society. I am also glad that I get the chance to work on a research topic that I like, to develop an intelligent writing assistant that can help people make better writings.

This thesis would not be possible without the help of many people. First I would like to thank my advisor, Diane Litman. The working experience with her is more than amazing. She showed great patience in the process of leading me to the NLP research. I can still remember the time when she spent a lot of time with me on a short workshop paper. I still feel embarrassed about my ignorance of paper writing at the time but she manages to teach me the correct path with great patience. She also taught me many rules that are crucial to every job: stay organized and do not procrastinate. I have also been lucky to learn from incredible mentors, including my committee members, Rebecca Hwa, Christian Schunn and Adriana Kovashka. Their valuable suggestions and comments on this thesis have polished it from a summary of my publications to an organized piece of work. I am also thankful for all the annotators that have worked on the annotation of the revision corpora, especially Jiaoyang Li, their excellent work greatly expedite the process of my thesis research.

I am thankful for the amazing people that accompanied me through my PhD study. I feel so lucky that I could meet my girlfriend Wenchen in PITT. She has always been a great supporter of me since we started our relationship. The road to graduation is much less tiring with her companion. I also want to thank my parents and grandparents, who always sent their care from the other side of the earth. I also want to thank all the friends and colleagues that I met here, I learned a lot from them and their accompany simply make the life at PITT more colorful.

1.0 INTRODUCTION

1.1 BACKGROUND

According to a recent national assessment of writing in the United States (NCES, 2011), 73 % of high school seniors demonstrated only a basic or even worse understanding of the knowledge and skills that are fundamental for competent writing. One key bottleneck is the limitation of teaching resources. The improvement of writing requires practices and regular feedback. However, there are not sufficient teaching resources to meet the requirement. The development of Natural Language Processing (NLP) techniques provides one potential solution. Techniques such as automatic scoring of grammar or argumentation structures can inspire students to improve their essays accordingly (Attali and Burstein, 2006; Graesser et al., 2012). Based on these techniques, multiple commercial products have been developed (Grammarly, 2016; Turnitin, 2016; Draft, 2015).

While a lot of works has been done to provide feedback on the problems of a single essay (**what to improve**), none of them paid enough attention to the process of rewriting (**how to improve**). The skill of rewriting is considered to be an important skill for successful writing. According to (Faigley and Witte, 1981), experienced writers revise in ways different from inexperienced writers. Learning to revise is a critical part. Through rewriting, students are likely to have questions such as "What kind of revisions are more likely to improve my essay scores?", "Does my revision achieve the effect that I want?", "Is the problem mentioned in the review resolved by my revision?", etc. Providing feedback on students' revisions allows the students to make their revisions more effectively and further acquire the skills of rewriting. Comparing to existing revision assistance systems such as WriteLab (Writelab, 2015), Turnitin (Turnitin, 2016) and Draft (Draft, 2015) which aims at

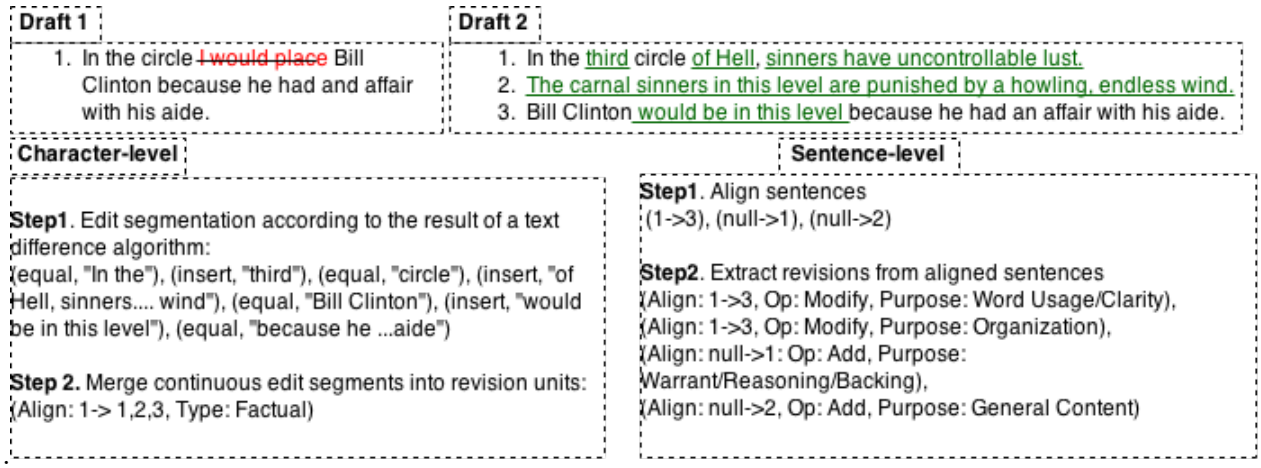


Figure 1.1: A character/word level revision extraction approach typically extracts differences between drafts at the character level to get edit segments. Sentence 1 in Draft 1 is wrongly marked as being modified to 1, 2, 3 in Draft 2 because character-level text comparison could not identify the semantic similarity between sentences.

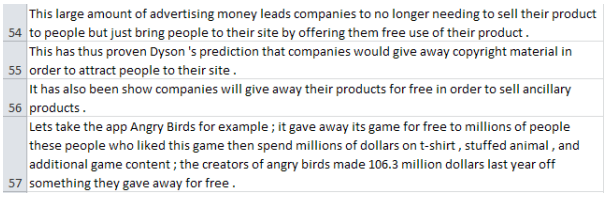
detecting the problems of the essay and providing solution suggestions, I propose to target the automatic suggestions on the revision process directly.

This work focuses on argumentative writing revisions as most of the collected corpora are argumentative writings. Argumentative writing is a common form of writing in education. Argumentation plays an important role in analyzing many types of writing such as persuasive essays (Stab et al., 2014), scientific papers (Teufel, 2000) and law documents (Palau and Moens, 2009). A preliminary study by us indicates that it is possible to extend the current works to other writing genres.

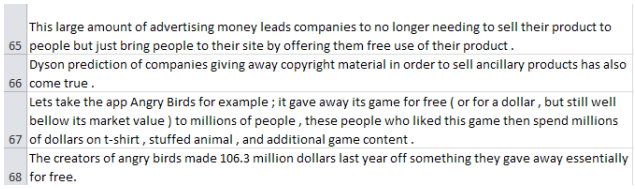
1.2 RESEARCH OVERVIEW

Two major problems were addressed in this thesis.

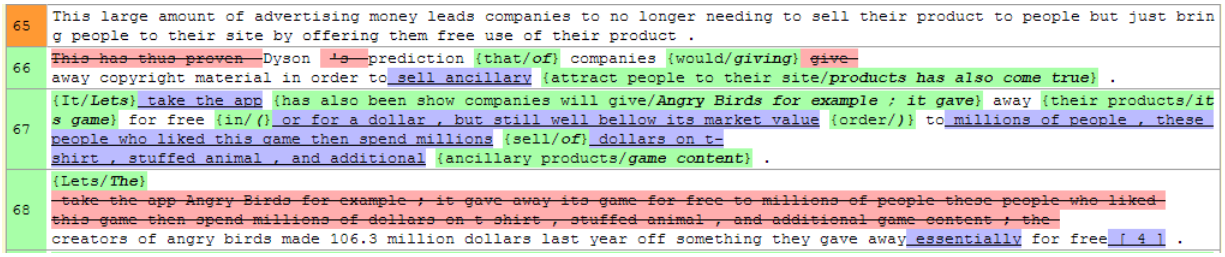
The description of revisions. The challenges involve both the decision of revision



(a) first draft



(b) final draft



(c) Revision detection using text-diff (Hashemi and Schunn, 2014)

Figure 1.2: Fragments of a paper, green for recognized modifications, blue for insertions and red for deletion

granularity and the definition of revision categories. First, different levels of granularity are used in prior revision researches (Bridwell, 1980; Bronner and Monz, 2012; Iida and Tokunaga, 2014; Ferschke et al., 2011). As shown in Figure 1.1, the levels of granularity influence the description of the revisions. Intuitively the word/character level granularity is the most precise; however, the automatic extractions of word/character level revisions can be error-prone in practice because the semantic information cannot be easily incorporated into the extraction process. The level has to be chosen for both the convenience of manual annotation and automatic revision extraction. Second, most of the prior revision studies only focus on the correction of spelling and grammar mistakes (Xue and Hwa, 2014; Mizumoto et al., 2011). The categorization of content revisions is typically not fine-grained. A common content-oriented categorization is a binary classification of revisions according to whether the information of the essay is changed or not (e.g. text-based vs. surface as defined by (Faigley and Witte, 1981)). This categorization ignores potentially important differences between revisions under the same high-level category. For example, changing the evidence

of a claim and changing the reasoning of a claim are both considered as text-based changes. Usually changing the evidence makes a paper more grounded, while changing the reasoning helps with the paper's readability. Thus, it is necessary to define revision categories for the description of argumentative revision purposes. It would also be helpful if the defined categories can be extended for more generic revision descriptions. The defined categories should be both distinguishable to humans to allow reliable annotation and pedagogically meaningful to demonstrate the impact differences on writings of different revision types. A revision assistant built on the defined schema should have a positive impact on the user's revision behaviors.

The identification of revisions. The identification of the revisions involve locating revisions (**revision extraction**) and categorizing revisions (**revision classification**). As shown in Figure 1.2, the location of revisions cannot be simply solved with a text difference algorithm. A traditional textual difference algorithm is likely to generate overwhelming revision information when there are heavy edits to the essay. In our work, we propose to solve it as a variation of parallel corpus alignment problem using both semantic similarity and global sentence ordering. For the classification of revisions, while there are works on Wikipedia revision classification (Bronner and Monz, 2012; Daxenberger and Gurevych, 2013), there is a lack of work on revision classification in other datasets such as student writings. It is not clear whether current features and methods can still be adapted. Thus, it is necessary to investigate whether the existing approaches can be applied on our task of argumentative rewriting classification. Also, many types of useful information have not been fully utilized in revision classification. Existing works (Adler et al., 2011; Bronner and Monz, 2012; Daxenberger and Gurevych, 2013) typically compare two versions of a text to extract revisions and then classify the purpose of each revision in isolation. While limited contextual features such as revision location have been utilized, such features are computed from the revision being classified but typically not its neighbors. This thesis presents new features and approaches to improve revision classification. A joint approach that identifies the location and type of revisions at the same time is also proposed.

1.3 RESEARCH HYPOTHESES

In the process of addressing the problems above, the following hypotheses were explored.

For the first problem, the description of revisions, we have the following hypotheses:

- H1.1. We can define a revision schema that can be reliably annotated by human.
- H1.2. There is significant correlation between the number of revisions and the writing improvement. Different types of revisions have different impacts.
- H1.3. There is a difference in participants' revising behaviors depending on which aspects of the revision schema are used to provide feedback.
- H1.4. Revision feedback based on the defined schema will inspire the users to further modify their revision.

For the second problem, the automatic identification of revisions, we have the hypotheses:

- H2.1. The existing features and approaches in Wikipedia revision classification can be adapted to the prediction of argumentative writing revisions.
- H2.2. Using contextual features can improve the classification performance.
- H2.3. Using discourse information can improve the classification performance.
- H2.4. The identification of revision location and revision type can be jointly predicted.

1.4 RESEARCH METHODOLOGY

To test the hypothesis H1.1, we followed the steps of previous works in developing corpora. We first defined the representations of the tasks and then developed manual annotation schemes in iterations. Inter-annotator agreement studies were conducted to validate the reliability of the annotation schemes. According to the commonly used convention in the field of NLP (Wilson, 2008), an agreement value of 0.80 allows for firm conclusions to be made, and a value of at least 0.67 is sufficient for drawing tentative conclusions. We consider an agreement value of 0.67 to support the hypothesis and a higher agreement value indicates stronger evidence. Corpora were annotated based on the schema proposed. Based on the

corpora collected, data analysis is conducted for H1.2. Partial correlation controlling for the pre-test score is one common way to measure learning *gain* in the tutoring literature (Baker et al., 2004). Similarly, we tested the partial correlation between the number of revisions and the post-test score. A significance test result will suggest the correctness of H1.2. To test the hypotheses H1.3 and H1.4, we created a prototype revision assistant that provided revision feedback based on the defined revision schema. Another naive revision assistant was developed to be compare with the developed assistant. A user study was conducted by asking the participants to write 2 drafts in advance and then revise based on the feedback on their previous revisions. Subjective feedback provided by the participants and the objective number of revisions were used to measure the user's revision behaviors. To test H1.3, we choose to test the difference between different revision feedback using ANOVA test with two factors. The experiment type (experiment feedback or control feedback) is used as one factor. A significant result will suggest the correctness of H1.3. To test H1.4, we counted the number of revisions that got further revised. H1.4 can be supported by the observation that the ratio of being further revised is higher than average when the assistant's feedback differs from the participants' own recognition.

To test the hypothesis H2.1, we first followed the prior works on Wikipedia revision classification (Adler et al., 2011; Javanmardi et al., 2011). A SVM model using only unigram features was used as the baseline. Features and methods from prior works were repeated on our corpora. We can demonstrate the correctness of H2.1 if we observe a significantly better performance with the features from prior works. Unweighted average F-measure is used as the evaluation metric. 10-fold cross-validation is conducted and t-test is conducted to check whether the performance is significantly better. We then explore whether we can improve the performance to demonstrate the correctness of H2.2, H2.3 and H2.4. For H2.2, we explored from two perspectives: 1) extract features from the sentences nearby, 2) utilize a sequence model Conditional Random Fields (CRF) to utilize the contextual dependency between revisions. For H2.3, we utilize Penn-Discourse Treebank (PDTB) (Prasad et al., 2008) to represent the discourse information. Deciding that the local PDTB information might not be sufficient for improving the classification performance, we propose new approaches to infer long-distance PDTB information for our investigation. For H2.4, we choose to generate a se-

quential representation of the sentence alignment information and combine such information with the type of revisions. Then the joint problem is transformed into a sequence labeling problem. We compare the results with all the prior works to demonstrate the correctness of H2.4.

1.5 RESEARCH SUMMARY

We defined a sentence-level revision schema to describe the argumentative writing revisions (Zhang and Litman, 2015). Sentences across drafts are aligned and the revision types are labeled on the aligned sentence pairs. The revisions are categorized to two major types: *Content* and *Surface*. The *Content* revisions are categorized to 5 types according to the argumentation role that has been changed: *Claim/Ideas*, *Warrant/Reasoning/Backing*, *General Content*, *Rebuttal/Reservation* and *Evidence*. The *Surface* revisions describe the surface changes that does not change the content of the essay. They are categorized as *Conventions/Grammar/Spelling*, *Word Usage/Clarity* and *Organization*. Revisions are annotated based on the schema. Multiple corpora have been developed to serve as the resources for this thesis and further research. The corpora consist of essays written by high school students, undergraduate and graduate students. Two of the corpora have essays graded by experts for revision schema analysis. The other corpora are used for automatic revision identification and revision behavior analysis. One of the collected corpora, the ArgRewrite Corpus (Zhang et al., 2017) is made publicly available[1].

To study the schema, we first tested whether the revisions could be reliably annotated and whether they could capture salient features of writing improvement. We analyzed whether two annotators could reach a good agreement score using the defined schema. The corpora with expert gradings were used to analyze the correlation between revision and writing improvement. Afterwards we explored whether the schema could be useful for providing revision feedback. We designed a system that provides the revision type information as the feedback (Zhang et al., 2016a), and then conducted a user study to evaluate whether the

[1] http://argrewrite.cs.pitt.edu

feedback can influence the participant's revision behaviors (Zhang et al., 2017).

For the automatic identification of revisions, we first explored the problem in a pipeline manner. We first introduce an algorithm for the first part of the task: identification of revision location (revision extraction). We treat the problem as an automatic sentence alignment problem (Zhang and Litman, 2014). Then we introduce our efforts for the second part of the task: identification of revision types (revision classification). We introduce three approaches developed for the problem: 1) Utilizing the features and method proposed in existing works (Zhang and Litman, 2015) 2) Utilizing the contextual information to improve the performance. (Zhang and Litman, 2016) and 3) Utilizing the discourse information to improve the performance (Zhang et al., 2016b; Forbes-Riley et al., 2016).

Afterwards we explored on the joint identification of revision location and revision type. The approach involves combining both types of information into one prediction sequence and improves the sequence likelihood using evolutionary computing techniques (Zhang and Litman, 2017).

1.6 MAIN CONTRIBUTIONS

The results in this thesis contribute to both education and NLP researches.

For the education research community.

- Develops an argumentative revision schema that can be reliably annotated by human. The schema captures salient characteristics of writing improvement.
- Develops a prototype intelligent rewriting assistance tool for rewriting tutoring.
- Conducts a user study on the impact of revision feedback on the users' writings.

For the NLP research community.

- Proposes to use contextual information to improve the performance of revision classification.
- Proposes a novel way to utilize the discourse information for revision classification study.

- Proposes a joint model that combines the identification of revision location and revision type classification together.
- Collects a publicly available corpus for possible revision researches.

1.7 OUTLINE

This chapter introduces the motivations and challenges of this thesis. The rest of this thesis is organized as follows: Chapter 2 introduces the annotation schema for argumentative writing revisions. Chapter 3 introduces data collected based on the schema and the results of the revision study on the correlation between revision and writing improvement. Chapter 4 introduces a prototype intelligent revision assistant ArgRewrite and a user study to examine whether feedback based on the schema is helpful. Chapter 5 presents the algorithms for revision location identification and revision type classification separately. Chapter 6 further presents a joint approach that identifies the location and the revision type together at the same time. Finally, Chapter 7 and Chapter 8 summarizes all the works and presents the possible future directions of our work.

2.0 REVISION SCHEMA DEFINITION

Data-driven development of a rewriting assistance tool requires the definition of a schema for the annotation and classification of revisions. The definition should be clear enough for humans to distinguish. Also, the categorization of revisions should capture the difference of their impacts on writing improvement. This chapter introduces our efforts in the design of a revision schema for argumentative writings. Portions of this work were originally published in (Zhang and Litman, 2014, 2015). In the end we introduce our efforts in generalizing our framework for another writing genre (science report).

2.1 ARGUMENTATIVE WRITING REVISION SCHEMA AND REVISION ANNOTATION

2.1.1 Related Work

Faigley and Witte (1981) categorized revisions using two categories: surface change and text-base (content) change. Bronner and Monz (2012) chose a similar categorization (factual vs. fluency) for Wikipedia revisions. They both classified revisions according to whether they change the information of the text or not. The following researchers typically reuse the categorization as the coarse level revision categorization in their own schema (Cho and MacArthur, 2010; Early and Saidy, 2014; Daxenberger and Gurevych, 2012). Besides the coarse categorization of revisions, specific revision categories are typically defined according to the researchers' task. Pfeil et al. (2006) defined revision categories according to the revision action (add information, reversion, vandalism, etc.) in the effort of finding differ-

ences in collaboration between different cultures. Jones (2008) designed revision categories according to the characteristics of the Wikipedia dataset (Wikipedia policy violation, add image, add link, etc.) for the analysis of the revision pattern in Wikipedia. Daxenberger and Gurevych (2012) chose similar categories when they analyze the difference of edit categories between featured articles and not featured articles and listed them as sub categories of Faigley and Witte's definition. Early and Saidy (2014) followed the coarse definition of Faigley and Witte and further categorized revisions to *Main Idea*, *Developing Argument*, *Textual Evidence*, *Rhetorical Strategies*, and *Language Choice* for the analysis of the students' revision strategies. Our schema is similar to Early and Saidy's work, which defines sub-categories under Faigley and Witte's definition for the study of student writings. Their work focus on describing what kind of revision strategies are used by students; our work attempts to describe the details of each individual revision (the location, the operation and the purpose). We focus the design of our schema on argumentative writings first and then investigate whether we can apply the schema to scientific writings with minor adaptation.

In Bridwell's study of students' revising strategies, revisions were studied at 6 different levels, including Surface level, Lexical level, Phrase level, Clause level, Sentence level and Multi-sentence level (Bridwell, 1980). These six levels were used to categorize revisions in Bridwell's work. In other research, typically 1 or 2 of the levels was chosen as the granularity for revision description. Bronner and Monz (2012) first extracted the word-level diff segments between different versions of Wikipedia drafts and then extracted the user edits as minimal sets of sentences overlapping with deleted or inserted segments. Iida and Tokunaga (2014) built a corpus of manually revised texts from the discourse perspective. A discourse parser was applied to segment text into discourse units and the author's changes to the ordering and the connectives of units were annotated. The addition and deletion of sentences were not addressed in their paper. Lee et al. (2015) developed a large-scale corpus containing drafts and final versions of essays written by non-native speakers. The corpus includes the tutor's comments on the issues of the students' paper and the alignment of the students' drafts at the sentence level and the word level. They focused on the issues of the students' essays but did not explicitly annotate how each of the changes relates to the student's issues. Daxenberger and Gurevych (2012) extracted revisions by aligning sentences first and

extracting character-level edits on minor-changed sentences. A similar approach was chosen in our work, we extracted revisions at the sentence level by aligning the sentences across drafts and assign the revision purpose for each of the aligned sentence pair. However, we did not further extract revisions at word/character level as we found the sentence-level to be the most appropriate for our task.

2.1.2 Argumentative Revision Schema Definition

As shown in Table 2.1, two dimensions are considered in the definition of the schema: the author's behavior (**revision operation**) and the purpose of the author's behavior (**revision purpose**, i.e. the aspect the author aiming to improve).

Revision operations include three categories: *Add*, *Delete*, *Modify*. The operations are decided automatically after sentences get aligned. As in the Example 2 of Table 2.2, Sentence 2 in Draft 1 is aligned to null, the revision operation is *Delete*; Sentence 2 and 3 in Draft 2 are aligned to null, their operations are both *Add*.

Following the definition of Faigley and Witte (Faigley and Witte, 1981), revisions are categorized to two major categories: surface and meaning (content) changes. For surface changes, similar to Faigley and Witte's classification of *Format changes* and *Meaning preservation changes*, we define the category *Conventions/Grammar/Spelling* to describe the revisions for correcting convention/language errors and *Word usage/Clarity* to describe the revisions for improving text fluency. We also add a category *Organization* to describe the author's change to the structure of the text (e.g. merging sentences together).

For meaning changes, we first define four categories according to Toumin's model of argumentation (Toulmin, 2003): *Claims/Ideas*, *Warrant/Reasoning/Backing*, *Rebuttal/Reservation*, *Evidence* [1]. Inspired by the category *Introductory material* defined by Burstein et al. (Burstein et al., 2003) in essay-based discourse categorization, we introduce a category *General Content* for text that serves as introductory materials or summaries.

In Table 2.2 we provide examples and explanations for the revision purpose categories.

[1] Corresponds to *Claim*, *Warrant*, *Backing*, *Rebuttal*, *Grounds* defined in Toulmin's model

Purpose Category	Operation	Purpose Definition
Content		
Claims/ Ideas	Add/Delete/Modify	change of the position or claim being argued for
Warrant/ Reasoning/ Backing	Add/Delete/Modify	change of principle or reasoning of the claim
Evidence	Add/Delete/Modify	change of facts, theorems or citations for supporting claims/ideas
Rebuttal/ Reservation	Add/Delete/Modify	change of development of content that rebut current claim/ideas
General Content	Add/Delete/Modify	change of content that do not directly support or rebut claims/ideas
Surface		
Word Usage/ Clarity	Modify	change of words or phrases for better representation of ideas
Conventions/ Grammar /Spelling	Modify	changes to fix spelling or grammar errors, mis-usage of punctuation or to follow the organizational conventions of academic writing
Organization	Modify	changes to the structure/organization of the text

Table 2.1: Revision schema definition

2.1.3 Deciding the Granularity for Revision Extraction

Changes between drafts need to be extracted first before further annotations. We compared the extraction of revisions at three levels: word-level, clause-level and sentence-level.

Ideally word-level revision extraction grabs more details than other two levels. In practice, however, while word-level comparison can detect small changes between single sentences, it does not work well on heavy edits. As in Figure 2.1, the sentence in Draft 1 will be aligned to all sentences in Draft 2, which creates difficulty for more precise revision categorization.

Segmenting text to clauses offers the benefit of more precisely described revisions. As the example in Figure 2.1, changes between the 1st sentences would be more precisely extracted as the *Claim/Idea* change from "the unbaptized and those before Christ dwell" to "the unbaptized" and the *General Content* addition of "live in complete darkness". However, there are two major drawbacks for the clause-level revision annotation. First, while theoretically the clauses-level annotation would cover more details, the current automatic clause segmentation techniques are still not reliable enough for our purpose yet. The state-of-art discourse parser (Feng and Hirst, 2014) (which typically segments text into clauses) reports 92.8 precision and 92.3 recall on RST Discourse Treebank. The results are expected to be worse on our dataset where the essays are less formal as they are written by students. Extra work needs to be done to fix clause segmentation. Second, while the clause-level annotation describes changes between one sentence more precisely (i.e. the *Modify* operations), it is overkill for the description of the addition/deletion of a whole block of text (i.e. the *Add* and *Delete* operations). In the example of Figure 2.1, annotators have to label the revision purpose for both "They are not punished" and "because they did not know Christ", which increases the workload for the annotators. We also found that annotators often got confused about whether they should assign the same or different labels for different clauses of the same sentence during the annotation process.

Teacher's Prompt	Now that we are seven cantos and five levels into Hell; you should be able to correlate sinners and punishments that Dante feels appropriate. Your task is to construct a well written, concise essay placing contemporaries into each level and specifically justify why each modern-day person appropriately fits...at least according to your thought process. Be certain to cite evidence from the text as needed!
Ex.1	Draft 1: (1, "Finally **are** the wrathful, here are the people who are full of hate") Draft 2: (1, "Finally **is** the wrathful."), (2, "**Here** are the people who are full of hate")
Rev	(1->1,2, Modify, "Organization"), (1->1,2, Modify, "conventions/grammar/spelling")
Reason	The sentence in Draft 1 is split to two sentences in Draft 2, this change of text structure is labeled as an "Organization" change. The modification of "are" to "is" in Draft 2 is an attempt to fix grammar mistakes and thus labeled as "Conventions/Grammar/Spelling". Note that we label the purpose of a revision only according to the author's purpose no matter the revision really improves the paper or not.
Ex.2	Draft 1: (1, "In this circle I would place Fidel."), (2, "He was a ruthless dictator.") Draft 2: (1, "In the circle I would place Fidel"), (2, "**He was annoyed with the existence of the United States and used his army to force them out of his country**"), (3, "**Although Fidel claimed that this is for his peoples' interest, it could not change the fact that he is a wrathful person.**")
Rev	(2->null, "Delete", "Warrant/Reasoning/Backing" (null->2, "Add", "Evidence"), (null->3, "Add", "Rebuttal/Reservation")
Reason	Sentence 1 of Draft 1 is aligned to the first sentence 2 of Draft 2, there is no change between the aligned sentences. Sentence 2 of Draft 1 is the author's reasoning of why Fidel should be put into the wrathful level. The author deleted the sentence and added Fidel's behavior as the support his claim. In sentence 3 the author added a rebuttal for the reasoning that Fidel's behavior is for his peoples' interest.
Ex.3	Draft 1: none

	Draft 2: (1, Before Dante actually enters H. he has to go through the Vestibule.), (2, In this level the people get stung by hornets and bees but they bleed because of all of the stings.), (3, I think for this level I would place Mrs. X , the band director for X high school.)
Rev	(null->1, "General Content"), (null->2, "General Content"), (null->3, "Claims/Ideas")
Reason	Sentence 1,2 are added to introduce the author's claim of who should be in the level of Vestibule. However, these two sentences are not directly reasoning or supporting the author's claim, thus categorized as "General Content". As the prompt is requesting students to put contemporaries into different levels of Hell. Sentence 3 is the author's statement of which person being put into Vestibule.
Ex.4	Draft 1: (1, Saddam Hussein and Adolf Hitler belong to this level.), (2, They both killed many people when they were in their position.), (3, Many people were killed for no reason.) Draft 2: (1, Fidel Castro belongs to this level.), (2, He killed many people when he was in his position.), (3, Many people were executed for no reason.)
Rev	(1->1, "Claims/Ideas"), (2-2, "Word Usage/Clarity"), (3->3, "Word Usage/Clarity")
Reason	The author changed the claim of the person belonging to the wrathful level in the modification of Sentence 1. The revision of Sentence 2 is a "Word Usage/Clarity" change as the main content is the same, but "both" is removed and "were" is changed to to "was". The revision of Sentence 3 is the replacement of "killed" to "executed", which is a regular "Word Usage/Clarity" change.

Table 2.2: Examples of different revision purposes.

Comparing to word-level revision extraction, alignment of text between drafts is more accurate at the sentence level as the semantic similarity is considered during alignment annotation. Meanwhile, segmenting text according to the natural boundary of sentences rather than clauses reduces the workload of the annotators and the difficulty of automatic

revision extraction. While it has the limitation that sometimes two types of changes can be involved in one sentence (as in Figure 2.1), this case happen infrequently in our dataset and we can still get around the problem by allowing annotators to annotate multiple revision purposes to one alignment.

In summary, we argue that the granularity of the revision extraction should be carefully chosen for different purposes. Word/Character level works best for the task of identifying surface changes (e.g. grammar error detection). Clause level can describe sentence modifications more accurately if a reliable clause segmentation tool is available. The workload of clause-level annotation would increase if there is a higher percentage of *Add* or *Delete* edits. Sentence level is more robust comparing to the other levels for annotation. For our task, we decide to annotate revisions at the sentence level, which allows annotators to easily align the text between drafts and annotate the revision purpose.

2.2 DATA ANNOTATION

2.2.1 The Revision Annotation Process

The complete revision annotation process involves 3 stages: text preprocessing, sentence alignment and revision purpose labeling.

The documents are segmented into sentences using Stanford Document Preprocessor (Manning et al., 2014) before annotation. After segmentation, for each draft, the N sentences in the draft are assigned indexes from 1 to N according to the sequence of their occurrence in text. The pre-processed results are stored in spreadsheet files, each draft corresponds to a single sheet.

In the alignment stage, the annotators decide the alignment of sentences according to whether the sentences are semantically similar to each other. Each sentence in the revised draft is assigned the index of its aligned sentence in the original draft. If a sentence is added, it will be annotated as *Add*. The sentence alignments can be one-to-one, one-to-many and many-to-one. If a sentence in the revised draft is aligned to multiple sentences in the original

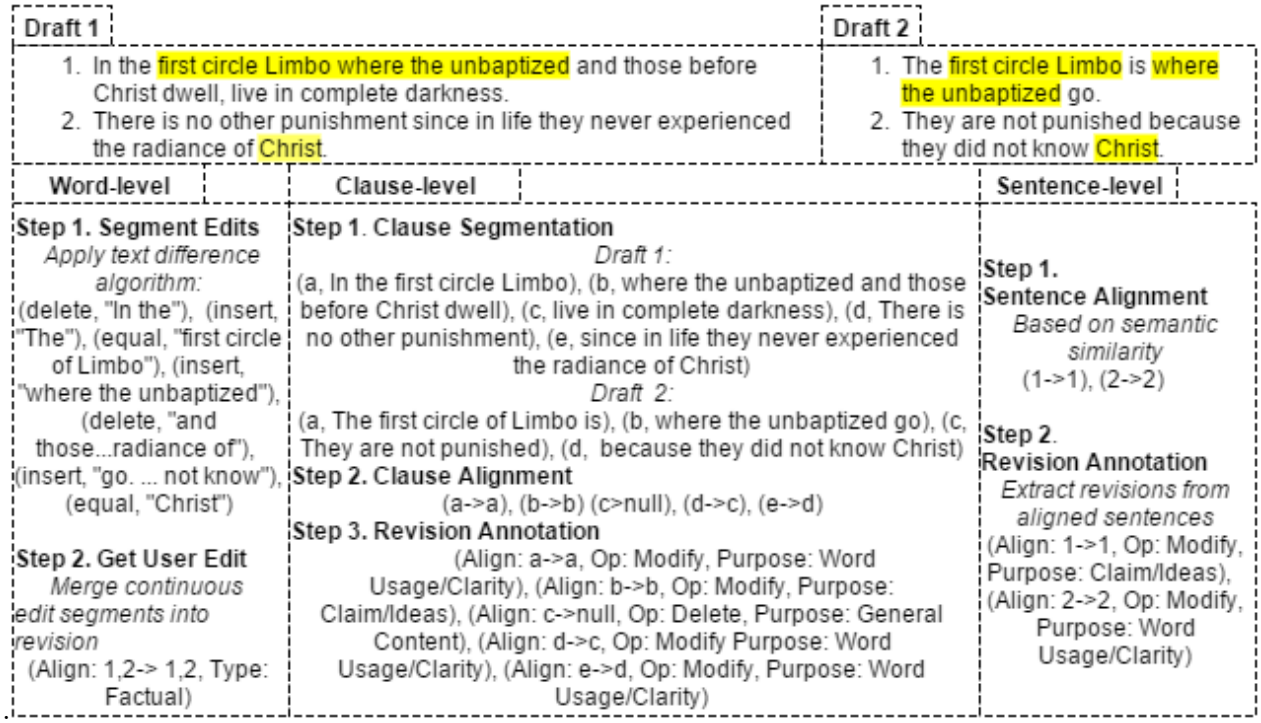


Figure 2.1: In the example, the author removed "those before Christ dwell" from the claim. The language is also made simpler and more accurate in the second draft. In the word-level revision annotation, we adapted methods in prior works (Bronner and Monz, 2012) and extracted changes based on the results of a text-diff algorithm (https://code.google.com/p/google-diff-match-patch/). In the clause level, the sentences were first segmented into clauses. In the clause/sentence level annotation, clauses/sentences were first aligned and revisions were then annotated on the aligned pairs.

draft, the annotator should mark all the aligned sentences' indexes. If multiple sentences in the revised draft are aligned to one sentence in the original draft, the annotator should mark the aligned sentence's index for every sentence in the revised draft. We also specified that one-to-many and many-to-one annotations are only allowed cases where the similarity between the aligned pairs are explicit (e.g. a sentence with two clauses is broken to two consecutive sentences).

The annotators are required to annotate on the aligned sentences (including the *Add* and *Delete* pairs). In the annotation of our first data set, each aligned pair can have one to many revision purposes. In the other annotations, annotators can label multiple surfaces changes but at most 1 content change for each aligned pair [2]. Specific rules are made for ambiguous cases. In this section the most important ones are listed (more details can be seen in the Appendix A). The annotator can only annotate multiple revision purposes to one aligned pair if they cannot differentiate the purposes based on the rules. Figure 2.2 shows a screenshot of our annotation tool.

Conventions/Grammar/Spelling* vs. *Word Usage/Clarity These two genres are similar as they don't change the content of the text and improve the quality of the text. The annotators annotates a change as the former one only when there are convention/spelling/grammar mistakes being addressed.

Evidence* vs. *Warrant/Reasoning/Backing These two categories are similar as they both provide support to the authors' claim. The annotators are required to distinguish these two categories according to whether the sentences are stating facts. The facts include (1) Citation: the citation of papers, reports, news and books. (2) Example: facts of history or personal experiences. (3) Scientific proof. Revisions involving facts would be marked as Evidence, otherwise would be marked as Warrant.

Claim/Ideas* vs. *Warrant/Reasoning/Backing One paper typically contains a major claim and several sub-claims to support the major claims. These sub-claims are often also used as the reasoning to support the major claim. To distinguish sub-claim and reasoning, a sentence is annotated as *Claim/Ideas* if it is further supported or objected by

[2]The annotators of the first data set were required to select one major content revision if there are multiple content revision purposes labeled.

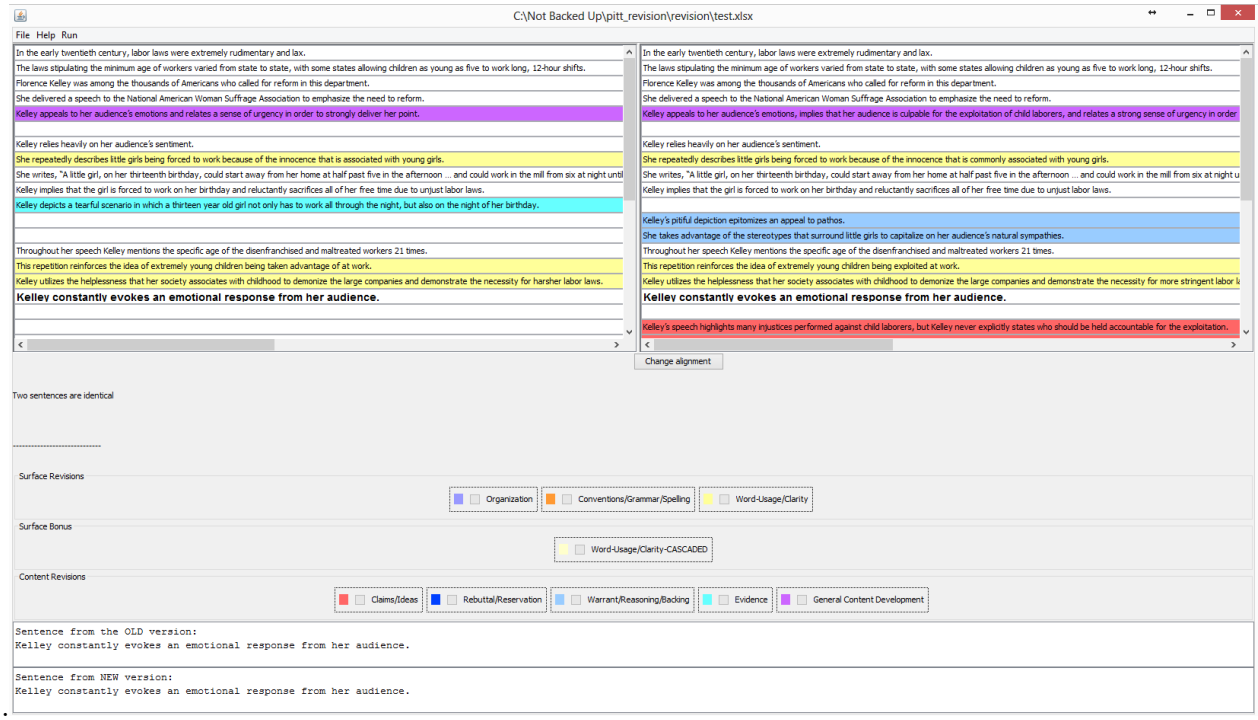


Figure 2.2: A screenshot of the annotation tool

other sentences. Otherwise it is marked as "Warrant/Reasoning/Backing".

General Content* vs. *Warrant/Reasoning/Backing A revision is annotated as *Warrant/Reasoning/Backing* only when it is directly reasoning about the claim. If it is just providing hints to the claim, even if the hint is explicit and strong, it should still be marked as *General Content*.

2.3 GENERALIZING THE REVISION FRAMEWORK

After observing the character of the science dataset, we choose to keep the categories of our schema while adding a new category "Precision" for the case where the author make changes to make his statement or argument more precise. For example, the modification from "The bacteriorhodopsin absorbed a high majority of the wavelengths. " to "The bacteriorhodopsin used by aquatic organisms at this time absorbed a high amount of green light like they do now." would be a "Precision" revision, where the author added restrictions to make the statement more precise.

2.4 SUMMARY

In this chapter I describe our efforts in the definition of an argumentative revision schema. I first compared different levels of granularity for describing revisions and the sentence level is determined to be the best option. Using the natural sentence boundary, we avoid the possible segmentation errors of machine learning algorithms. Also, describing revision at the sentence level allows the consideration of semantic information in text alignment, thus making it easier for humans to decide the revision categories. Then we described our revision schema and the annotation process based on the schema. A preliminary attempt on the extension of our schema to other genres of writings is also conducted.

In Chapter 3 we report our analysis on whether the revision schema defined can be reliably annotated by human. In Chapter 4 we report our analysis on whether revision

feedback based on our schema can influence users' revision behaviors.

3.0 REVISION SCHEMA APPLICATION ANALYSIS (SWORD)

This chapter describes the application of the revision schema defined in Chapter 2. This chapter investigates two major questions: 1) Can revisions be reliably annotated under the revision schema? 2) Can the defined schema capture salient features of writing improvement? We first describe the corpora used in our study and then report the results of the agreement study for the first question. Afterwards we report the data analysis results on the annotated corpora for the second question.

3.1 DATA DESCRIPTION

Corpora used in this chapter were collected from SWoRD (Cho and Schunn, 2007). The data consists of the first draft (Draft1) and second draft (Draft2) of papers written by high school or undergraduate students; students were required to provide feedback on other students' writings on specified aspects (Thesis, Writing, Evidence, etc.); papers were revised after receiving and generating peer feedback.

Corpora *Align1* and *Align2* were written by undergraduate students in a course "Social Implications of Computing Technology". In corpus *Align1*, the students discussed the role that Big Data played in Obama's presidential campaign. This corpora contains 11 pairs of the first and final drafts of short papers. Corpus *Align2* talks about intellectual property and contains 10 pairs of paper drafts. We used these two corpora for our sentence alignment study in Chapter 5.1.

Corpora *HSchool1* and *HSchool2* were written by high school students taking English writing courses. *HSchool1* contains papers about Dante's Inferno and contains drafts from 47

students. After data was collected, a score from 0 to 5 was assigned to each draft by experts (for research prior to our study). The score was based on the student's performance including whether the student stated the ideas clearly, had a clear paper organization, provided good evidence, chose the correct wording and followed writing conventions. The class's average score improved from 3.17 to 3.74 after revision. In *HSchool2*, students were required to explain the rhetorical strategies used by the speaker/author of a previously read lecture (Topic 1)/essay (Topic 2). As in the previous corpus, students wrote Draft 1 and Draft 2 essays on one of the two topics. Afterwards the students wrote another draft (Essay2) on the other topic. Like HSchool1, the Draft1 essays were graded by experts (but at a scale from 0 to 6). Essays were scored on the quality of thesis, rhetorical strategies, textual evidence, explanations, organization, writing style and standard English. Draft2 essays were not graded. Instead, Essay2 were graded to study whether there exists transfer learning effect (Students will learn from previous revisions and do better in writing new essays). The average of Draft1 scores is 4.59 and the average of Essay2 score is 4.51. These corpora were used in our revision analysis in Chapter 2 and automatic revision identification study in Chapter 5 and Chapter 6.

Science reports written by high school students were collected as Corpus *Science*. 9 pairs of reports have been annotated so far, we utilize the science reports to study the generalization of our revision schema.

3.2 DESCRIPTIVE STATISTICS

Table 3.1 summarizes some basic characteristics of the corpora collected.

3.3 AGREEMENT STUDY

As stated in Chapter 1, annotators did a study on sentence alignment first on Corpus *Align1*. Two pairs of drafts were separately annotated by two annotators and the agreement is 98.77%

Corpus	Writers	Size	D1Num	D2Num	Description
Align1	Undergraduate	11	23	23	Used in revision extraction study in Section 5.1.2 in Chapter 5
Align2	Undergraduate	10	25	25	
HSchool1	High school (English)	47	38	53	Draft1 and Draft2 graded by experts, used in revision study in Chapter 3 and revision classification in Chapter 5 and Chapter 6
HSchool2	High school (English)	63	26	29	Draft1 in Essay1 and Essay2 graded by experts. Used in revision study in Chapter 3 and revision classification in Chapter 5 and Chapter 6 .
Science	High school (Science)	9	13	17	Used in extended revision study in section 2.3 in Chapter 3

Table 3.1: Corpora collected via SWoRD, size indicates the number of essay pairs, D1Num indicates the average number of sentences in Draft1, D2Num indicates the average number of sentences in Draft2

[1], which indicates that humans can align the sentences reliably. *HSchool1*, *HSchool2* were annotated by 2 annotators each. One of the annotators who participated in the study of sentence alignment did the alignment of sentences of *HSchool1*. The other annotator checked the alignment before labeling revision purposes. Through this process, we make sure both annotators read the essays before annotation. Annotators first practiced annotations on two files and discussed with each other on disagreements. Afterwards another 5 files were double-coded to examine the agreement. The Kappa of *HSchool1*[2] is 0.75. In the annotation of *HSchool2*, annotators practiced sentence alignment and revision purpose annotation on two files from *HSchool1* first and then aligned sentences on 5 files of *HSchool2*, the agreement of sentence alignment is 98.43%. After reaching agreement on the sentence alignments, annotators annotated revision purposes on the 5 files. The agreement Kappa is 0.69.

One annotator who participated in the annotations of Corpora *HSchool1* and *HSchool2* and one annotator who participated in the annotation of Corpus *HSchool2* double-coded 9 files on Corpus *Science* with the adapted schema. The annotators report no difficulty in the adaption of the schema. Two annotators reach 100% accuracy in sentence alignment and 0.95 Kappa in revision purpose annotation, which indicate that annotators can annotate revisions in other types of writings under an adapted version of our revision schema. Details of the annotated data are in Table 3.2.

Number of revisions
Hschool1 (47), total: 1273

[1] $Agreement = \frac{\#AgreedAlignedSentencesDraft1+\#AgreedAlignedSentenceDraft2}{\#Draft1Sentences+\#Draft2Sentences}$, adapted from (Raghava et al., 2003)

[2]Each change have 5 categories of revisions: *Claim*, *Warrant*, *Evidence*, *General Content*, *Surface*. In our schema, each aligned pair can have multiple surface purposes. we merged the surface change categories into one single *Surface* category as we need only one label on an aligned pair for calculation and it's easy for annotators to distinguish different surface changes. A pair is considered to be a surface change only if it does not have the labeling of content changes. *Rebuttal/Reservation* is not included as it only occurred once.

Purpose	#Add	#Delete	#Modify
Total	797	95	381
Surface	0	0	309
Organization	0	0	45
Conventions	0	0	84
Word-usage	0	0	180
Content	797	95	72
Claim	79	23	8
Warrant	335	40	15
Rebuttal	1	0	0
Evidence	95	10	5
General	287	22	44

HSchool2 (63), total: 1054

Purpose	#Add	#Delete	#Modify
Total	344	152	558
Surface	0	0	401
Organization	0	0	9
Conventions	0	0	109
Word-usage	0	0	283
Content	344	152	157
Claim	27	12	37
Warrant	188	82	57
Rebuttal	0	0	0
Evidence	13	5	16
General	116	53	47

Science (9), total: 116

Purpose	#Add	#Delete	#Modify
Total	49	14	53
Surface	0	0	31
Organization	0	0	5
Conventions	0	0	3
Word-usage	0	0	23
Content	49	14	22
Claim	3	2	9
Warrant	36	12	9
Rebuttal	0	0	0
Evidence	0	0	0
General	10	0	1
Precision	0	0	3

Table 3.2: Distribution of revisions in the corpora collected via SWoRD

3.4 REVISION ANALYSIS

Two studies were conducted to demonstrate the utility of the schema. We first conducted a corpus study analyzing relations between the number of each revision type in our schema and student writing improvement based on the expert paper scores available for *HSchool1*. In particular, the number of revisions of different categories are counted for each student. Pearson correlation between the number of revisions and the students' Draft 2 scores is calculated. Given that the students' Draft 1 and Draft 2 scores are significantly correlated ($p < 0.001$, $R = 0.63$), we controlled for the effect of Draft 1 score by regressing it out of the correlation.[3] We expect surface changes to have smaller impact than content changes

[3]Such partial correlations are one common way to measure learning *gain* in the tutoring literature, e.g. (Baker et al., 2004).

Revision Purpose (N = 1272)	**R**	**p**
Surface	0.10	0.510
Organization	0.09	0.554
Conventions	-0.07	0.627
Word-usage	0.15	0.333
Content	**0.53**	**<0.001**
Claim	**0.47**	**0.001**
Warrant	**0.45**	**0.002**
Evidence	**0.41**	**0.004**
General Content	0.24	0.116

Table 3.3: *HSchool1* Study. Partial correlation between Draft 2 score and the number of revisions (control draft 1 score out).Rebuttal/Reservation is not included because of rare occurrence

as Faigley and Witte (1981) found that advanced writers make more content (text-based) changes comparing to inexperienced writers.

Data analysis on the effectiveness of revision feedback on users' writings was conducted on the *ArgRewrite* corpus, details of the study will be described in Chapter 4.

We followed similar approaches for *HSchool2* and analyzed relations between the number of each revision type and student writing skill improvement. After testing that the students' Essay 1 and Essay 2 scores are significantly correlated ($p < 0.001$, $R = 0.54$), Pearson correlation between the number of revisions and the students' Essay 2 score is calculated controlling for the effect of Essay 1 score. We expect the observation of a transfer learning effect, where students making more changes would have a higher improvement in the writing of a new paper on a different topic.

The results of *HSchool1* study are shown in Table 3.3, Pearson correlation between the number of revisions and the students' Draft 2 scores was calculated. Results show that only

Revision Purpose (N = 1045)	**R**	**p**
Surface	-0.02	0.890
Conventions	-0.02	0.880
Word-usage	-0.01	0.968
Content	0.24	0.057
Claim	**0.35**	**0.005**
Warrant	**0.40**	**0.001**
Evidence	0.20	0.122
General Content	-0.13	0.302

Table 3.4: *HSchool2* Study. Partial correlation between Essay2 score and the number of revisions (control Essay1 score out).Rebuttal/Reservation, Organization are not included because of rare occurrence

the number of content revisions is significantly correlated ($R = 0.53, p < 0.001$). Within the content revisions, only *Claims/Ideas*, *Warrant/Reasoning/Backing* and *Evidence* are significantly correlated. Table 3.4 demonstrates that only the number of *Claim/Ideas* and *Warrant/Reasoning/Backing* revisions are significantly correlated with the writing improvement in a different topic.

Through the results, we do find significant correlations between the author's rewriting effort and writing improvement. We also observed a transfer effect that the authors' effort in revising one essay would improve their performance in the writing of a new essay. The findings also demonstrate that different categories of revisions have different relationships to students' writing success, which suggests that our schema is capturing salient characteristics of writing improvement.

3.5 SUMMARY

In this chapter, we first introduce the results of agreement study on data annotation using our schema. The high agreement in annotation indicates that our schema can be annotated reliably by humans, which suggests the correctness of hypothesis **H1.1**. Data analysis on the annotated corpora suggests the correctness of hypothesis **H1.2**, showing that the number of revisions is significant correlated with both the improvement of essay quality within one paper and also the author's skill improvement in the writing of a new paper. The study results also suggest that different revision categories have different correlations with the writing improvement, which indicates that our schema captures the salient features of writing improvement.

4.0 REVISION SCHEMA APPLICATION ANALYSIS (ARGREWRITE)

Proving that the schema captures salient features of improvement, we further investigate whether the schema-based application can have an impact on the user's rewriting behaviors. To investigate the effects of providing schema-based revision feedback, a revision assistance tool was developed and a user study was conducted on the application of the tool. This chapter describes our work in (Zhang et al., 2016a, 2017).

4.1 THE BUILDING OF A PROTOTYPE INTELLIGENT REVISION ASSISTANT: ARGREWRITE

We argue that an intelligent writing assistant ought to be aware of the revision process; it should: 1) identify all significant changes made by a writer between the essay drafts, 2) automatically determine the purposes of these changes, 3) provide the writer the means to compare between drafts in an easy to understand visualization, and 4) support instructor monitoring and corrections in the revision process as well.

In this chapter we assume 1) and 2) has been resolved and develops a web-based interface to support student argumentative writings. The purpose of each change between revisions is demonstrated to the writer as a kind of feedback. If the author's revision purpose is not correctly recognized, it indicates that the effect of the writer's change might have not met the writer's expectation, which suggests that the writer should revise their revisions.

4.1.1 System Overview

The design of ArgRewrite aims to encourage students to concentrate on revision improvement: to iteratively refine the essay based on the feedback of the automatic system or the writing instructor. Our framework consists of two major components, arranged in a server client model. On the server side, the **automatic analysis** component extracts revision changes by aligning sentences across drafts and infers the purposes of the extracted revisions. On the client side, a web-based **rewriting assistant interface**[1] allows the student to retrieve the feedback to their revisions from the server, make changes to the essay and submit the modified essay to the server for another round of analysis.

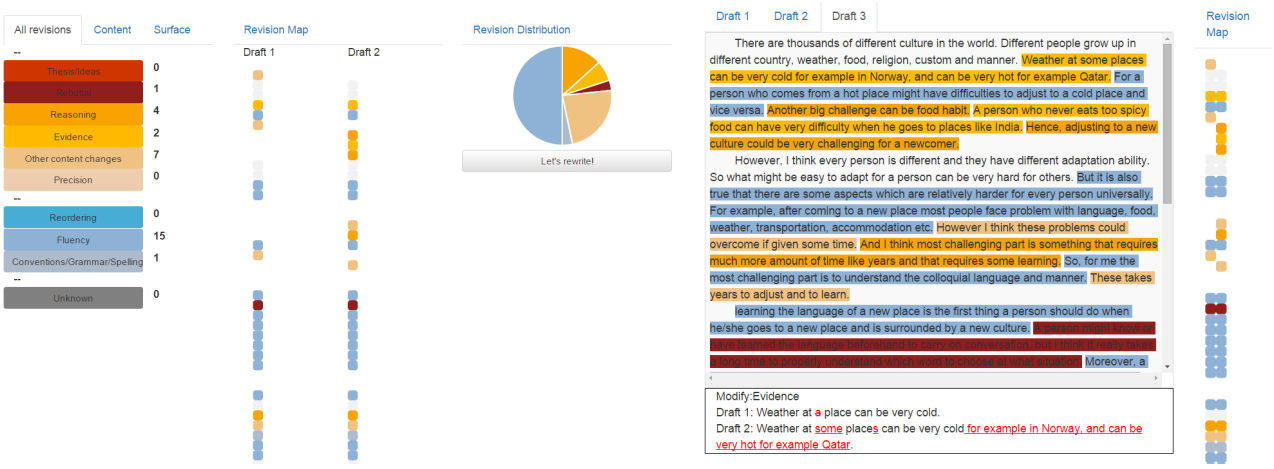
The complete process of the student's writing using our system starts with the student's rewriting and submission of the essay. The student writes the first draft of the essay before using our system and then modifies the original draft in our rewriting assistant interface. The submitted writings are automatically analyzed immediately after the receipt of the student's submission. After receiving the analysis feedback, the student can choose to continue with the cycle of essay revising until the revisions are satisfactory.

4.1.2 Rewriting Assistance Interface Design

In this chapter we focus on the discussion of the rewriting assistance interface design and leave the discussion of the **automatic analysis** to Chapter 5 and Chapter 6.

Our rewriting assistant interface is designed with several principles in mind. 1) Because the revision classification taxonomy goes beyond the binary textual versus surface distinction, we want to make sure that users don't get lost distinguishing different categories; 2) We want to encourage users to think about their revisions holistically, not always just focusing on low-level details; 3) We want to encourage users to continuously re-evaluate whether they succeeded in making changes between drafts (rather than focusing on generating new contents). Thus, we have designed an interface that offers multiple views of the revision changes. As demonstrated in Figure 4.1, the interface includes a *revision overview* interface

[1]rewriting assistant interface: http://argrewrite.cs.pitt.edu/demo.html now supported on chrome and firefox browser only



(a) revision overview interface

(b) revision detail interface

Figure 4.1: Screenshot of the web interface, which includes (a) the *revision overview* interface with the *revision statistics* (the numbers indicate the numbers of specified revision purposes) region, the *revision map* region and the *revision distribution* region, (b) the *revision detail* interface with the *revision text area* region and the *revision map* region (from left to right).

for the overview of the authors' revisions and a *revision detail* interface that allows the author to access the details of their essays and revisions.

Inspired by works on learning analytics (Liu et al., 2013; Verbert et al., 2013), we design the *revision overview* interface which displays the statistics of the revisions. Following design principle #1, the revision purposes are color coded and each purpose corresponds to a specific color. In Chapter 3 we have demonstrated that only *Content* revisions are significantly correlated with the writing improvement. To inspire the writers to focus more on the important *Content* revisions, cold colors are chosen for the *Surface* revisions and warm colors are chosen for the *Content* revisions. The statistics and the pie chart provide a quantitative summary of the writer's revision efforts. For example, in Figure 4.1, the writer makes many changes on the *Fluency* (15) of sentences but makes no change on the *Thesis/Ideas* (0). To allow the users to concentrate on improving one revision type at a time, the interface allows the user to click on a single revision purpose type and view only the specified revisions.

Following our design principle #2, the *revision map* in both interfaces presents an at-a-glance visual representation of the revision. This design is inspired by (Southavilay et al., 2013). Each sentence is represented as a square in the map. The left column of the map represents the sentences in the first draft and the right column represents the sentences in the second draft. The paragraphs within one draft are segmented by blanks in the map. The aligned sentences appear in the same row. The added/deleted sentences would be aligned to blank in the map. The revision map allows a user (either an instructor or a student) to view the structure of the essay and identify the locations of all the changes at once. For example, in Figure 4.1, the user can quickly identify that the writer aims at improving the clarity and soundness of the third paragraph by making a *Rebuttal* modification on the second sentence and *Fluency* modifications on all other sentences. The user can also click on the square to view the details of the revision in the *revision text area* region of the *revision detail interface*.

To encourage students to make revisions (design principle #3), in the *revision detail* interface the *revision text area* region highlights the revisions (color-coded by the revision categories) in the essay and allows the writer to modify it directly. The writer clicks on the text to read the revision and examine whether the revision purpose is recognized by the

instructor/system. A character-level diff[2] is done on the aligned sentences to help the writer identify the differences between two drafts. In the example the writer can see that their "Evidence" change is recognized, indicating that the revision effort is clear and effective. If the writer finds out that their real revision purpose is not recognized, they can modify the essay in the textbox directly and submit the essay to the server when all the edits are done.

4.2 ARGREWRITE USER STUDY

Based on the developed assistant, we conducted a user study to investigate whether the schema-based revision feedback has an impact on the users' revision behaviors.

4.2.1 Hypotheses

We design our user study experiment following the two hypotheses proposed in Chapter 1.

For **H1.3**. To investigate whether there is a difference in participants' revising behaviors when different aspects of the revision schema is used to provide feedback, we decided to split the users to two groups using different revision feedback and observe the difference between the two groups. The experiment group utilizes ArgRewrite as the interface for their feedback, where revisions are colored according to their types and highlighted. The control group will only have their revisions listed and no additional information is given.

For **H1.4**. To investigate whether the difference between the recognized revision type and the users' own recognition can motivate the users' changes, we require the participants to record whether they agree with the revision type recognized by the system.

To evaluate these hypotheses, besides the collection of objective statistics such as revision numbers, we also collect the participants' subjective responses to the system.

[2]google diff match: https://code.google.com/archive/p/google-diff-match-patch/

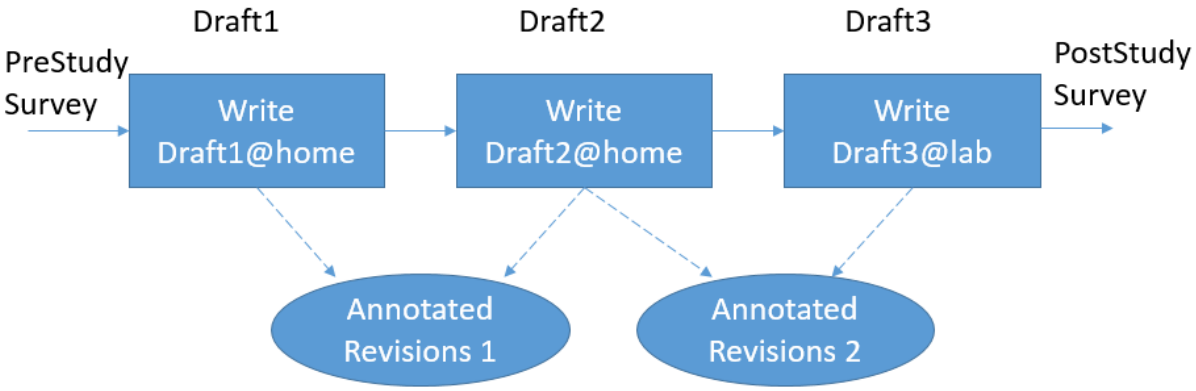


Figure 4.2: The process of the ArgRewrite study

4.2.2 User Study Experiment Procedure

We recruited 60 participants aged 18 years and older, among whom 40 were English native speakers and 20 were non-native speakers with sufficient English proficiency.[3] The study is carried out in three 40-60 minute sessions over the duration of two weeks.

Figure 4.2 demonstrates the procedure of the user study.

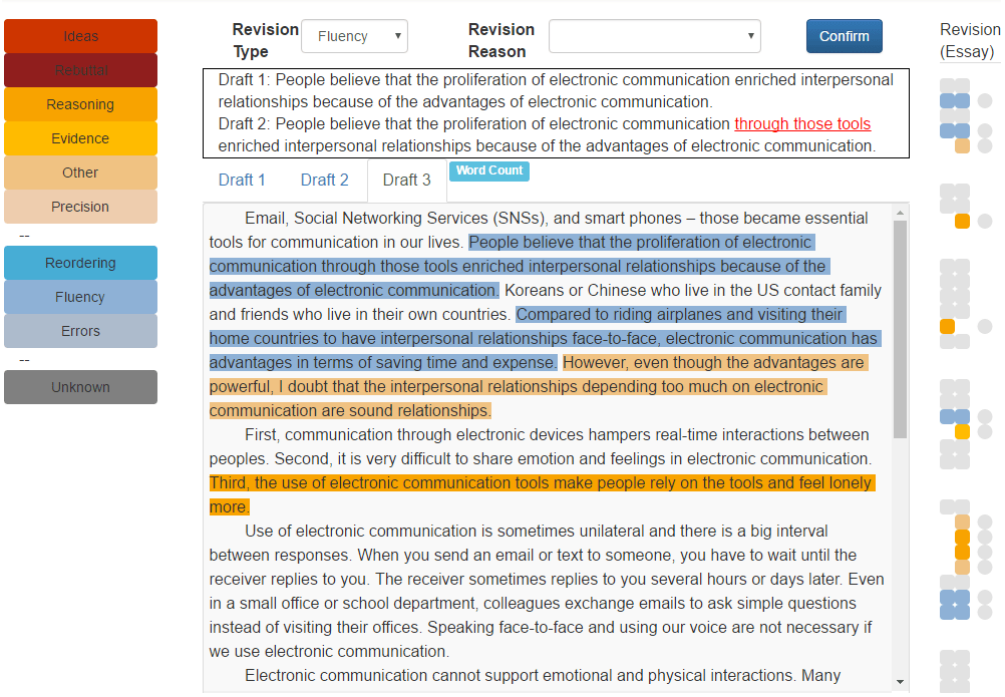
Pre-Study Survey The pre-study survey contains participant demographic information as well as their self-reported writing background, such as their confidence in their writing ability, the number of drafts they typically make, etc. Details of the question are listed in Appendix B.1.

Draft1 Each participant begins by completing a pre-study questionnaire and writing a short essay online. Participants are instructed to keep the essay around 400 words, making a single main point with two supporting examples. They are given the following prompt:

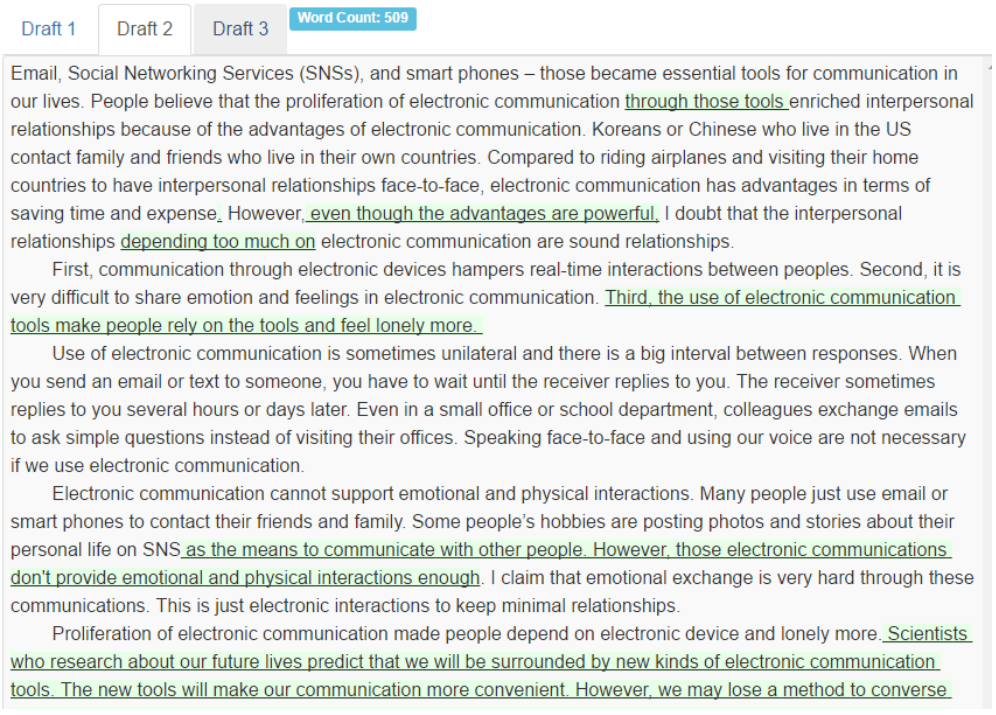
> *"Suppose you've been asked to contribute a short op-ed piece for The New York Times. Argue whether the proliferation of electronic communications (e.g., email, text or other social media) enriches or hinders the development of interpersonal relationships."*

Draft2 A few days after (typically around a week (avg: 5.88, std: 5.61)), participants

[3] i.e., with a TOEFL score higher than 100.



(a) Interface ArgRewrite.



(b) Interface Diff.

Figure 4.3: Screenshot of the interfaces. (a) *ArgRewrite (Experiment)* with the annotated revision purposes, (b) *Diff (Control)* with a streamlined character-based diff.

are asked to revise their first draft online based on the following feedback: Strengthen the essay by adding one more example or reasoning for the claim; then add a rebuttal to an opposing idea; keep the essay at 400 words.

Annotated Revisions I (Rev12) The two drafts are semi-manually aligned at the sentence level.[4] Then, the purpose of each pair of sentence revision is manually coded by a trained annotator, following the annotation guideline as in Appendix A.

Draft3 Participants perform their third revision in a lab environment. This time, they are not given additional instructional feedback. Instead, participants are shown a computer interface that highlights the differences between their first and second draft. They are asked to revise the third draft to improve the general quality of their essay. We experimented with two variations of elicitation. Chosen at random, half of the participants are shown Interface *ArgRewrite*, which highlights the annotated differences between the drafts (Figure 4.3(a)); half of the participants are shown Interface *Diff*, a streamlined character-based diff (Figure 4.3(b)). Comparing to the ArgRewrite interface published in (Zhang et al., 2016a), dropboxes are added on the top to allow the participants to report their own recognition of the revision. Word counts are added to help participants keep track of the words they have written. To remove the possible impacts of revision classification errors, the revision types are manually corrected. Both groups are asked to read a tutorial (details of the tutorials are listed in Appendix B.3) about their respective interfaces before beginning to revise. Additionally, participants in the experiment group are also asked to verify the manually annotated revision purposes between their first and second draft. After completing the final revision, all participants are given a post-study survey about their experiences. Additionally, participants in the experiment group are asked to verify the automatically predicted revision purposes between their second and third draft.

Post-Study Survey The post-study survey contains questions about the participants' in-lab revision experience, such as whether they found the computer interface helpful. Details of the survey questions are shown in Appendix B.2.

Annotated Revisions II (Rev23) Regardless of which interface the participant used,

[4]Sentences are first automatically aligned using the algorithm in (Zhang and Litman, 2014) and then manually corrected by human.

Language	Writers	Size	D1Num	D2Num	D3Num	Description
ESL	College	20	19	20	26	Used in revision identification study in Chapter 5 and Chapter 6 and feedback impact study in Chapter 4
Native	College	40	19	20	27	

Table 4.1: ArgRewrite Corpus, size indicates the number of essay pairs, D1Num indicates the average number of sentences in Draft 1, D2Num indicates the average number of sentences in Draft 2, D3Num indicates the average number of sentences in Draft 3

the second and third draft are compared and annotated by the trained annotator in the same process as before.

4.2.3 Data Annotation

Table 4.1 summarizes the descriptive characteristics of the corpus collected in the ArgRewrite user study. Similar to the annotation in Chapter 3, the *ArgRewrite* corpus was also annotated by the annotator who participated in the annotation of *HSchool2*. To check the agreement on the annotation of *ArgRewrite* corpus, two annotators annotated on 10 randomly selected essay pairs and the agreement Kappa is 0.71. Table 4.3 demonstrates the distribution of revisions in the annotated corpora.

	Content		*Surface*	
	Rev12	Rev23	Rev12	Rev23
L2 (20)	172	78	163	176
Interface ArgRewrite	91	37	71	85
Interface Diff	81	41	92	91
Native (40)	334	285	303	246
Interface ArgRewrite	177	154	149	111
Interface Diff	157	131	154	135

Table 4.2: Number of revisions, by participant groups (language, interface), coarse-grain purposes, and revision drafts (Rev12 is between Draft1-Draft2; Rev23 is between Draft2-Draft3.

Revision Purpose	**Draft1 to Draft2**			**Draft2 to Draft3**		
	#Add	#Delete	#Modify	#Add	#Delete	#Modify
Content	294	179	33	320	27	16
Claims/Ideas	25	8	4	5	0	0
Warrant/Reasoning/Backing	166	83	7	191	13	3
Rebuttal/Reservation	23	1	0	13	0	0
General Content	50	80	18	86	13	13
Evidence	30	7	4	25	1	0
Surface	0	0	466	0	0	422
Word Usage/Clarity	0	0	362	0	0	357
Conventions/Grammar/Spelling	0	0	75	0	0	52
Organization	0	0	29	0	0	13

Table 4.3: Number of revisions, by fine-grain revision purposes and edit types (add, delete, modify).

4.2.4 Data Analysis

Methodology As our study involves both ESL[5] and Native speakers, we also decide to investigate whether there exists revision behavior difference between ESL and Native speakers besides the two main hypotheses.

To test the hypotheses, we will use both subjective and objective measures. Subjective measures are based on participant post-study survey answers. Ideally, objective measures should be based on an assessment of improvements in the revised drafts; since we do not have evaluative data at this time, we approximate the degree of improvement with the number of revisions, since these two quantities were demonstrated to be positively correlated (Zhang and Litman, 2015). The objective measures are computed from Tables 4.2 and 4.3.

To compare differences between specific sub-groups on the subjective and objective measures, we conduct ANOVA tests with two factors. For example, one factor is the native language of the participant, and another is the interface used.

To determine correlation between quantitative measures, we conduct Pearson and Spearman correlation tests.

Results and Discussions

Testing for H1.3 Comparing participants from the experiment and the control group, we observe some differences. First, we detect that the experiment group agrees with the statement "The system helps me to recognize the weakness of my essay" more so than the control group (Experiment group has a mean ratings of 3.97 ("Agree") while the control group's is 3.17 ("Neutral"), $p < 0.003$, F:9.976, Partial Eta Squared: 0.151).

Second, in the experiment group, there is a trending positive correlation between the number of revisions from Draft2 to Draft3 and the ratings for the statement "The system encourages me to make more revisions than I usually make" (ρ=.33 and $p < .07$); whereas there is no such correlation for the control group. Additional information about revision purposes may elicit a stronger self-reflection response in the experiment group participants.

In contrast, in the control group, there is a significant negative correlation between the number of revisions from Draft1 to Draft2 and ratings for the statement "it is convenient to

[5]In specific, the L2 speakers (proficient non-native speakers)

view my previous revisions with the system" (ρ=-.36 and $p < .05$). This suggests that the character-based interface is ineffective when participants have to reflect on many changes.

On the other hand, when comparing the number of revisions made by both groups on Rev23 (controlling for their Rev12 numbers), we did not find a significant difference (p: 0.330, F:1.180, Partial Eta Squared: 0.079).

As we did not observe a significant difference in the number of revisions made by the two interface groups, we cannot verify that **H1.3** is true; possibly a larger pool of participants is needed, or possibly the writing assignment is not extensive enough (in length and in the number of drafts). Another possible explanation is that the system might only motivate the users to make more revisions when the feedback is different from the user's intention. To further verify the correctness of **H1.3**, we plan to have the essays graded by experts. The graded scores could allow us to analyze whether essays improved more when ArgRewrite was used.

Testing for H1.4 Focusing on the 30 participants from the experiment group, we check the impact of the feedback regarding Rev12 on how they subsequently revise (Rev23). We counted the *Add* and *Modify* revisions where the participant disagrees with the revision purpose assigned by the annotator in Rev12. Of those, we then count the number of times the corresponding sentences were further revised[6]. Of the 53 sentences where the participants disagreed with the annotator, 45 were further revised in the third draft. The ratio is 0.849, much higher than the overall ratio of general Rev12 revisions being further revised in Rev23 ($161/394 = 0.409$) and the ratio of the agreed Rev12 revisions being revised in Rev23 ($67/341 = 0.196$). In further analysis, a Pearson correlation test is conducted to check the correlation between the number of Rev23 and the number of disagreements for different types of agreement/disagreements, controlling for the number of Rev12. We find a negative correlation between Rev23 and the number of cases (r=-0.41, $p < .03$) in which the revisions annotated as *Content* are verified by the participants; we also find a positive correlation between Rev23 and the number of cases (r=0.36, $p <= .05$) in which the revisions annotated as *Surface* are intended to be *Content* revisions by the participants. Both findings are consistent with **H1.4**, suggesting that participants will revise further if they perceive that their intended

[6] *Delete* revisions were ignored as the deleted sentences are not traceable in Draft3

revisions were not recognized.

From the finding in **H1.4** one might argue that the users might revise more when the system makes recognition errors. However, we argue that it is important that the revision assistant should at least be "mostly" accurate to be "trusted" by the users. Thus, it is still important to improve the accuracy of automatic revision identification.

Testing for Language impact We observe that native and L2 speakers exhibit different behaviors.

First, we detect a significant difference in the number of Content and Surface revisions made by L2 and native speakers ($p < .02$ and $p < .03$). More specifically, native speakers tend to make more *Content* changes while the L2 speakers are likely to make more *Surface* changes.

Second, there is also a significant interaction effect among two factors of Group and users' native language ($p < .021$) on their ratings for the statement "the system helps me to recognize the weakness of my essay".

Third, we observe a significant positive correlation in the native group between the number of content revisions in Rev23 and the ratings of the statement "the system encourages me to make more revisions than I usually make" (ρ=.4 and $p < .009$). This suggests that giving feedback (from either interface) encourages native speakers to make more content revisions.

Finally, in the L2 group, there is a significant negative correlation between the number of surface revisions in Rev12 and the ratings for the statement "the system helps me to recognize the weakness of my essay" (ρ=-.57 and $p < .008$). This shows that giving feedback to L2 speakers is less helpful when they make more surface revisions.

4.3 SUMMARY

This chapter describes the prototype revision assistant tool we developed and the user study conducted based on the tool. The tool allows the participant to quickly identify the locations and types of changes they have made. A user study on the effectiveness of such feedback is

then conducted, where the participants revise their essays according to the manually labeled revision feedback. Both the participants' revisions and their subjective ratings to the system were recorded, data analysis was conducted on the collected data to analyze the impact of revision feedback on users' writings.

The user study results provide support for hypotheses **H1.3** and **H1.4**. For **H1.3** on the impact difference of different revision feedback, while we observe a significant difference from the participants' subjective ratings, we cannot observe a significant difference from the number of revisions made by the participants. For **H1.4**, we observe that 1) the participants make more changes when there is discrepancy between their own intention and system recognition 2) Giving feedback encourages native speakers to make more content revisions. It is important to notice that besides the revision interface, there are also other factors that might influence the behaviors of the users' rewritings. Study results have demonstrated the impact of the language difference, yet there are still other factors such as education level to be studied in the future.

5.0 AUTOMATIC REVISION IDENTIFICATION (PIPELINE)

This chapter introduces our works on the automatic identification of revisions. In this chapter the identification of revisions is solved in a pipelined fashion, involving the identification of where the revision happens (Revision Extraction) (Zhang and Litman, 2014) and what the revision is (Revision Classification) (Zhang and Litman, 2015, 2016; Zhang et al., 2016b).

5.1 REVISION EXTRACTION

This section describes our work as stated in (Zhang and Litman, 2014). As introduced in Chapter 2, revisions are extracted by aligning sentences. An added sentence or a deleted sentence is treated as aligned to null. The aligned pairs where the sentences in the pair are not identical are extracted as revisions.

5.1.1 Related Work

We borrow ideas from the research on sentence alignment for monolingual corpora. Existing research usually focuses on the alignment from the text to its summarization or its simplification (Jing, 2002; Barzilay and Elhadad, 2003; Bott and Saggion, 2011). Barzilay and Elhadad (2003) treat sentence alignment as a classification task. The paragraphs are clustered into groups, and a binary classifier is trained to decide whether two sentences should be aligned or not. Nelken and Shieber (2006) further improves the performance by using TF*IDF score instead of word overlap and also utilizing global optimization to take sentence order information into consideration. We argue that summarization could be considered as

a special form of revision and adapted Nelken's approach to our approach.

5.1.2 Alignment Based on Sentence Similarity

The alignment task goes through three stages.

1. Data preparation: for each sentence in the annotated final draft, if it is not a new sentence, create a sentence pair with its aligned sentence in the first draft. The pair is considered to be an aligned pair. Also, randomly select another sentence from the first draft to make a negative sentence pair. Thus we ensure there are nearly equal numbers of positive and negative cases in the training data.

2. Training: according to the similarity metric defined, calculate the similarity of the sentence pairs. A logistic regression classifier predicting whether a sentence pair is aligned or not is trained with the similarity score as the feature. In addition to classification, the classifier is also used to provide a similarity score for global alignment.

3. Alignment: for each pair of paper drafts, construct sentence pairs using the Cartesian product of sentences in the first draft and sentences in the final. Logistic regression classifier is used to determine whether the sentence pair is aligned or not.

We added Levenshtein distance (LD) (Levenshtein, 1966) as another similarity metric in addition to Nelken's metrics. Together three similarity metrics were compared: Levenshtein Distance, Word Overlap(WO), and TF*IDF.

5.1.3 Global Alignment

Sentences are likely to preserve the same order between rewritings. Thus, sentence ordering should be an important feature in sentence alignment. Nelken's work modifies the Needleman-Wunsch alignment (Needleman and Wunsch, 1970) to find the sentence alignments and goes in the following steps.

Step1: The logistic regression classifier previously trained assigns a probability value from 0 to 1 for each sentence pair $s(i, j)$. Use this value as the similarity score of sentence pair: $sim(i, j)$.

Step2: Starting from the first pair of sentences, find the best path to maximize the likelihood between sentences according to the formula $s(i,j) = max\{s(i-1,j-1)+sim(i,j), s(i-1,j) + \mathbf{sim(i,j)}\ , s(i,j-1) + \mathbf{sim(i,j)}\}$

Step3: Infer the sentence alignments by back tracing the matrix $s(i,j)$.

We found out that changing bolded parts in the formula to $s(i,j) = max\{s(i-1,j-1) + sim(i,j), s(i-1,j) + insertcost\ , s(i,j-1) + deletecost\}$ shows better performance in our problem. *insertcost* and *deletecost* are both set to 0.1 as they are found to be the most effective during practice.

5.1.4 Experiments and Evaluation

We use accuracy as the evaluation metric. For each pair of drafts, we count the number of sentences in the final draft N_1. For each sentence in the final draft, we count the number of sentences that get the correct alignment as N_2. The accuracy of the sentence alignment is $\frac{N_2}{N_1}$.

We use Hashemi's (Hashemi and Schunn, 2014) approach as the baseline. For our method, we tried four groups of settings. Group 1 and group 2 perform leave-one-out cross validation on corpora *Align1* and *Align2* (test on one pair of paper drafts and train on the others). Group 3 and group 4 train on one corpus and test on the other.

Group	Levenshtein Distance	Word Overlapp	TF*IDF	Baseline
1	0.9811	0.9863	**0.9931**	0.9427
2	0.9649	0.9593	**0.9667**	0.9011
3	**0.9727**	0.9700	**0.9727**	0.9045
4	0.9860	**0.9886**	0.9798	0.9589

Table 5.1: Accuracy of our approach vs. baseline on Corpora *Align1* and *Align2*

Table 5.1 shows that all our methods beat the baseline on corpora *Align1* and *Align2*[1].

[1]For Groups 1 and 2, we calculate the accuracy of Hashemi's approach under a leave-one-out setting, each time remove one pair of document and calculate the accuracy. A significance test is also conducted, the worst metric LD in Group 1 and WO in Group 2 both beat the baseline significantly ($p_1 = 0.025$,$p_2 = 0.017$) in two-tailed T-test.

Among the three similarity metrics, TF*IDF is the most predictive.

We also test our algorithms on corpora *HSchool1* and *HSchool2* separately with the setting of the 10-fold (student) cross-validation using TF*IDF as the similarity metric. We achieve 0.9199 accuracy on *HSchool1* and 0.9121 accuracy on *HSchool2*. We achieved 0.8991 accuracy when trained on *HSchool1* and tested on *HSchool2* and 0.9119 accuracy in reverse.

We also tested our algorithms on the corpus *ArgRewrite* with 10-fold (student) cross-validation[2] and achieved accuracy 0.9328.

5.2 REVISION CLASSIFICATION USING FEATURES FROM EXISTING WORKS

This section describes our work in (Zhang and Litman, 2015), where we investigated whether the existing features and approaches in Wikipedia revision classification can be adapted to the prediction of argumentative writing revisions.

5.2.1 Related Work

There are multiple works on the classification of revisions (Adler et al., 2011; Javanmardi et al., 2011; Bronner and Monz, 2012; Daxenberger and Gurevych, 2013; Zhang and Litman, 2015). While different classification tasks were explored, similar approaches were taken by extracting features (location, text, meta-data, language) from the revised text to train a classification model (SVM, Random Forest, etc.) on the annotated data.

As our task focuses on identifying the argumentative purpose of writing revisions, work in argument mining is also relevant. In fact, many features for predicting argument structure (e.g., location, discourse connectives, punctuation) (Burstein and Marcu, 2003; Moens et al., 2007; Palau and Moens, 2009; Feng and Hirst, 2011) are also used in revision classification. Our work also investigated the use of such features. Different from works in argument mining

[2]Note that the revisions from the same student will be either all in the training data or all in the test data

which extract features from one single draft, our work collected features for sentences in both drafts.

5.2.2 Classifying Revisions in Isolation

We first followed prior works and investigated features that can be used in argumentative revision classification. In this section we only extract features from the revision sentence pair and ignores the contextual information around the revision. As shown in Table 5.2, besides using unigram features as a baseline, our features are organized into *Location*, *Textual*, and *Language* groups following prior work (Adler et al., 2011; Bronner and Monz, 2012; Daxenberger and Gurevych, 2013).

Baseline: unigram features. Similarly to Daxenberger and Gurevych (2012), we choose the count of unigram features as a baseline. Two types of unigrams are explored. The first includes unigrams extracted from all the sentences in an aligned pair. The second includes unigrams extracted from the differences of sentences in a pair.

Location group. As Falakmasir et al. (2014) have shown, the positional features are helpful for identifying thesis and conclusion statements. Features used include the location of the sentence and the location of paragraph .[3]

Textual group. A sentence containing a specific person's name is more likely to be an example for a claim; sentences containing "because" are more likely to be a sentence of reasoning; a sentence generated by content revisions is possibly more different from the original sentence compared to a sentence generated by surface revisions. These intuitions are operationalized using several feature groups: *Named entity features*[4] (also used in Bronner and Monz (2012)'s Wikipedia revision classification task), *Discourse marker features* (used by Burstein et al. (2003) for discourse structure identification), *Sentence difference features* and *Revision operation* (similar features are used by Daxenberger and Gurevych (2013)).

Language group. Different types of sentences can have different distributions in POS tags (Daxenberger and Gurevych, 2013). The difference in the number of spelling/grammar

[3]Since Add and Delete operations have only one sentence in the aligned pair, the value of the empty sentence is set to 0.

[4]Stanford parser (Klein and Manning, 2003) is used in named entity recognition.

mistakes[5] is a possible indicator as Conventions/Grammar/Spelling revisions probably decrease the number of mistakes.

Text	From the 3rd paragraph of Draft 1 (5 paragraphs) (1, The third circle is for Wrathful people.), (2, Saddam Hussein and Osama Bin Laden come to mind when mentioning wrathful person) From the 3rd paragraph of Draft 2 (7 paragraphs) (1, The third circle contains wrathful people), (2, Fidel Castro comes to mind when mentioning wrathful people)
Features	**Unigram** Unigrams of all: ["Saddam", "Hussein", "and", "Osama", "Bin", "Laden", "come", "to", "mind", "when", "mentioning", "wrathful", "people", "Fidel", "Castro", "comes"] Unigrams of diff: ["Saddam", "Hussein", "and", "Osama", "Bin", "Laden", "Fidel", "Castro", "come", "comes"] **Location** First sentence of paragraph? Draft 1: No, Draft 2: No Last sentence of paragraph? Draft 1: No, Draft 2: No First paragraph of essay? Draft 1: No, Draft 2: No Last paragraph of essay? Draft 1: No, Draft 2: No Sentence in the paragraph (Ratio) Draft 1: (2-1)/(5-1) = 0.25, Draft 2: 0.125 Diff: -0.125 Sentence in the paragraph (Number): Draft 1: 2, Draft 2: 2, Diff: 0 Paragraph in the essay ... **Textual** Named entity:

[5]The spelling/grammar mistakes are detected using the languagetool toolkit (https://www.languagetool.org/).

	PERSON count: Draft 1: 2, Draft 2: 1, Diff: -1 LOCATION count: Draft 1: 0, Draft 2: 0, Diff: 0 Discourse markers: Contains "because", "due to",... Draft 1: No, Draft 2, No ... Sentence difference: Diff in commas: 0, Diff in digits: 0, ...Edit distance: 31 Revision Operation: Modify **Language** POS tags: count of adjectives: Draft 1: 1, Draft 2: 1, Diff: 0 count of nouns: ... ... Ratio of POS tags ratio of adjectives: Draft 1: 0.077 Draft 2: 0.111, Diff: 0.034 ... Spelling mistakes: Draft 1: 0, Draft 2: 0, Diff: 0 Grammar mistakes: Draft 1: 0, Draft 2: 0, Diff: 0

Table 5.2: An example of features extracted for the aligned sentence pair (2->2).

5.2.3 Experiments and Results

Experiments

We conducted three different experiments to compare the performance of our approaches. In the first two experiments, the performance on surface vs. content classification are compared both intrinsically and extrinsically. In the third experiment, we compared the effect of different feature groups using SVM[6] as the classifier for binary classification tasks on each

[6]We compared three models used in discourse analysis and revision classification (C4.5 Decision Tree, SVM and Random Forests) (Burstein et al., 2003; Bronner and Monz, 2012; Stab and Gurevych, 2014) and SVM yielded the best performance.

revision purpose.

Paired t-tests are utilized to compare whether there are significant differences in performance. Performance is measured using unweighted F-score. In the extrinsic evaluation, we repeat the corpus study from Chapter 3 using the predicted counts of revision. If the results in the intrinsic evaluation are solid, we expect that a similar conclusion could be drawn with the results from either predicted or manually annotated revisions.

Experiment 1: Surface vs. Content

As the corpus study in Chapter 3 shows that only content revisions predict writing improvement, our first experiment is to check whether we can distinguish between the surface and content categories. The classification is done on all the non-identical aligned sentence pairs with *Modify* operations[7]. We choose 10-fold (student) cross-validation for our experiment. SVM of the Weka toolkit (Hall et al., 2009) is chosen as the classifier for the unigram baseline. Considering the data imbalance problem, the training data is sampled with a cost matrix decided according to the distribution of categories in training data in each round. Two baselines (Majority and Unigram) were compared. The results are evaluated both intrinsically and extrinsically.

Experiment 2: Pipelined revision extraction and classification

In this experiment, revision extraction and Experiment 1 are combined together as a pipelined approach. The output of sentence alignment is used as the input of the classification task. The predicted *Add* and *Delete* revisions are directly classified as content changes. Features are used as in Experiment 1.

Experiment 3: Binary classification for each revision purpose category

In this experiment, we test whether the system could identify if revisions of each specific category exist in the aligned sentence pair or not. The same experimental setting for surface vs. content classification is applied. This experiment compares the effects of different features (*Textual*, *Language*, *Location*) using the SVM model.

Analysis

Analysis of Experiment 1, 2

Table 5.3 presents the results of the classification between surface and content changes

[7] *Add* and *Delete* pairs are removed from this task as only content changes have *Add* and *Delete* operations.

HSchool1	**N = 381**	**Precision**	**Recall**	**F-score**
	Majority	32.65	50.00	37.14
	Unigram	45.47	49.82	46.71
	Basic	**62.55**∗	**58.01**∗	**54.39**∗
HSchool2	**N = 558**	**Precision**	**Recall**	**F-score**
	Majority	32.59	50.00	37.69
	Unigram	48.01	47.09	42.01
	Basic	**57.60**∗	**51.17**∗	**49.64**∗
ArgRewrite	**N = 937**	**Precision**	**Recall**	**F-score**
	Majority	47.85	50.00	48.89
	Unigram	47.85	50.00	48.89
	Basic	**50.40**∗	**50.60**∗	**49.98**∗

Table 5.3: Experiment 1 on corpora HSchool1, HSchool2 and ArgRewrite (Surface vs. Content): average unweighted precision, recall, F-score from 10-fold (student) cross-validation; Basic represents the combination of features Location, Textual, Language and Unigram; ∗ indicates significantly better than majority and unigram.

HSchool1	**Predicted purposes**	**R**	**p**
	#All revisions (N = 1273)	0.52	<**0.001**
	#Surface revisions	0.18	0.245
	#Content revisions	0.55	<**0.001**
	Pipeline predicted purposes	**R**	**p**
	#All (predicted N = 1356)	0.51	<**0.001**
	#Surface revisions	0.23	0.124
	#Content revisions	0.54	<**0.001**
HSchool2	**Predicted purposes**	**R**	**p**
	#All revisions (N = 1054)	0.27	**0.041**
	#Surface revisions	-0.03	0.808
	#Content revisions	0.27	**0.038**
	Pipeline predicted purposes	**R**	**p**
	#All (predicted N = 1101)	0.27	**0.039**
	#Surface revisions	-0.05	0.775
	#Content revisions	0.27	**0.042**

Table 5.4: Partial correlation between number of predicted revisions and Draft2/Essay2 score on corpora HSchool1 and HSchool2. (Upper: Experiment 1, Lower: Experiment 2)

N = 1273	**Content**				**Surface**		
Experiments	**Claim**	**Warrant**	**General**	**Evidence**	**Org.**	**Word**	**Conv**
	110	390	353	110	45	84	180
Majority	47.87	41.44	41.12	47.90	25.49	46.12	48.06
Unigram	59.67	61.64	66.38	48.39	49.23	51.63	53.58
All features	62.30	**67.08**∗	**72.47**∗	48.28	**54.01**∗	**73.79**∗	**70.95**∗
Textual+unigram	56.18	64.84∗	72.08∗	49.55	52.62∗	58.75∗	66.35∗
Language+unigram	57.76	66.27∗	69.23∗	48.81	49.21	65.01∗	69.62∗
Location+unigram	**62.79**∗	66.46∗	70.55∗	49.61	49.25	52.36	49.25

Table 5.5: Experiment 3 on corpus HSchool1: average unweighted F-score from 10-fold (student) cross-validation; ∗ indicates significantly better than majority and unigram baselines. *Rebuttal* is removed as it only occurred once.

N = 1054	**Content**				**Surface**	
Experiments	**Claim**	**Warrant**	**General**	**Evidence**	**Word**	**Conv**
	76	327	216	34	283	109
Majority	43.23	37.05	40.17	44.25	37.91	42.17
Unigram	44.44	51.50	47.09	44.48	50.33	47.44
All features	**46.03**∗	**60.18**∗	46.96	47.56∗	**64.89**∗	68.75∗
Textual+unigram	43.22	56.64∗	45.72	48.62∗	64.33∗	68.19∗
Language+unigram	44.34	54.50∗	47.23	45.81	65.01∗	**69.62**∗
Location+unigram	45.61∗	56.73∗	**47.53**∗	**49.28**∗	54.96∗	48.37

Table 5.6: Experiment 3 on corpus HSchool2: average unweighted F-score from 10-fold (student) cross-validation; ∗ indicates significantly better than majority and unigram baselines. *Rebuttal* and *Organization* are removed because of rare occurrence.

N = 1757	**Content**				**Surface**		
Experiments	**Claim**	**Warrant**	**General**	**Evidence**	**Org.**	**Word**	**Conv**
	42	463	260	67	42	719	127
Majority	49.41	42.54	46.06	48.99	49.39	36.90	48.06
Unigram	49.41	55.97	48.24	48.99	49.39	71.51	48.76
All features	49.41	**75.78**∗	**55.41**∗	48.99	49.39	**86.64**∗	**64.03**∗
Textual+unigram	49.41	70.81∗	51.76∗	48.99	49.39	86.27∗	58.65∗
Language+unigram	49.41	63.27∗	49.23∗	48.99	49.39	86.11∗	63.62∗
Location+unigram	49.41	67.17∗	50.56∗	48.99	49.39	80.17	48.01

Table 5.7: Experiment 3 on corpus ArgRewrite: average unweighted F-score from 10-fold (student) cross-validation; ∗ indicates significantly better than majority and unigram baselines. *Rebuttal* is removed as it only occurred once.

on corpora *HSchool1*, *HSchool2* and *ArgRewrite*. Results show that our learned models significantly beat majority and unigram baselines for all of unweighted precision, recall and F-score.

According to Table 5.4 , the conclusions drawn from the predicted revisions and annotated revisions are similar (Table 3.3). Content changes are significantly correlated with writing improvement, while surface changes are not. We observe significant correlation between content changes and we can also see that the coefficient of the predicted content change correlation is close to the coefficient of the manually annotated results.

Analysis of Experiment 3

Table 5.5, Table 5.6 and Table 5.7 show the classification results for the fine-grained categories. Our results are not significantly better than the unigram baseline on *Evidence* of *HSchool1*. While the poor performance on *Evidence* might be due to the skewed class distribution, our model also performs better on *Conventions* where there are not many instances. For the categories where our model performs significantly better than the baselines, we see that the location features are the best features to add to unigrams for the content changes (significantly better than baselines except *Evidence* on *HSchool1* and better than all baselines on *HSchool2*), while the language and textual features are better for surface changes. The contribution of the feature groups also varies on different corpora. For example, the

textual feature group is predictive for *Evidence* changes on *HSchool2* while not predictive on *HSchool1*. We have similar findings on corpus *ArgRewrite*, where textual features and location features work better for *Content* revisions and language features work better for *Surface* revisions. We also see that using all features does not always lead to better results, probably due to over fitting. Replicating experiments in these three corpora also demonstrates that our schema and features can be applied across essays with different topics with similar results.

5.3 ENHANCE THE CLASSIFICATION PERFORMANCE WITH CONTEXTUAL FEATURES

Because the investigation of the feature groups for revision classification indicated there was still significant room for improvement, we explored the ways to enhance the classification performance. In this study, we focus on the classification task assuming we have the gold standard alignments. As we focus on argumentative changes, we merge all the *Surface* subcategories into one *Surface* category. As we found that both *Rebuttals* and multiple labels for a single revision were rare, we merge *Rebuttal* and *Warrant* into one *Warrant* category and allow only a single (primary) label per revision. Different revision types are assigned different priority and the type with highest priority is selected[8]. This simplification allows us to explore the classification of revision purposes using the sequence labeling technique. The features used in the previous section were used together as the baseline features. This section describes our work in (Zhang and Litman, 2016).

5.3.1 Related Work

Lawrence et al. (2014) use changes in topic to detect argumentation, which leads us to hypothesize that different types of argumentative revisions will have different impacts on text cohesion and coherence. Guo et al. (2011) and Park et al. (2015) both utilize Condi-

[8]Order of priority: *Claim* > *Rebuttal* > *Evidence* > *Reasoning* > *General* > *Word Usage/Clarity* > *Conventions/Grammar/Spelling*

tional Random Fields (CRFs) for identifying argumentative structures. While we focus on the different task of identifying revisions to argumentation, we similarly hypothesize that dependencies exist between revisions and thus utilize CRFs in our task.

5.3.2 Methodology

Adding contextual features

We proposed two new types of contextual features to enhance the classification performance. The first type (**Ext**) extracts the baseline features from not only the aligned sentence pair representing the revision in question, but also for the sentence pairs before and after the revision. The second type (**Coh**) measures the cohesion and coherence changes in a 2-sentence block around the revision.

Utilizing the cohesion and coherence difference.

Inspired by works in (Lee et al., 2015; Vaughan and McDonald, 1986), we hypothesize that different revisions can have different impacts on the cohesion and coherence of the essay. We propose to extract features for both impact on cohesion (lexical) and impact on coherence (semantic). Inspired by Hearst (1997), sequences of blocks are created for sentences in both Draft 1 and Draft 2 as demonstrated in Figure 5.1. Two types of features are extracted. The first type describes the cohesion and coherence between the revised sentence and its adjacent sentences. The similarity (lexical/semantic) between the revised sentence block and the sentence block before ($Sim(Block\_Up, Block\_Up\_Self)$) and after ($Sim(Block\_Down, Block\_Down\_Self)$) are calculated as the cohesion/coherence scores Coh_Up and Coh_Down. The features are extracted separately for Draft 1 and Draft 2 sentences[9]. The second type describes the impact of sentence modification on cohesion and coherence[10]. Features Change_Up and Change_Down are extracted as the division of the cohesion/coherence scores of two drafts ($\frac{Coh\_Up(Draft2)}{Coh\_Up(Draft1)}$, $\frac{Coh\_Down(Draft2)}{Coh\_Down(Draft1)}$).

A bag-of-word representation is generated for each sentence block after stop-word filtering and stemming. Jaccard similarity is used for the calculation of lexical similarity between sentence blocks. Word embedding vectors (Mikolov et al., 2013) are used for the calculation

[9]For the added and deleted sentences, features of the empty sentence in the other draft are set to 0.
[10]The feature values of sentence additions/deletions are 0

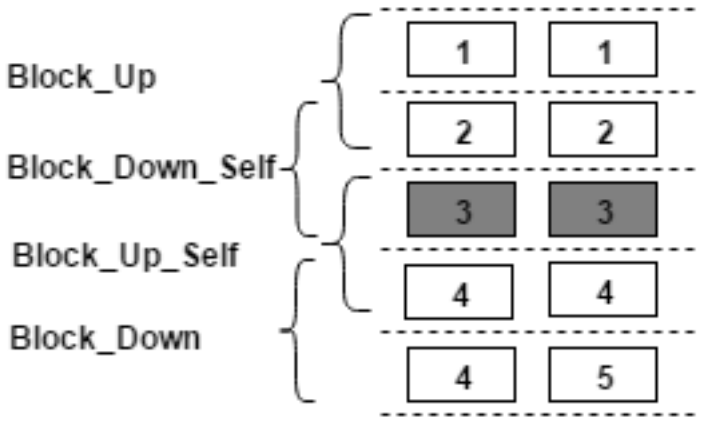


Figure 5.1: Example of cohesion blocks. A window of size 2 is created for both Draft 1 and Draft 2. Sequence of blocks were created by moving the window at the step of 1 (sentence).

of semantic similarity. A vector is calculated for each sentence block by summing up the embedding vectors of words that are not stop-words. Afterwards the similarity is calculated as the cosine similarity between the block vectors. This approach has been taken by multiple groups in the SemEval-2015 semantic similarity task (SemEval-2015 Task 1)(Xu et al., 2015).

Transforming to sequence labeling

To capture dependencies among predicted revisions, we transform the revisions to a consecutive sequence and label it with Conditional Random Fields (CRFs) as demonstrated in Figure 5.2. For both drafts, sentences are sorted according to their order of occurrence in the essay. Aligned sentences are put into the same row and each aligned pair of sentences is treated as a unit of revision. The "cross-aligned" pairs of sentences[11] (which does not often occur) are broken into deleted and added sentences[12]. After generating the sequence, each revision unit in the sequence is assigned the revision purpose label according to the annotations, with unchanged sentence pairs labeled as *Nochange*.

We conducted labeling on both essay-level and paragraph-level sequences. The essay-

[11]Sentences in Draft 1 switched their positions in Draft 2, the cross-aligned sentences cannot be both in the same row and and following their order of occurrence at the same time.

[12]I.e, the cross-aligned sentences in Draft 1 are treated as deleted and the sentences in Draft 2 are treated as added.

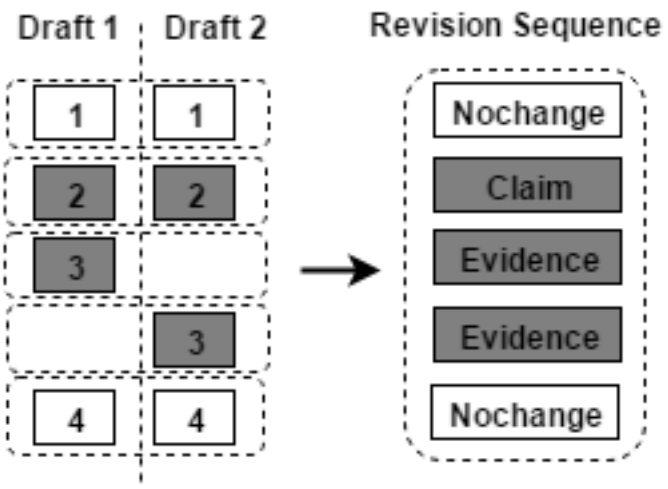


Figure 5.2: Example of revision sequence transformation. Each square corresponds to a sentence in the essay, the number of the square represents the index of the sentence in the essay. Dark squares are sentences that are changed. In the example, the 2nd sentence of Draft 1 is modified, the 3rd sentence is deleted and a new sentence is added in Draft 2.

level treats the whole essay as a sequence segment while the paragraph-level treats each paragraph as a segment. After labeling, the label of each changed sentence pair is marked as the purpose of the revision[13].

5.3.3 Experiments and results using the contextual information enhancement

In this work, we directly compare the performance of our CRF + Contextual features approach with the SVM baseline[14].

Experiments

We tested whether the performance is improved by considering contextual features and transforming the problem to a sequence labeling problem. We first conducted a 10-fold (student) cross-validation on the performance of surface vs. text classification using the

[13]Revisions on cross-aligned pairs are marked as *Surface*.

[14]SVM model implemented with Weka (Hall et al., 2009) and CRFs model implemented with CRFSuite (Okazaki, 2007)

Corpora	**Precision**	**Recall**	**F-score**
HSchool1 (SVM)	62.55	58.01	54.39
HSchool1	**70.66**∗	**68.23**∗	**69.27**∗
HSchool2 (SVM)	57.60	51.17	49.64
HSchool2	**66.79**∗	**63.46**∗	**64.96**∗
ArgRewrite (SVM)	50.40	50.60	49.98
ArgRewrite	**53.50**∗	**52.41**∗	**53.16**∗

Table 5.8: Experiment 1 using the enhanced approach on corpora *Hschool1*, *HSchool2* and *ArgRewrite* (Surface vs. Content): average unweighted precision, recall, F-score from 10-fold (student) cross-validation; same set of folds as Table 5.3 are used for comparison, all results are significantly better than the SVM approach

contextual enhancement approach. The same set of data were used as in Experiment 1 for the comparison with the SVM model. Afterwards we compared the SVM model and the enhanced approach in a 5-class classification setting (Experiment 4)[15]. We merge all the surface revision types into one "Surface" revision type and ignore the revision type (Rebuttal/Reservation) that rarely occurs. In the experiment we also examined whether the contextual features can improve the performance of the SVM model. All experiments are conducted using 10-fold (student) cross-validation with 300 features selected using learning gain ratio[16].

Analysis

Table 5.8 demonstrate that our CRF approach is significantly better than the prior SVM approach on surface vs. content classification. On multi-class classification, as demonstrated in Table 5.9, we observe that the **Coh** features yield a non-significant improvement over the baseline features on Corpus *HSchool1*, and a significant improvement on Corpora *HSchool2*

[15]Experiment 3 (binary classification for each category) is not fit for the testing of the CRFs approach as the CRFs approach needs to take advantage of the label information

[16]We tested with parameters 100, 200, 300, 500 on a development dataset disjoint from HSchool1, HSchool2 and ArgRewrite and chose 300 which yielded the best performance.

		SVM				CRFs
		Base(B)	**B+Ext**	**B+Coh**	**All**	**All**
HSchool1	**P**	0.666	0.689	0.673	0.684	**0.701**∗
	R	0.620	0.632	0.630	0.630	**0.642**∗
	F	0.615	0.630	0.619	0.626	**0.643**∗
HSchool2	**P**	0.530	0.543	0.559 ∗	0.553∗	**0.655**∗
	R	0.516	0.525	0.534 ∗	0.532	0.532
	F	0.502	0.510	0.524 ∗	0.520∗	**0.584**∗
ArgRewrite	**P**	0.565	0.575	0.590 ∗	0.610∗	**0.658**∗
	R	0.544	0.546	0.543	0.532	0.524
	F	0.533	0.540	0.563 ∗	0.540	**0.640**∗

Table 5.9: Experiment 4. The average of 10-fold (student) cross-validation 5-class classification (*Claim/Ideas*, *Warrant/Reasoning/Backing*, *Evidence*, *General Content*, *Surface*) results on Corpora HSchool1, HSchool2 and ArgRewrite. Unweighted average precision (P), Unweighted recall (R) and Unweighted F-measure (F) are reported. Results of CRFs on paragraph-level segments are reported (there is no significant difference between essay level and paragraph level). The first four columns of Table 5.9 show the performance of baseline features with and without our new contextual features using an SVM prediction model. The last column shows the performance of CRFs using all features. ∗ indicates significantly better than the baseline, **Bold** indicates significantly better than all other results (Paired T-test, $p < 0.05$).

and *ArgRewrite* . This indicates that changes in text cohesion and coherence can indeed improve the prediction of argumentative revision types. The **Ext** feature set - which computes features for not only the revision but also its immediately adjacent sentences - also yields a slight (although not significant) improvement. However, adding the two feature sets together does not further improve the performance using the SVM model. The CRF approach almost always yields the best results for all corpora, with all such CRF results significantly better than all other results. This indicates that dependencies exist among argumentative revisions that cannot be identified with traditional classification approaches.

5.4 ENHANCING THE CLASSIFICATION PERFORMANCE WITH DISCOURSE INFORMATION

Discourse analysis is a hot research topic recently. We also believe using discourse analysis results can improve the performance of classification since the discourse relations represent the impact of the revised text to the argument. For example, if the discourse relation of a sentence with its adjacent sentence is *Expansion*, the sentence is less likely to be the thesis of the essay, and thus unlikely to be a *Claim* revision. This is inspired by (Cabrio et al., 2013) where the discourse relations are mapped to argument structures. In (Forbes-Riley et al., 2016), discourse relations on corpus *HSchool1* were annotated under the Penn Discourse Treebank (PDTB) Framework, which allows us to explore whether it is possible to improve the performance with discourse information. This section describes our work in (Zhang et al., 2016b).

5.4.1 PDTB Introduction

We decide to have the discourse relationship described using the Lexicalized Tree Adjoining Grammar for Discourse (D-LTAG) (Webber, 2004) as it is easier to adapt its discourse relations to the features used in our model. The Penn Discourse TreeBank (PDTB) (Prasad et al., 2008) is an annotated corpus based on the D-LTAG model and multiple automatic

Draft2 Essay	(1) The lustful are those who long and crave for one another. (2) The person guilty of lust is put in this layer of hell because of his over indulgence of sexual-pleasure. (3) The man who is stuck in this layer is Hue Heffner. (4) He has devoted his entire life for other people 's lustful pleasure and his own. (5) He has spent millions on working on his mansion which is for the purpose of other lustful desires. (6) People who were stuck in this layer are constantly whipped around and " banging " into one another. (7) What you do in your Earthly presence follows with you into Hell. (8) For him and like many others he is now tortured in a whirlwind of torment with others lustful accommodators with himself.
Annotated PDTB	(1->2, EntRel), (2->3, Expansion), (3->4, Contingency), (4->5, Expansion), (5->6, EntRel), (6->7, Contingency), (7->8, Contingency)

Table 5.10: A paragraph from an essay about putting contemporaries into different levels of hell (top), and annotated PDTB relations between sentences (bottom). The paragraph can be divided into two segments. In the first segment (sentences (1) to (3)) the author introduces the person to be put in the lustful layer. In the second segment (sentences (4) to (8)), the author states why this person belongs there and how he will be treated. PDTB relations are processed from PDTB annotations ignoring the discourse connectives, e.g. (1->2, EntRel) represents the discourse information: (Arg1: Sentence1, Arg2: Sentence2, Relation Type: EntRel).

discourse analysis research works have been done on the corpus (Pitler et al., 2009; Pitler and Nenkova, 2009; Zhou et al., 2010; Wang et al., 2010). PDTB-style annotation (Prasad et al., 2008) adopts a lexically grounded approach by anchoring discourse relations according to discourse connectives. In a typical PDTB annotation process, an annotator first locates discourse connectives (explicit or implicit) then annotates text spans as their arguments. While the process of manual PDTB annotation has been demonstrated to yield reliable results (Alsaif and Markert, 2011; Danlos et al., 2012; Zhou and Xue, 2015; Zeyrek et al., 2013), it yields more shallow annotation when compared to another widely-used discourse scheme, namely Rhetorical Structure Theory (RST) (Mann and Thompson, 1988; Carlson et al., 2002). This is because when using RST a text is represented as a hierarchical discourse tree, while when using PDTB the relations exist only locally (typically between sentences or clauses). Table 5.10 presents an example of PDTB annotation.

The lack of discourse information across larger contexts potentially limits the utility of PDTB-style labels. Feng et al. (2014) found that when applied to the tasks of sentence ordering and essay scoring, an RST-style discourse parser outperformed a PDTB-style parser. Performance on both tasks was also likely impacted by parsing errors. To address both the local nature of PDTB-style annotations as well as the errors introduced by state-of-the-art discourse parsers, we propose to first build paragraph-level discourse structures from annotated PDTB labels, then to infer discourse relations based on these structures. We hypothesize that features extracted from inferred relations will improve performance in downstream applications, compared to features extracted from only original annotations. Thus three approaches were attempted to utilize PDTB information for revision classification. Besides using PDTB annotations directly, two approaches were proposed to infer long-distance PDTB relations between sentences.

5.4.2 Intuitions for PDTB Inference

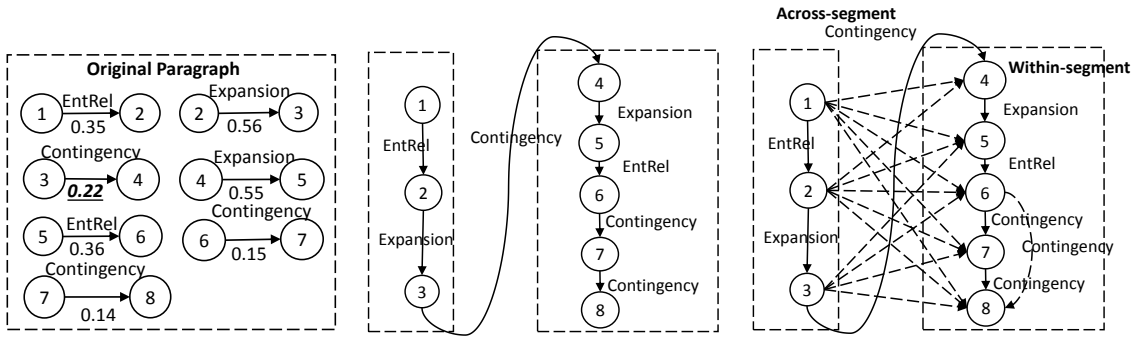
Different from other discourse annotations, the PDTB annotation schema anchors at the labeling of discourse connectives and labels text spans around the connective. The annotator either locates the "Explicit" connectives or manually fills in the "Implicit" connectives be-

tween two text spans. The text span where the connective structurally attaches to is called **Arg2**, while the other text span is called **Arg1**. The spans are usually used at the level of sentence/phrase. In Prasad et al. (2014), five relation types are annotated: *Explicit*, *Implicit*, *AltLex*, *EntRel* and *NoRel*. Within the *Explicit*/*Implicit* relations, the senses of relations are further categorized at multiple levels. In Level-1, the relations are categorized to 4 senses: *Comparison*, *Contingency*, *Expansion* and *Temporal*. We focus on the type/sense of Level-1 relations only and ignore the discourse connectives[17]. *Arg1*, *Arg2* and the discourse relation type/sense are used as demonstrated in Table 5.10. For the *Explicit*/*Implicit* relations, we use the sense of the relation directly to represent the relation. Below we explain our intuitions for inferring new discourse relations within the paragraph.

Intuition 1. Latent discourse relations can be inferred from annotated discourse relations. In this paper we explore two possible cases: **1) Same type transition**: If sentence A has relation type T with sentence B and sentence B has the same relation type with sentence C, we can infer that A has relation type T with C. In the example in Table 5.10, a *Contingency* relation between sentences 6 and 8 will be inferred from the *Contingency* relationships between sentences (6,7) and sentences (7,8). **2) Across segment propagation**: If a paragraph can be segmented to text segments semantically dissimilar to each other (i.e. the two text segments are serving different semantic purposes), the discourse relation of sentences on the boundary of two segments can be propagated to infer weaker relations between all sentences in the segments. In the example in Table 5.10, due to the discourse relation between sentences 3 and 4 and the segment boundary between them, the segment from 4 to 8 will also be viewed as a reasoning (*Contingency*) of the segment from 1 to 3 (why and how Hue Heffner belongs to the lustful level), and weak relations are inferred between sentences (1,2,3) and (4,5,6,7,8).

Intuition 2. The importance of discourse relations to argumentation varies even if the relation types are the same. The relations connecting the semantically dissimilar segments are likely to be more important than the relations within a segment. In Table 5.10, the *Contingency* relation between sentences 3 and 4 transits the thesis introduction to the arguments supporting the thesis. The *Contingency* relation between sentences 6 and 7 is just

[17]We plan to explore connectives in future work.



(a) Original PDTB annotation (b) After linear segmentation (c) After relation propagation

Figure 5.3: The construction of PDTBSegment structure of the example in Table 5.10. As sentence similarity between 3 and 4 is 0.22, smaller than the value 0.56 (before) and 0.55 (after), the paragraph is segmented to segment(1-3) and segment (4-8). Afterwards relations are inferred both within the segment and across the segments. The dashed lines represent the propagated relations.

a transition to smooth the description of how Hue Heffner is going to be treated.

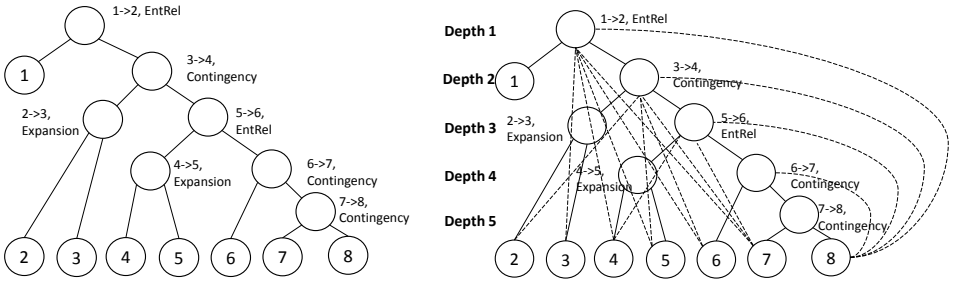
5.4.3 PDTB Inference - PDTBSegment

Based on intuition 1, the *PDTBSegment* approach emphasizes the inference of discourse relations.

Step1. Linear segmentation. Inspired by the TextTiling algorithm (Hearst, 1997) for linear segmentation, we utilize the "valley" of semantic similarity scores between sentences as the segmentation boundary.

The summed word-embedding vector is calculated for each sentence[18] and cosine value between vectors is used as the similarity score. Similarity scores indicates a possible segmentation boundary. In the example of Figure 5.3(a), the similarity between (2,3) and the similarity between (4,5) are larger than the similarity between (3,4), in other words, sentence 3 and 4 has a low similarity score preceded by and followed by high similarity scores, thus

[18]Pre-trained word2vec vectors from (Mikolov et al., 2013).



(a) After tree structure construction (b) After relation propagation

Figure 5.4: PDTBTree structure of Table 5.10 example. The dashed lines represent the propagated relations.

the paragraph is first segmented into segment (1,2,3) and segment (4,5,6,7,8) as in Figure 5.3(b).

Step2. Relation inference. **1) Within segment.** We conduct "same type transition" for sentences within the same segment. As in Figure 5.3(c), there exists relation *Contingency* between 6 and 8 as the same relationship exists between 6, 7 and 7, 8. **2) Across segment.** "Across-segment propagation" is conducted for sentences in different segments. If there exists relation (type T) between two segments Seg1 and Seg2, a relation with type T is inferred for each sentence in Seg1 and each sentence in Seg2. In Figure 5.3(c), we propagate the *Contingency* relations between sentence (1,2,3) and sentences (4,5,6,7,8).

5.4.4 PDTB Inference - PDTBTree

PDTBTree focuses on intuition 2 using sentence aggregation. To better separate important discourse relations, a hierarchical tree structure is constructed for each paragraph and relations then inferred.

Step 1. Tree construction. As in Figure 5.4(a), the tree is constructed iteratively starting with each sentence constructed as a leaf node. Semantic similarities between adjacent sentences are calculated in the same manner as the *PDTBSegment* approach. In each round, the two most similar nodes are selected and merged into one node and similarities

between the merged node and its adjacent nodes are calculated[19]. The selection and merge of nodes repeats until there is only one root node left.

Discourse relations are assigned to the non-leaf nodes after tree construction. For each tree node, sentences in its left and right child are listed as $Nodes_{left}$ and $Nodes_{right}$. Relations with Arg1 in $Nodes_{left}$ and Arg2 in $Nodes_{right}$ are assigned to the merged node. For example, the discourse relation (1->2, EntRel) is assigned to the root node as sentence 1 is in its left child and sentence 2 is in its right child. After this step we bind each non-leaf node with one or several discourse relations.

Step 2. Relation inference. Relations are first assigned different levels of importance as depths. As in Figure 5.4(b), the assignment starts at the root node and traverses the whole tree until all the non-leaf nodes are labeled. Depth starts from 1 and smaller number indicates larger importance. As in the example, we notice that the transition from the thesis to its reasoning (3->4) is recognized as a depth-2 relation while the transitions between sentences 6,7,8 are recognized as depth-4 and depth-5 relations.

Afterwards discourse relations are inferred by traversing up from the leaf nodes back to their parent nodes. The parent node is used as the discourse connector and its child leaf nodes are used as Arg1 and Arg2. For example, in Figure 5.4(b), sentence 3 is the left child of the node (3->4, Contingency) and sentence 5 is the right child. Thus we infer the discourse relation between 3 and 5 as (3->5, Contingency).

5.4.5 Utilizing PDTB Information

Constructing the relation matrix

For both approaches, relation matrices are constructed to represent the discourse information as in Table 5.11. Extraction of features using the matrix is described in the next section. Relations already labeled by the annotator/parser are directly recorded in the matrix. Observing that the reliability of an inferred relation decreases as the number of annotated relations connecting the arguments increases, we record not only the relation types but also the **"distance"** information for the inferred relations.

[19]The similarity between merged nodes is calculated as the average of the similarity between their child leaf nodes.

Segment	1(Arg2)	2	3	4	5	6	7	8
1(Arg1)	N/A[a]	Ent	N/A[b]	Cont (2,0)[c]	Cont (2,1)	Cont (2,2)	Cont (2,3)	Cont (2,4)
2	N/A	N/A	Expan	Cont(1,0)	Cont (1,1)	Cont (1,2)	Cont (1,3)	Cont (1,4)
3	N/A	N/A	N/A	Cont	Cont (0,1)	Cont (0,2)	Cont (0,3)	Cont (0,4)
4	N/A	N/A	N/A	N/A	Expan	N/A	N/A	N/A
5	N/A	N/A	N/A	N/A	N/A	EntRel	N/A	N/A
6	N/A	N/A	N/A	N/A	N/A	N/A	Cont	Cont (1)
7	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Cont
8	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Tree	1(Arg2)	2	3	4	5	6	7	8
1(Arg1)	N/A	Ent-1	Ent-1(0,1)[d]	Ent-1(0,1)	Ent-1(0,2)	Ent-1(0,2)	Ent-1 (0,3)	Ent-1(0,3)
2	N/A	N/A	Expan-3	Cont-2(1,0)	Cont-2(1,1)	Cont-2(1,1)	Cont-2(1,2)	Cont-2(1,3)
3	N/A	N/A	N/A	Cont-2	Cont-2(0,1)	Cont-2(0,1)	Cont-2(0,2)	Cont-2(0,3)
4	N/A	N/A	N/A	N/A	Expan-4	Ent-3(1,0)	Ent-3(1,1)	Ent-3(1,2)
5	N/A	N/A	N/A	N/A	N/A	Ent-3	Ent-3(0,1)	Ent-3(0,2)
6	N/A	N/A	N/A	N/A	N/A	N/A	Cont-4	Cont-4(0,1)
7	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Cont-5(0,1)
8	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 5.11: Relation matrix constructed for the PDTBSegment approach (Upper) and the PDTBTree approach (Below). Ent is short for EntRel, Expan short for Expansion and Cont short for Contingency.

[a]Relationship between clauses within the sentence is not used in relation inference.

[b]No relations can be inferred between 1 and 3.

[c]Cont (2,0) means distance to real Arg1 is 2 and distance to real Arg2 is 0. Here the inferred relation is coming from the labeled relation (3->4, Contingency). 3 is the real Arg1 and 4 is the real Arg2. Distance to real Arg1 is 2 as the the distance between 1 and 3 is 2.

[d]Ent-1(0,1) stands for Depth 1 distance, distance=0 for Arg1 and distance = 1 for Arg2.

For the **PDTBSegment** approach, distances are recorded separately for within-segment relations and across-segment relations. For within-segment relations, the distance is recorded according to the number of sentences between Arg1 to Arg2. For example, distance for sentence 6 and 8 is recorded as 2 as there is one sentence between the two sentences. For across-segment relations, distances are recorded for both Arg1 and Arg2 according to their distances to the real Arg1/Arg2 of the across-segment relation as (Dist1, Dist2). For example, distance between sentence 1 and 5 is recorded as (2,1) as there is the distance of 2 between sentence 1 and 3 and there is the distance of 1 between sentence 4 and 5.

For the **PDTBTree** approach, we traverse up from Arg1 and Arg2 to their closest common parent node and count the distances for both arguments as (Dist1, Dist2). In Table 5.11, distance between sentence 2 and 5 is recorded as (1,1) as we back trace both nodes to their parent node (3->4). As sentence 2 is the real text span in relation node (2->3) and sentence 5 is in the node (5->6), we get distance 1 for sentence 2 as the distances between (2->3) and (3->4) in the tree is 1; similarly, we get distance 1 for sentence 5.

Extracting Features

The *PDTBSegment* and *PDTBTree* structures are constructed for both drafts as in Figure 5.5. Table 5.12 shows the PDTB features extracted for the added sentence 6 in Table 5.10, with features explained below.

Features using the labeled local PDTB information (Local). Features are extracted as the types of relations a sentence is involved with (i.e. the relation where the sentence acts as Arg1 or Arg2.) Features are extracted for sentences in both drafts. If a sentence is added or deleted, the features for the empty sentence are marked as N/A.

Features using PDTBSegment (Segment).

- *Individual features* Within each draft, the features of sentences are extracted based on the relation matrix. Similar to **Local**, we extract the discourse relation type of each sentence acting as Arg1 and Arg2[20]. Features for across-segment relations are extracted separately since the discourse relations across segments are likely to be more important than relations within segments. Weights are assigned to relations according to their distance information. A within-segment relation with distance (d1) is assigned weight

[20]The row of the sentence in the relation matrix corresponds to Arg1 and the column corresponds to Arg2.

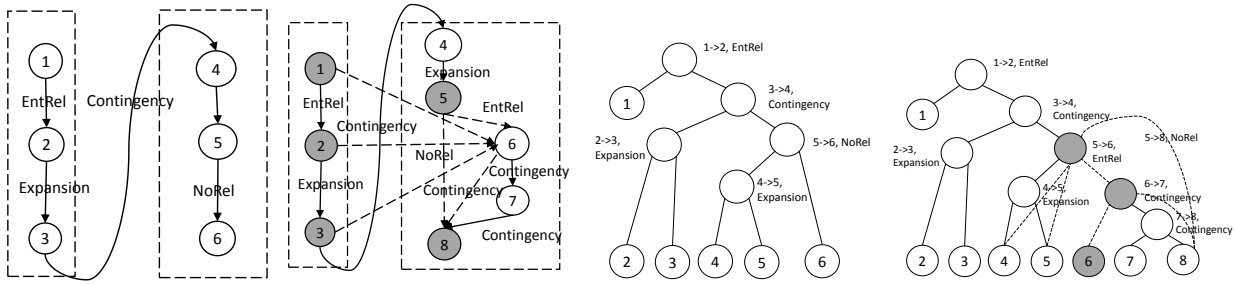
$\frac{1}{d1+1}$; an across-segment relation with distance (d1, d2) is assigned weight $\frac{1}{(d1+1)*(d2+1)}$. If a sentence is involved with multiple same-type relations, the relation with the largest weight is chosen.

- *PDTBSegment Structure change features* For these across draft features, the segment structures created for draft 1 and draft2 are compared. Nodes of segment structures are aligned according to the sentence alignment information. After comparison, the aligned nodes that are affected by the revision are selected, where the change of their related relations with the revised sentence are counted. For example in Figure 5.5(b), sentences 1, 2, 3, 5, 8 are affected by the addition of sentence 6. For sentence 1, 2, 3, sentence 6 brings addition of three across-segment relations. For sentence 5, the original "NoRel" label between sentence 5 and sentence 8 is removed. For sentence 8, relation between sentence 6 and sentence 8 is added. A vector of relation changes is thus created according to the relation matrix.

Features using PDTBTree (Tree).

- *Individual features* Features are collected in a similar manner as the **PDTBSegment** approach. To enlarge the difference of different-depth relations, weight $\frac{1}{2^{d1+d2}}$ is assigned to a relation with distance (d1, d2) .
- *Structure change features* Due to the complexity of the tree structure, only the non-leaf nodes that are directly related to the revised sentence (i.e. the sentence as Arg1 or Arg2 of the relation) are considered in the extraction of structure changes. As in Figure 5.5(d), the added sentence 6 acts as Arg2 in node (5->6) and Arg1 in node (6->7). Change of relations (4->6), (5->6) are considered as the changed relations of node (5->6). Change of relations (5->8) and (6->8) are the changed relations of node (6->7). Change vectors are calculated in similar manners as the **PDTBSegment** approach at each depth. To avoid data sparsity, the depth number is limited to 4 to reduce the number of features[21].

[21]If the depth of tree is larger than 4, the depth of the relation is still considered as 4.



(a) Segment for Draft1 (b) Segment for Draft2 (c) Tree for Draft1 (d) Tree for Draft2

Figure 5.5: The change of discourse structure from Draft 1 (D1) to Draft 2 (D2). The gray nodes are the affected nodes and the dashed lines are the affected relations. Sentences are aligned as (1->1), (2->2), (3->3), (4->4), (5->5), (6->8), (Null->6), (Null->7).

Features	**Example**
Loc	D1-Arg1[a]: N/A, D1-Arg2: N/A, D2-Arg1: Contingency, D2-Arg2: EntRel
Seg	**Individual**: WithinSegment: D1-Arg1: N/A, D1-Arg2: N/A, D2-Arg1: Contingency, D2-Arg2: EntRel, AcrossSegment: D2-Arg2: (Contingency, $\frac{1}{3}$)[b] **Structure**: WithinSegment[c]: (-1, 1, 0, 0, 0.5, 0, 0), AcrossSegment: (0, 0, 0, 0, $\frac{1}{3}$, 0, 0)
Tree	**Individual**: D1-Arg1: N/A, D1-Arg2: N/A, D2-Arg1: Contingency-4, D2-Arg2: (EntRel-1,$\frac{1}{8}$), (Contingency-2, $\frac{1}{4}$), EntRel-3 **Structure**: Depth1Vector: (0,0,0,0,0,0), Depth2Vector:(0,0,0,0,0,0), Depth3Vector: (-1,1,0,0,0,0), Depth4Vector: (0,0, 0,0,1,0,0)

Table 5.12: Examples of the features extracted for the added sentence 6 in Table 5.10.

[a]D1-Arg1 means features of sentence acting as Arg1 in the first Draft.

[b](Contingency, $\frac{1}{3}$) represents relation type Contingency with weight $\frac{1}{3}$.

[c]The columns of the change vector are (NoRel, EntRel, AltLex, Comparison, Contingency, Expansion, Temporal).

Corpus HSchool1	Base	Base+Local	Base+Segment	Base+Tree
Claim (110)	.623	.630	.640	**.668**‡*
Warrant (390)	.671	.693	**.715**‡*	.713‡*
Evidence (110)	.482	.510*	.538‡*	**.544**‡*
General (353)	.725	.732	**.756**‡*	.744*
Organization (45)	.540	.538	.544	.538
Word Usage/Clarity (84)	.738	.740	.732	.740
Conventions/Grammar/Spelling (180)	.710	.721	.733	.728

Table 5.13: Experiment 3. 10-fold (student) cross-validation. The unweighted average F-measure is reported. $*$ indicates significantly better than the baseline (paired T-test, $p<0.05$), ‡ indicates significantly better than (Base+local), **bold** indicates best.

5.4.6 Experiments and Results Using the Discourse Information Enhancement

Experiment We first repeated Experiment 3 using our new proposed feature group. We compared the results using inferred information to the baseline results, and to the results with baseline features plus each individual feature group[22]. Table 5.13 demonstrates the results.

Afterwards we repeated Experiment 4 using our new proposed feature group. The enhanced approach in Section 5.3 was used as the baseline. Table 5.14 demonstrates the results.

Analysis

According to Table 5.13, comparing to the baseline, Base+Local (using only features from the labeled PDTB relations) yields a significant improvement only when classifying *Evidence* revisions. In contrast, both Base+Segment and Base+Tree (our inference-based approaches) yield several significant improvements over the baseline[23]. Comparing to the

[22]We also experimented mixing all the features groups together but did not observe significant improvement.

[23]We also tested using just individual features (without the structure change features) and both approaches

Corpus HSchool1	Base	Base + Local	Base + Segment	Base + Tree
Precision	0.701	0.710	**0.730**∗‡	0.728∗
Recall	0.642	0.644	0.652	0.651
F-Measure	0.643	0.648	**0.678**∗‡	0.669∗

Table 5.14: Experiment 4. The average F-measure of 10-fold (student) cross-validation is reported, ∗ indicates significantly better than the baseline (paired T-test, $p<0.05$), ‡ indicates significantly better than (Base+local), **bold** indicates best.

baseline, the **PDTBSegment** approach yields significant improvement in the classification of *Warrant*, *Evidence* and *General Content* revisions and the **PDTBTree** approach yields significant improvement in the classification of all revisions except *Surface*. For the minority category *Evidence*, the **PDTBTree** approach improved F1 from 0.288 to 0.415. Comparing to the results using only labeled PDTB, the **PDTBSegment** approach yields significant improvement in the classification of *Warrant*, *Evidence* and *General*, while the **PDTBTree** approach yields significant improvement in the classification of *Claim*, *Warrant* and *Evidence* and a significant overall F1 improvement. Table 5.14 further demonstrates that the features extracted from the inferred PDTB relations can improve the performance of the contextual enhancement approach, indicating that the inferred relations introduces additional information for revision classification.

It is worthy to notice that the results reported are based on manually labeled PDTB information, and thus shows an upperbound of PDTB application. The real application of PDTB in revision classification can be influenced by the correctness of automatic PDTB recognition.

Also, while the PDTB inference approach works for the revision classification problem in this thesis, it is unclear whether this approach can be generalized for other tasks such as argument mining. We can investigate the problem from two directions 1) Applying the

still significantly outperform the baseline.

method on another task and check whether a similar improvement can be observed. 2) An intrinsic evaluation of the inferred PDTB relations, one possible way to do the evaluation is to check whether the Amazon Mechanical Turkers would agree with the inferred relation.

5.5 SUMMARY

In this chapter I describe our efforts in the automatic identification of revisions. We first investigated the automatic extraction of revisions and treated the problem as a monolingual sentence alignment problem. Afterwards we investigated the automatic classification of revisions. We first investigated the application of features and approaches used in prior studies to our problem. Three groups of features were collected: *Location*, *Textual*, *Language*. The performance is evaluated both intrinsically and extrinsically. In the intrinsic evaluation, performance was compared both on surface vs. content classification and binary classification for each individual category. Both results demonstrated significantly better performance than the unigram and majority baseline. In the extrinsic evaluation, we repeated the revision study in Chapter 3 using the number of predicted revisions. The study on predicted revisions demonstrate similar results as the manually annotated revisions. Afterwards we explored enhancing the classification performance using contextual information. Results demonstrated that by using contextual features and transforming the classification problem to a sequence labeling problem, we achieved significantly improvement over our previous approach. Finally we explored the possibility of improving classification performance using discourse relations between sentences and results demonstrated that the performance could be improved with discourse information.

The results of section 5.2 suggest the correctness of hypothesis **H2.1**. Section 5.3 suggests the correctness of hypothesis **H2.2**. Section 5.4 supports hypothesis **H2.3**.

6.0 AUTOMATIC REVISION IDENTIFICATION (JOINT)

In Chapter 5, we introduced our pipelined approach to address the problem of revision identification. One problem of the pipelined approach is that the errors of the revision extraction step are propagated to the revision classification step. To solve this problem, an approach that can conduct revision extraction and revision classification at the same time is needed. This chapter describes our solution in (Zhang and Litman, 2017).

6.1 INTRODUCTION

Table 6.1 demonstrates an example of error propagation in argumentative revision classification. According to human annotation, (D1-2) should be aligned to (D2-2), (D1-3) should be aligned to (D2-3). Based on alignment, their revision types should be *Surface*[1]. However, when the automatic sentence alignment misses the alignment, the revision classification step considers the sentences as deleted and categorizes them as *Reasoning*.

We propose a sequence labeling-based joint identification approach by incorporating the output of both tasks into one sequence. The approach is designed based on two hypotheses. **First, the classification of a revision can be improved by considering its nearby revisions.** For example, a *Claim* revision is likely to be followed by a *Reasoning* revision[2]. In (Zhang and Litman, 2016) we used the types of revisions as labels and transformed the revision classification task to a sequence labeling problem. Results demonstrated significantly better performance than SVM-based classification approaches. In this work, we extend the

[1] *Surface* include changes such as spelling correction and sentence reorderings that do not change a paper's content.

[2] If you changed the thesis/claim of your essay, you have to change the way you reason for it.

Draft 1	
(D1-1) Tone has a lot to say for Louv. **(D1-2)** On account that Louv uses words to sound completely annoyed and disgusted with how far people have drifted, says he is very disgusted and annoyed. **(D1-3)** The beginning paragraph tells that scientists can now, at will, change the colors of butterfly wings. **(D1-4)** Telling how humans are in control, at will, with nature.	
Draft 2	
(D2-1) The way Louv talks throughout the essay is his tone. **(D2-2)** Using words to sound very annoyed and completely disgusted. **(D2-3)** In the beginning of the excerpt, Louv tells of what scientists are doing now with nature, such as changing the colors of butterfly wings. **(D2-4)** Telling how humans are in control, at will, with nature.	
Gold-standard revision extraction	**Automatic revision extraction**
(D1-1, D2-1), (D1-2, D2-2), (D1-3, D2-3), (D1-4, D2-4)	(D1-1, D2-1), **(D1-2, Null), (Null, D2-2), (D1-3, Null), (Null, D2-3)**, (D1-4, D2-4)
Gold-standard revision classification	**Automatic revision classification**
(D1-1, D2-1, Modify, Surface)	(D1-1, D2-1, Modify, Surface)
(D1-2, D2-2, Modify, Surface)	**(D1-2, Null, Delete, Reasoning)** **(Null, D2-2, Add, Reasoning)**
(D1-3, D2-3, Modify, Surface)	**(D1-3, Null, Delete, Reasoning)** **(Null, D2-3, Add, Reasoning)**
(D1-4, D2-4, Nochange)	(D1-4, D2-4, Nochange)

Table 6.1: An example of pipeline revision identification errors (**Bolded**). A revision is represented as (D1-SentenceIndex, D2-SentenceIndex, RevisionOp, RevisionType) (e.g. (D1-1, D2-1, Modify, Surface)). In the example 6 revisions are identified. The revision extraction step aligns D1-2 and D1-3 wrongly as the syntactic similarities between the gold-standard sentences are not strong enough. The errors of the alignment step propagates to 4 false "Reasoning" revisions in the revision classification step.

ideas by introducing **EditSequence** to also utilize alignment information for revision type prediction. An EditSequence describes a consecutive sequence of edits where not only the revision type but also the alignment information are incorporated into the labels of the edits. We hypothesize that adding alignment information can further improve revision type prediction. **Second, the alignment of sentences can be corrected according to the types of labeled revisions.** For example, the predicted types in Table 6.1 as a whole are rare, as there are 2 deleted *Reasoning* sentences and 2 added *Reasoning* sentences without any *Claim* change. Such a sequence is likely to have a small likelihood in sequence labeling and thus a possible alignment error is detected. We introduce the idea of "mutation" from genetic algorithms to generate possible corrections of sentence alignments. The alignment of sentences after correction allows us to conduct a new round of revision type labeling. Our approach iteratively mutate and label sequences until we cannot find sequences with larger likelihood. Two approaches are utilized to generate seed sequences for mutation: (1) Direct transformation from predicted sentence alignment (Zhang and Litman, 2014) (2) Automatic sequence generation using a Recurrent Neural Network (RNN). These settings together allow us to achieve better performance for both revision extraction and revision classification.

6.2 RELATED WORKS

The idea of using sequence labeling for revision identification derives from the work in (Zhang and Litman, 2016), where the types of revisions used as labels. Revisions are transformed to a sequence of labels according to the gold-standard alignment information. In this section, the sentence alignment step is also included as a target of our identification[3]. We extend our prior work by grouping sentence alignment and revision type together into one label for joint identification.

As our tasks involve alignment, the problem in this chapter can look similar to a labeled alignment problem, which can be solved with approaches such as CRFs (Blunsom and Cohn, 2006) or structured perceptrons/SVMs (Moore et al., 2006). For example, Blunsom and Cohn

[3]As the models in this chapter are trained at the paragraph level, we assume the paragraphs were aligned

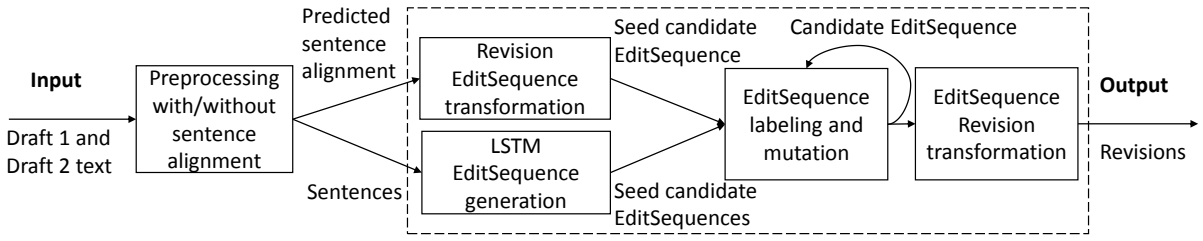


Figure 6.1: Overall approach architecture. Components within the dashed box are covered in this chapter. Notice that sentence alignment in the preprocessing step can be skipped with LSTM sequence generation.

(2006) utilized CRFs to induce word alignment between bilingual sentence pairs. In their work, each sentence in the source document is treated as a sequence. Sequence labeling is conducted on the source sentence and the index of the aligned word in the target sentence is used as the label. Features such as translation scores between words are used and the Viterbi algorithm is used to find the maximum posterior probability alignment for test sentences. Our problem is more complicated as our labels cover both the alignment and the revision type information. In labeled alignment, labels are used to represent the alignment information itself in **one** sequence. In revision identification, labels are used to represent the **interaction between two sequences** (the difference between sentences). Thus, our work utilized the revision operation (add/delete/modify) instead of the sentence index to mark the alignment information. Such design allows us to have the location information better coupled with the revision type information, and meanwhile allows us to update the alignment prediction by simply mutating the revision operation part of the labels.

The idea of sequence mutation is introduced from genetic algorithms to generate possible sentence alignment corrections. There are works on tagging problems (Araujo, 2002; Alba et al., 2006; Silva et al., 2013) where genetic algorithms are applied to learn a best labeling or rules for labeling. However, our approach does not follow the standard genetic algorithm in that we do not have crossover operations and we stop mutating when the current generation is worse than last. The idea behind our seed generation approach is similar

to Sequential Monte-Carlo (Particle-filter) (Khan et al., 2004), where the sequence samples are generated by sampling labels according to their previous labels. In the chapter we utilize a Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) RNN to generate sample sequences as seeds. The advantage of LSTM is that it can utilize long distance label information instead of just the label before.

6.3 APPROACH DESCRIPTION

6.3.1 Approach Overview

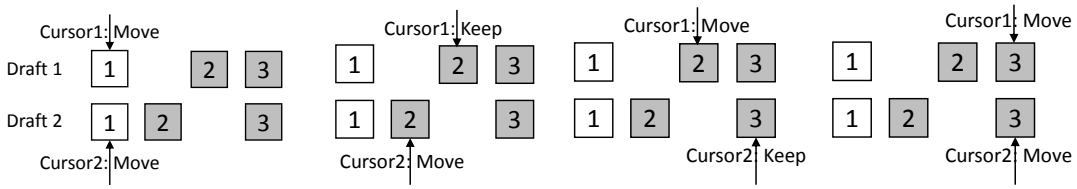
Figure 6.1 demonstrates the workflow of our approach. The sentence alignment approach in (Zhang and Litman, 2014) is first utilized to segment the essays into sentences and generate a sentence alignment prediction. Afterwards seed EditSequences are generated either using a LSTM network or by transforming directly from the predicted sentence alignment. The seed EditSequences are then labeled by the trained sequence labeling model. The candidate EditSequences are mutated according to the output of the sequence model. Finally the best EditSequence is chosen and transformed to the list of revisions.

6.3.2 Transformation between Revision and EditSequence Representation

Instead of using the sentence indices as the alignment information as in other works (Blunsom and Cohn, 2006), we propose **EditSequence** as a sequence representation of revisions. It incorporates both the alignment information and the revision type information in one sequence[4].

EditStep is defined as the basic unit of an **EditSequence**. An EditSequence contains a consecutive sequence of EditSteps. An EditStep unit contains 3 elements $(Op1, Op2, RevType)$. For a pair of revised essays (Draft1, Draft2), a cursor is created for each draft separately and we define $D1Pos, D2Pos$ to record cursor locations. $Op1$ and $Op2$ record the **action**s

[4]Following (Zhang and Litman, 2016), we treat a revision that reorders two sentences as a Delete and an Add revisions.



(a) M-M-Nochange (b) K-M-Reasoning (c) M-K-Reasoning (d) M-M-Surface

Figure 6.2: Example of EditSequence transformation. The first row represents the sentences of the original essay (Draft1) and the second row represents the sentences of the revised essay (Draft2). The vertical direction indicates sentence alignment. The shadowed sentences are revised and there are three revisions: (Null, 2, Add, Reasoning), (2, Null, Delete, Reasoning) and (3, 3, Modify, Surface). With the cursors, we transform the revisions to 4 consecutive EditSteps from left to right and generate the sequence representation (M-M-Nochange -> K-M-Reasoning -> M-K-Reasoning -> M-M-Surface).

of the cursors. There are two cursor actions: **Move** (M) and **Keep** (K). **Move** indicates that the corresponding cursor is going to move to the position of the next sentence while **Keep** indicates that the cursor remains at the same location. $RevType$ records the revision type information. Following (Zhang and Litman, 2016), revision types include five types[5] for sentences changed[6] and one type *Nochange* when aligned sentences are identical.

Revisions to EditSequence. Figure 6.2 demonstrates how we transform from the revision representation used in prior works to our sequential representation EditSequence. In Figure 6.2(a), the cursors of the two drafts start at the beginning of the segment with $D1Pos$ and $D2Pos$ set to 1. Given that sentence 1 in Draft1 is the same as sentence 1 in Draft 2, both cursors move to the next sentence and we generate an EditStep (M, M, Nochange). In Figure 6.2(b), $D1Pos$ and $D2Pos$ are set to 2 according to the action of the previous step. In the example, sentence 2 in Draft 2 is an added *Reasoning* sentence, thus we generate a new EditStep (K-M-Reasoning) by keeping the cursor of Draft 1 in its current

[5] *Claim/Ideas (Claim)*, *Warrant/Reasoning/Backing (Reasoning)*, *Evidence*, *General Content (General)* and *Surface*

[6] Added/Deleted/Modified

position (for comparison at the next step) and moving the cursor of Draft 2. Similarly, we move the cursor of Draft 1 in Figure 6.2(c). In Figure 6.2(d), $D1Pos$ and $D2Pos$ are set to 3. Sentences at the position are aligned to each other and both cursors are thus moved. Each **EditStep** is assigned a label as $Op1$-$Op2$-$RevType$ and thus we generate a labeled sequence representation of revisions. As there are only three possible Op combinations (M-M, K-M, M-K)[7], the total number of possible labels is $3 \times RevisionClassNum$.

EditSequence to Revisions. The sequence transformation step is reversible and we can infer all the revisions according to the sequence of edits. Head of the EditStep label indicates the revision location: a label starting with "M-M" indicates that two sentences are aligned, "M-K" indicates that a sentence is deleted while "K-M" indicates that a sentence is added. Tail of the label corresponds to the revision type.

6.3.3 EditSequence Labeling and EditSequence Mutation

For our first hypothesis, we conduct sequence labeling on EditSequence and use $RevType$ of the labeled sequence as the results of revision classification. **For our second hypothesis**, we utilize both the likelihood provided by the sequence labeler and the ($Op1$,$Op2$) information of labels to correct sentence alignments.

Given a candidate EditSequence, sequence labeling is conducted to assign labels to each EditStep in the sequence. The $RevType$ part of the assigned label is used as the revision type. Conditional Random Fields (CRFs) (Lafferty et al., 2001) is utilized for labeling[8]. Features used in (Zhang and Litman, 2015) are reused, which include unigrams and three feature groups.

Location group. For each EditStep, we record its corresponding $D1Pos$ and $D2Pos$ as features, We also record whether the $D1Pos$ and $D2Pos$ are at the beginning/end of the paragraph/essay.

Textual group. For each EditStep, we extract features for the aligned sentences pair ($D1Pos$, $D2Pos$). Features include sentence length (in both drafts), edit distance between aligned sentences and the difference in sentence length and punctuation. We not only cal-

[7] At least one of the cursors has to move.
[8] CRFSuite (Okazaki, 2007) is used in implementation.

culate the edit distance between sentence pair ($D1Pos$, $D2Pos$) but also for pairs ($D1Pos$, $D2Pos$+1)[9] and ($D1Pos$+1, $D2Pos$).

Language group. Part of speech (POS) unigrams and difference in POS counts are encoded. Again features are extracted for pairs ($D1Pos$, $D2Pos$+1) and ($D1Pos$+1, $D2Pos$) besides ($D1Pos$, $D2Pos$).

Besides assigning labels to the sequence, the CRFs model also provides us the likelihood of each label and the likelihood of the whole sequence. We compare the likelihood between sequences to decide which sequence is a better labeling. Within one sequence, we compare the likelihood between EditSteps to decide which EditStep is most likely to be corrected. Besides using the likelihood of each EditStep, we also compare the ($Op1$, $Op2$) information with the ($Op1$, $Op2$) information of the prior candidate EditSequence. We call it **collision** when such information does not match, which indicates that the candidate's alignment does not follow a typical sequence pattern and suggests correction.

We borrow the idea of "mutation" from genetic algorithms to generate possible corrections of sentence alignment. There are three possible kinds of "mutation" operations. **(1) "M-M" to "M-K" or "M-M" to "K-M"**. This indicates breaking an alignment of sentences to one *Delete* revision and one *Add* revision. Thus for a EditStep with tag "M-M-Type", we would remove the step from the sequence and add two new steps "M-K-Nochange" and "K-M-Nochange". Notice that here *Nochange* is a dummy label and will be replaced in the next round of labeling. **(2) "M-K" to "M-M" or "K-M" to "M-M"**. This indicates aligning a deleted/added sentence to another sentence. Depending on the labeling of the following EditStep, the operation can be different. "M-K" followed by "K-M"[10] indicates that the aligned sentence in Draft 2 is not aligned to other sentences. For example in Figure 6.3, the second EditStep (M-K-Nochange) is followed by EditStep (K-M-Nochange), which indicates that Sentence 2 (Draft 2) has not been aligned to other sentences and aligning sentence 2 (Draft 1) will not impact the alignment of Sentence 2 (Draft 2). In that case, we remove the two steps and add a step "M-M-Nochange". "M-K" followed by "M-M" indicates that the aligned sentence has been aligned to other sentences. For that

[9] If D2Pos+1 does not exceed paragraph boundary

[10] Or "K-M" is followed by "M-K".

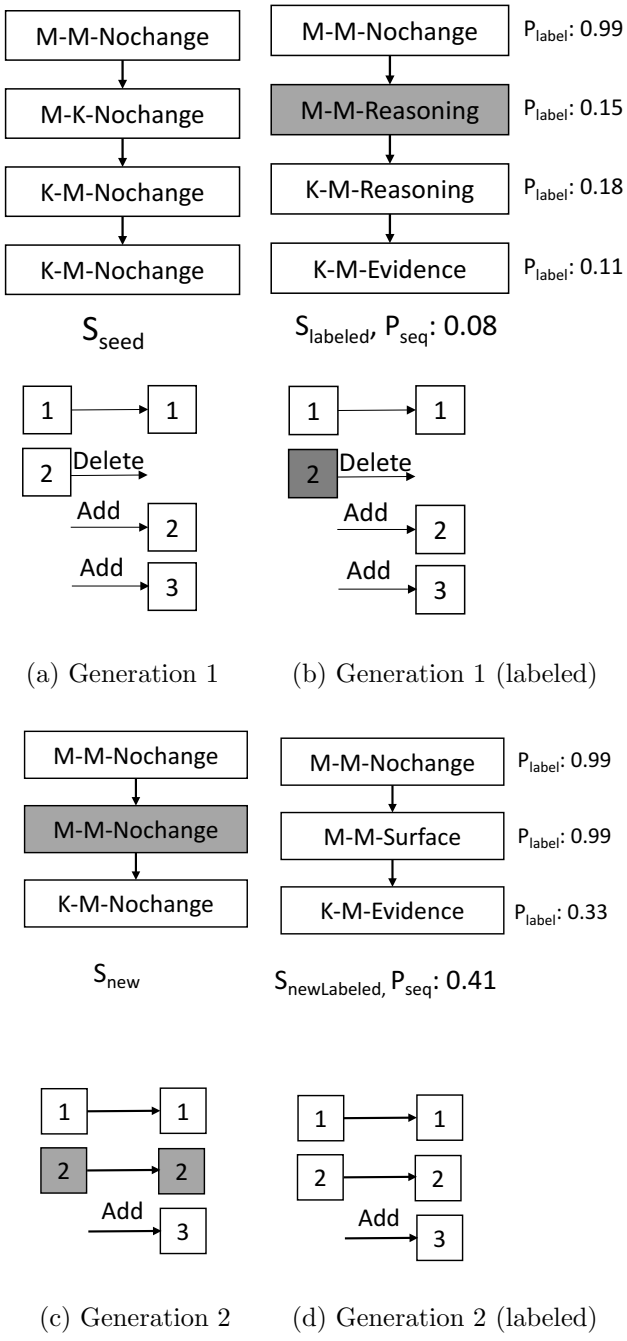


Figure 6.3: Example of EditSequence update. Two EditSequences can be mutated from $S_{labeled}$: one from the EditStep with collision (the shadowed EditStep) and one from the EditStep with the lowest likelihood (the last EditStep). The first generation (seed sequences) will always be mutated, while the other generations will only mutate if they have a larger likelihood than the prior generation. Note that only $RevType$ in labeled sequences ($S_{labeled}$ or $S_{newLabeled}$) will be used as the type of revisions.

case, we need to remove the "M-K" and "M-M" step and add two steps "M-M-Nochange" and "M-K-Nochange" for the misaligned sentence. **(3) "M-K" to "K-M" or "K-M" to "M-K"**. This means changing from *Delete* to *Add*. This is similar to the previous case, where the mutation operation depends on the labeling of the following EditStep. If the following EditStep starts with "M-M", it indicates that the sentence in the *Add* revision is aligned and we need to break the existing alignments and add a "M-K" EditStep besides changing "M-K" to "K-M".

Figure 6.3 provides an example of the EditSequence update process. The process starts with seed candidate sequences as the first generation (Figure 6.3(a)), the first generation will always be mutated (Figure 6.3(b)). For a seed EditSequence S_{seed} and its labeled sequence $S_{labeled}$, the alignment part of their EditStep labels are compared to check for collision. For every collision detected, we mutate S_{seed} to generate one new candidate sequence S_{new} as a member of the next generation (Figure 6.3(c) shows one mutation). After the mutation of the first generation is complete, all S_{new} in the new generation are labeled with CRFs again. The new labels provide us new revision types within the new alignments (Figure 6.3(d)). If the likelihood of the labeled sequence $S_{newLabeled}$ is larger than $S_{labeled}$, it indicates that the sentence alignment in S_{new} is more trustworthy than the alignment in S_{seed}, thus S_{new} should be further mutated to see if the alignment can be further improved. Otherwise we do not further mutate S_{new}. We keep mutating the EditSequences until we cannot conduct any further mutation. For the labeled EditSequences in all generations, we first select sequences with minimum number of collisions and then select the sequence with the maximum sequence likelihood. The ($Op1$, $Op2$) of labels are used as results of revision extraction and $RevTypes$ are used for revision classification. Through the process, sequence labeling provides likelihood for both alignments and revision types, while sequence mutation provides new possible sequences for labeling.

Groups	Model	Revision extraction (sentence alignment)	Revision classification
Baseline (Base)	Pipeline	Based on sentence similarity (Zhang and Litman, 2014)	CRF sequence labeling, using revision type as label (Zhang and Litman, 2016)
1Best	Joint	(Zhang and Litman, 2014) + EditSequence mutation	CRF sequence labeling, using both revision type and alignment as label
+NCandidate (+NC)	Joint	(Zhang and Litman, 2014) + LSTM EditSequence generation + EditSequence mutation	CRF sequence labeling, using both revision type and alignment as label

Table 6.2: Description of three implemented approaches

6.3.4 Seed Candidate EditSequence Generation

For a paragraph with m sentences in the first draft and n sentences in the second draft, there is a total of $\binom{m+n}{n} = \frac{(m+n)!}{m!n!}$ possible sequences[11]. While theoretically we can first generate a sequence without sentence alignment (all sentences in Draft 1 treated as deleted and all sentences in Draft 2 treated as added) as the seed sequence and keep mutating until the best sequence is found, such process is too computationally expensive and is likely to fall into local optima during mutation. Thus an approach is needed for the generation of high-likelihood seed EditSequences. We propose two approaches for sequence generation, one based on the revision extraction method proposed in (Zhang and Litman, 2014), the other based on automatic sequence generation with LSTM.

1-Best EditSequence generation based on alignment prediction During preprocessing, the essays are segmented into sentences and sentences are aligned following (Zhang and Litman, 2014). A logistic regression classifier is first trained on the training data with Levenshtein distance as the feature and alignment is conducted using Nelken's global alignment approach (Nelken and Shieber, 2006) based on the likelihood provided by the classifier. As the number of essays in the dataset is limited, we construct sequences at the paragraph level. We trained our models on paired paragraphs assuming paragraphs have been aligned.

[11]With m sentences of Draft 1 set, there are m+n slots to put in the n sentences of Draft 2.

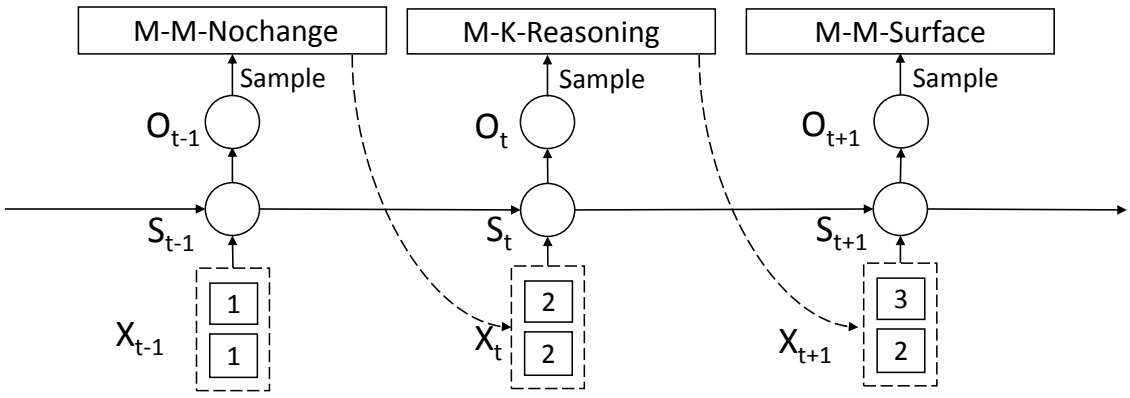


Figure 6.4: LSTM recurrent neural network for generating candidate sequences. X are features extracted according to the location of the cursors. For example, X_{t-1} corresponds to features extracted when sentence index in Draft 1 is 1 and sentence index in Draft 2 is 1.

For each paragraph pair, an EditSequence is generated following the sequence transformation method with *RevType* of all EditSteps set to *Nochange*[12].

N-Candidate EditSequence generation with LSTM network The 1-best approach can provide a good sequence to start with, however, it is more likely to fall into local optima in the labeling step with only one seed candidate. Thus we also trained LSTM to generate multiple possible candidates. As demonstrated in Figure 6.4, we construct the neural network with LSTM units. Due to the size limit of our current training data, we only include one layer of LSTM units to reduce the number of parameters in the network. Each EditStep is treated as a time step in the neural network. According to the $D1Pos$ and $D2Pos$ property of the EditStep, we extract features X as the input to the neural network. The same set of features used in the sequence labeling step is used. The model transforms the input to hidden state S, where hidden state S_{t-1} at time (t-1) is used together with input X_t to predict the hidden state S_t at time t. A softmax layer is added on the top of the hidden state to predict O_t, which describes the probability distribution of the sequence labels. At the training step, we fit the model with EditSequences transformed from revisions between the paragraphs.

[12] *Nochange* is a just a placeholder as the real *RevType* are to be labeled in the labeling step.

At the generation step we start with both $D1Pos$ and $D2Pos$ set to 1 and extract features for X_1. In each time step, a label is sampled according to the probability distribution O_t. According to the sampled label, we change the positions of $D1Pos$ and $D2Pos$ to extract the features for X_2. In the example, the sampled label at X_{t-1} is M-M-Nochange, this label moves $D1Pos$ and $D2Pos$ to (2,2) and the X_t is extracted and used together with S_{t-1} to predict S_t. According to the probability distribution O_t, a new label is sampled and the result is used to move the cursors for the next EditStep. The process is repeated until an EditSequence is generated for the whole paragraph pair. We repeat the algorithm until N candidates are collected.

6.4 EXPERIMENTS AND RESULTS

Experiments are conducted using different revision type settings. In addition we group *Warrant/Reasoning/Backing*, *Evidence* and *General Content* together as one *Support* category[13]. We first evaluate the performance of sentence alignment and *Content* vs. *Surface* vs. *Nochange* revision classification (**3-class**). Then we experimented with *Claim* vs. *Support* vs. *Surface* vs. *Nochange* (**4-class**). Finally we used all revision categories (**6-class**). For each experiment, three approaches are compared as in Table 6.2: **Baseline**, **1Best** and **+NCandidate**. 10 draft pairs from Corpus *HSchool2* were used as the development set for setting up parameters of LSTM[14] and choosing N. N is set to 10 for all our experiments. Afterwards 10-fold (student) cross-validation were conducted on corpora *HSchool1*, *HSchool2* (without the development set) and *ArgRewrite*. The same set of data folds and features were used for all three approaches. The training folds in each round will be used for training both CRFs and LSTM. For evaluation we used alignment accuracy[15] to measure the accuracy of **revision extraction** and precision/recall to measure the result of **revision identification**. Precision is calculated as $\frac{\#CorrectRevisions}{\#PredictedRevisions}$ and Recall is calculated as $\frac{\#CorrectRevisions}{\#GoldStandardRevisions}$.

[13] Content revisions that support the claim of the essay.

[14] LSTM implemented with deeplearning4j (`http://deeplearning4j.org`) with epoch set to 1, iteration numbers to 100 and output dimension of the first layer to 100

[15] $\frac{2\times AgreedAlignment}{\#Draft1Sentences+\#Draft2Sentences}$, adapted from Zhang and Litman (2014).

			Extraction	**Classification**	
			Accuracy	**Prec**	**Recall**
3-class	*HSchool1*	Base	0.940	0.780	0.830
		1Best	0.948∗	0.801∗	0.859∗
		+NC	**0.957**∗‡	**0.815**∗‡	**0.875**∗‡
	HSchool2	Base	0.928	0.780	0.834
		1Best	0.930	0.782	0.840
		+NC	**0.934**	**0.788**	**0.848**‡
	ArgRewrite	Base	0.933	0.524	0.540
		1Best	0.960∗	0.572∗	0.561∗
		+NC	**0.964**∗	**0.581**∗	**0.580**∗‡
4-class	*HSchool1*	Base	**0.940**	0.647	0.685
		1Best	0.937	0.648	0.703∗
		+NC	**0.940**	**0.652**	**0.723**∗‡
	HSchool2	Base	0.928	0.595	0.627
		1Best	0.935∗	0.620∗	0.654∗
		+NC	**0.944**∗‡	**0.647**∗‡	**0.702**∗‡
	ArgRewrite	Base	0.933	0.605	0.641
		1Best	0.960∗	0.698∗	0.768∗
		+NC	**0.968**∗‡	**0.717**∗‡	**0.789**∗‡
6-class	*Hschool1*	Base	0.940	0.397	0.376
		1Best	0.940	0.411∗	0.390∗
		+NC	**0.948**∗	**0.427**∗‡	**0.406**∗‡
	HSchool2	Base	0.928	**0.400**	**0.344**
		1Best	0.930	0.393	0.339
		+NC	**0.936**	0.390	0.338
	ArgRewrite	Base	0.933	**0.565**	**0.544**
		1Best	**0.960**∗	0.442⋄	0.433⋄
		+NC	0.959∗	0.440⋄	0.424⋄

Table 6.3: The average of 10-fold (student) cross-validation results on Corpora *HSchool1*, *HSchool2* and *ArgRewrite*. Alignment accuracy, Unweighted average precision/recall are reported. ∗ indicates significantly better than the baseline, ‡ indicates significantly better than 1Best (Paired T-test, $p < 0.05$), ⋄ indicates significantly worse than Base. **Bold** indicates best result.

Table 6.3 demonstrates our experimental results. We first compare the pipeline baseline with our joint model using 1Best seed EditSequence. With 3 revision types (3-class), the joint model achieves significantly better performance than the baseline on Corpora *HSchool1* and *ArgRewrite* for both revision extraction (sentence alignment) and revision classification. It also shows better performance on Corpus *HSchool2* (while not significant). The improvement on the precision/recall of revision classification supports our first hypothesis that alignment information can improve the accuracy of revision classification. The improvement on sentence alignment supports our second hypothesis that the patterns of predicted revisions can be used to correct the false alignments. We notice that the number of revision types impacts the performance of the model. On corpus *HSchool1*, the model shows significantly better performance than the baseline in almost all experiments. While on corpora *HSchool2* and *ArgRewrite*, the model yields significantly better performance in 4-class experiment. The impact of revision types on our model can be two-fold. On the one hand, more revision types indicates more detailed sequence information, which improves the chance of recognizing problems in sentence alignment. On the other hand, the increase of revision types increases the difficulty of sequence labeling, which in return can hurt the performance of joint identification. We leave the error analysis of performance difference between different revision types to the future work.

Next, we compare results using **1Best** and **+NCandidate** EditSequences. We observe that using generated sequences improves the 1Best performance, yielding the best result on almost all experiments (except on Corpora *HSchool2* and *ArgRewrite* with 6 revision types). We counted the number of generations in EditSequence mutation for both 1Best and +NCandidate on 3-class experiment. Results show that the 1Best approach will stop mutating after an average of 1.2 generations while +NCandidate stops mutating after an average of 2.3. This suggests that our approach prevents the model from easily falling into local optima.

6.5 SUMMARY

In this chapter a joint identification approach for argumentative writing revisions is described. For the two different sub tasks of revision identification (revision location extraction and revision type classification), we transform the location representation to a revision operation format and incorporate it together with the revision type into one label. The two different tasks are thus transformed to one joint sequence labeling task. With this design, the likelihood of a labeled sequence indicates not only the likelihood of sentence alignments but also the likelihood of the revision types. We utilize the mutation idea from genetic algorithms to iteratively update the labeling of sequences. LSTM is utilized to generate seed candidate EditSequences for mutation. Results demonstrate that our approach improves the performance of both tasks.

Experimental results suggest the correctness of our hypothesis **H2.4**, showing that we can improve the performance of revision location extraction and revision type classification by combining these two tasks together and predict them jointly using sequential models.

7.0 FUTURE DIRECTIONS

There are several remaining research questions to be addressed in the future. In the previous chapters, we discussed the two major issues addressed in this thesis. In this chapter, we discuss the possible future works on these two issues. Also, we describe the possible future works on the building of an intelligent revision assistant.

7.1 SCHEMA AND CORPORA COLLECTION

7.1.1 Expanding the Schema

Argumentation plays an important role in analyzing many types of writing such as persuasive essays (Stab et al., 2014), scientific papers (Teufel, 2000) and law documents (Palau and Moens, 2009). In Chapter 2.3 we discussed about the expansion of the revision schema to scientific writings, where we included a *Precision* category to cover the revisions that make the statement more precise. While the preliminary study on 9 scientific reports looks promising, it is uncertain whether more categories are needed for other types of scientific writings. The expansion of the schema on this corpus requires the collection of more complete and advanced scientific writings such as scientific paper writing. Similarly, it is also important to study how to extend the schema for the law document revisions.

Meanwhile, it is not clear whether the current schema can also capture the salient features of writing improvement in other kinds of writings. The study on the revision schema for academic paper writings could be an important extension of this thesis's works.

7.1.2 Collecting the Quality of Revisions

While the works in this paper addresses the problem of "what are the revisions?", the problem of "are these revisions useful?" has not been addressed. In Chapter 4 we have shown that the users tend to make changes if they don't agree with the revision type recognized by the system. Similarly, we can hypothesize that users would try to improve their changes if the system can provide feedback to the quality of revisions.

Annotation of the revision quality can be done from different scales. The annotation of *Surface* revision quality can be annotated at the sentence level, where we can compare whether the revised sentence looks better than the previous sentence. Tan and Lee (2014) has created a sentence-level statement strength comparison corpus via Amazon Mechanical Turk, where the Turkers annotated whether one sentence is having a stronger statement strength than the other. Similar approaches could be applied for the annotation of *Surface* revision quality. However, the annotation of *Content* revision quality would require more contextual information. One solution can be the annotation of revision quality within a paragraph. Each revision within the paragraph can be annotated based on the context of the paragraph. A new schema should be proposed to cover the possible types of paragraph-level improvement by a sentence-level revision.

7.1.3 Connecting Revisions to Reviews

Feedback has also been shown to be helpful for students' writing improvement (Cho and Schunn, 2007). We hypothesize that it is possible to improve the helpfulness of the revision assistant tool if it connects the user's revisions to the reviews received. In another project on corpus *HSchool2*, the implementation of reviews were annotated as *Praise*, *Implemented* and *Not implemented* as in Table 7.1. This allows us to create a corpus for the study on review revision connection.

For review linking annotation, we can first annotate the property of reviews and then connects the annotated reviews to the revisions.

- Review Unit. Each review unit corresponds to one issue of the author's essay. One review can contain multiple review units. Each review unit has one review type and contains 0

Review	**Aspect**	**Implementation**
"Your thesis completes the task entirely. You clearly stated what Kelley was trying to portray in her speech and showed how she accomplished this through the usage of rhetorical devices. It was concise and easy to follow. "	Thesis	Praise
"I think paragraph one would be more effective if it began with a topic sentence that included the rhetorical device in it (repetition), giving the reader a clear idea about what the paragraph will discuss."	Organization	Not implemented
"You should always end your quote with a citation to the text so that the reader can always look back to the text and see where the quotation came from"	Writing Style	Implemented

Table 7.1: The reviewers leave their comments on specified aspects: *thesis*, *rhetorical strategies*, *textual evidence*, *explanations*, *organization* and *writing style and standard English*. If a review contains only praises, it is marked as *Praise*; otherwise the annotators examined the revised essay to decide whether the problem pointed out in the review is implemented in the revision or not.

to multiple review targets and solutions.

- Review Type. Intuitively, the purpose of a revision should correspond to the problem type pointed out by the review. Thus we use the categories defined in our revision annotation schema for the reviews. The categories include: *Claims/Ideas*, *Warrant/Reasoning/Backing*, *Evidence*, *Rebuttal/Reservation*, *General Content*, *Word-Usage/Clarity*, *Organization*, *Conventions/Grammar/Spelling* and *Precision*. Besides annotation, the other solution is to use the meta information of review aspect directly (Thesis, Evidence, etc.).
- Review Target. The annotator is required to mark out the specific text segment that describes the location of the problem. If the review does not target on a specific problem of the essay, the attribute "IsGlobal" will be marked as "Yes", otherwise it will be marked as "No". The annotation of the review target will help the automatic systems discover the patterns for target extraction.
- Review Solution. The annotator is required to mark out the text segment of the solution suggested by the reviewer if there is solution suggested. Similar to review target, the marking of review solution would also help the systems develop the patterns for solution extraction. The extracted information would also be used for the matching of reviews and revisions. The annotator can leave the annotation empty is the review unit does not contain any solutions.

7.1.4 Expanding the Corpus Annotation

Besides the expansion of the revision schema, it can also be helpful to the NLP community to expand the corpus annotation. The expansion can involve works from three perspectives.

- Increase the size of the corpus. The size of the current corpus still limits the application of more advanced computation models. For example, the deep learning model proposed in Chapter 6 could potentially achieve better performance with more training data.
- Increase the drafts of essays written by each user. The ArgRewrite study in Chapter 4 collects three drafts from each user, which allows us to analyze the possible influence of

ArgRewrite by comparing two versions of revisions. Adding more drafts for each user would further allow us to observe a longer-term influence of the revision assistant tools.

- Increase the annotations on each single draft. The current annotations only contain the information of revision location and revision type. The annotation of other information on the corpus would allow us to conduct more advanced study for revision identification enhancement. For example, the annotation of PDTB information on corpus *Hschool1* allow us to improve the classification results with discourse information in Chapter 5.4. Similar researches can be conducted with the annotation of other information such as discourse roles(Burstein et al., 2003).

7.2 AUTOMATIC REVISION IDENTIFICATION

7.2.1 Revision Identification for Essays with Frequent Structure Changes

In Chapter 5 and Chapter 6 multiple approaches were proposed to take advantage of the context information. However, there were multiple assumptions made for the purpose of simplification. It is important to address these problems in the future.

First of all, the sequential approaches are shown to be helpful for the performance improvement. However, all the sequential approaches are built based on the simplification of removing structure changes (e.g. switching the locations of paragraphs). While it is acceptable in our dataset where the structure changes rarely happens, such assumption might not hold for more advanced writings such as academic paper writing.

Second, in Chapter 5.4, the manually labeled PDTB relation were utilized for improving revision classification performance. However, the state-of-art PDTB parser (Lin et al., 2014) has not demonstrated satisfactory performance in identifying PDTB relations. It is necessary to study whether it is still possible to improve the classification performance with noisy discourse parsing output.

Third, the joint approach in Chapter 6 made an assumption that the paragraphs have been aligned. However, in real applications, the errors in the paragraph alignment step can

be propagated to the revision identification step. There exists two possible directions to solve this problem: 1) The development of a highly accurate paragraph aligning algorithm 2) An approach that can utilized the results in the revision identification step to correct possible paragraph alignment errors.

7.2.2 Error Analysis for Revision Identification

It is important to mention that the automatic revision identification results in this thesis is still far from satisfactory. Thus, it is important to analyze and understand the difficulties of this problem. There are multiple works that need to be done:

- Automatic identification on other corpora. The corpora used in this thesis are essays are written by high school/university students. Those essays are less likely to be well-organized, thus it is also important to understand how well the algorithms work on a better organized essay. For example, we can investigate whether we can observe a good performance on the published writings such as Wall Street Journals.
- Detailed error analysis for each category. While the difficulty in revision identification can come from bad writings, it is also possible that not the correct features were used in our task. One possible future work is to analyze the most commonly mistakes made by the current recognizer and design specific rules for those mistakes.
- Understanding the models. In Chapter 6 we described a complicated joint model involving multiple components. However, we have not studied the role of each component. For example, is the "mutation" step or the "sequence labeling" step the essential component for writing improvement? Ablation test can be done here to understand the effectiveness of these approaches.

7.2.3 Automatic Revision Scoring

This task would be possible with the collection of revision quality data. I hypothesize that this problem would be similar to the current works on essay scoring. A list of possible questions to be asked are:

- Can we utilize the features from essay scoring for the revision quality scoring tasks?
- What is the best level (sentence or paragraph or essay) for us to conduct automatic revision quality evaluation?
- How do we utilize contextual information for this task?

7.3 BUILDING INTELLIGENT REVISION ASSISTANT

7.3.1 Improving the User Interface Design

The user interface design of the future revision assistant could be more complicated. In Chapter 4 we created **RevisionMap** to help users quickly locate the revisions they have made. With the works on the new directions, there could be new problems on the design of such assistant. Here is a list of possible questions:

- How to show the important revisions (E.g. highlight the ineffective revisions)?
- How to show revisions at different levels? (Paragraph level, sentence level or even phrase level)
- How to show the connection of revisions to reviews?

7.3.2 Study with a Fully Automated System

In Chapter 4.2, we described our study on the effectiveness of ArgRewrite with all the revisions manually corrected. However, we have not tested whether the system can still influence the user's rewritings with all the revisions automatically recognized. For the study with a fully automated ArgRewrite, a user can iteratively revise his essay with the instant feedback from ArgRewrite. This experiment would allow us to have a more accurate estimate of the effectiveness of our revision assistant system.

7.3.3 More Comprehensive User Study

In Chapter 4.2, we investigated the impact of the language factor on the user's rewriting behavior. However, there are other factors that might influence the user's writings. A more comprehensive study can be done in the future to understand the impacts of these factors. For example, there might exists correlation between the users' rewriting and their education level (undergrad vs. graduate). A user's major (social science vs. natural science) can also influence the user's writing behaviors. Besides, the user's own writing skills (how many drafts do they typically write for one essay, whether they are confident in their writings, etc.) can also impact the way they rewrite.

7.4 SUMMARY

In this chapter we listed the possible future works towards the building of an intelligent revision assistant. On the basis of automatic revision identification, the tool can be made more useful by constructing a system that can 1) cover more genres of writings 2) identifying the quality of the revisions 3) Link reviews to the revisions made. Works on these directions could contribute to researches both in the education field and the NLP field.

8.0 SUMMARY

This thesis present my works towards the building of intelligent revision assistant. The developed techniques are expected to provide automatic feedback to students on the purpose of the author's revision. Two major issues are resolved.

First, a sentence-level argumentative revision schema is developed. The schema describes the revisions at the level of sentence and categorizes the revisions according to the author's purpose. Based on the schema, several corpora were annotated. The analysis on the corpora annotation demonstrate that the schema can be reliably annotated by human. Also, statistical analysis on the annotated corpora indicates that the schema can capture salient characteristics of writing improvement. Results indicate that there is a significant correlation between the number of revisions and the writing improvement. I also show that it is possible to generalize our framework to scientific report writings. Based on the proposed schema, a prototype revision assistant is developed based on the collected corpora and developed algorithms. A user study on the effectiveness of a wizard-of-oz revision assistant is investigated, where both Native and ESL speakers are recruited. It is demonstrated that users tend to make more changes when they found that their intended revision was not recognized. Also, we found that the Native and ESL speakers are impacted differently by the revision feedback they received.

Second, approaches for the automatic identification of revisions are developed, including both the identification of revision location (revision extraction) and the classification of revision types (revision classification). The extraction problem is treated as a monolingual sentence alignment problem and high accuracy is achieved on the tested corpora. For the classification problem, three groups of features (*Location*, *Textual*, *Language*) are investigated for the classification task.Then the classification performance is enhanced by using contextual

features and transforming the classification problem to a sequence labeling problem. We also found that the classification performance can be improved by utilizing PDTB discourse information. We also investigated a joint approach that identifies the location and the type of revisions at the same time. Experiment results indicate that this approach can improve the performance of both tasks.

The works presented above have provided support for all of our hypotheses. We demonstrate that the schema proposed can be reliably annotated by human and captures salient features of writing improvement. Based on the schema, we collected multiple corpora. Multiple approaches are investigated to improve the accuracy of automatic revision identification. Our user study on the revision assistant suggests that providing feedback on certain types of revisions can inspire users to make more revisions to their essay.

9.0 BIBLIOGRAPHY

Adler, B. T., De Alfaro, L., Mola-Velasco, S. M., Rosso, P., and West, A. G. (2011). Wikipedia vandalism detection: Combining natural language, metadata, and reputation features. In *Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing - Volume Part II*, CICLing'11, pages 277–288, Berlin, Heidelberg. Springer-Verlag.

Alba, E., Luque, G., and Araujo, L. (2006). Natural language tagging with genetic algorithms. *Information Processing Letters*, 100(5):173–182.

Alsaif, A. and Markert, K. (2011). Modelling discourse relations for Arabic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 736–747. Association for Computational Linguistics.

Araujo, L. (2002). Part-of-speech tagging with evolutionary algorithms. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 230–239. Springer.

Attali, Y. and Burstein, J. (2006). Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3).

Baker, R. S., Corbett, A. T., Koedinger, K. R., and Wagner, A. Z. (2004). Off-task behavior in the cognitive tutor classroom: when students game the system. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 383–390.

Barzilay, R. and Elhadad, N. (2003). Sentence alignment for monolingual comparable cor-

pora. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 25–32. Association for Computational Linguistics.

Blunsom, P. and Cohn, T. (2006). Discriminative word alignment with conditional random fields. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 65–72. Association for Computational Linguistics.

Bott, S. and Saggion, H. (2011). An unsupervised alignment algorithm for text simplification corpus construction. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 20–26. Association for Computational Linguistics.

Bridwell, L. S. (1980). Revising strategies in twelfth grade students' transactional writing. *Research in the Teaching of English*, pages 197–222.

Bronner, A. and Monz, C. (2012). User edits classification using document revision histories. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 356–366, Stroudsburg, PA, USA. Association for Computational Linguistics.

Burstein, J. and Marcu, D. (2003). A machine learning approach for identification thesis and conclusion statements in student essays. *Computers and the Humanities*, 37(4):455–467.

Burstein, J., Marcu, D., and Knight, K. (2003). Finding the write stuff: Automatic identification of discourse structure in student essays. *Intelligent Systems, IEEE*, 18(1):32–39.

Cabrio, E., Tonelli, S., and Villata, S. (2013). From discourse analysis to argumentation schemes and back: Relations and differences. In *Computational Logic in Multi-Agent Systems*, pages 1–17. Springer.

Carlson, L., Okurowski, M. E., Marcu, D., Consortium, L. D., et al. (2002). *RST discourse treebank*. Linguistic Data Consortium, University of Pennsylvania.

Cho, K. and MacArthur, C. (2010). Student revision with peer and expert reviewing. *Learning and Instruction*, 20(4):328–338.

Cho, K. and Schunn, C. D. (2007). Scaffolded writing and rewriting in the discipline: A web-based reciprocal peer review system. *Computers & Education*, 48(3):409–426.

Danlos, L., Antolinos-Basso, D., Braud, C., and Roze, C. (2012). Vers le FDTB: French Discourse Tree bank. In *Proceedings of TALN 2012: 19ème conférence sur le Traitement Automatique des Langues Naturelles*, volume 2, pages 471–478. ATALA/AFCP.

Daxenberger, J. and Gurevych, I. (2012). A corpus-based study of edit categories in featured and non-featured Wikipedia articles. In *Proceedings of COLING 2012*, pages 711–726, Mumbai, India. The COLING 2012 Organizing Committee.

Daxenberger, J. and Gurevych, I. (2013). Automatically classifying edit categories in Wikipedia revisions. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 578–589, Seattle, Washington, USA. Association for Computational Linguistics.

Draft (2015). Draft. https://draftin.com/. [Online; accessed 10-03-2015].

Early, J. S. and Saidy, C. (2014). A study of a multiple component feedback approach to substantive revision for secondary ell and multilingual writers. *Reading and Writing*, 27(6):995–1014.

Faigley, L. and Witte, S. (1981). Analyzing revision. *College composition and communication*, pages 400–414.

Falakmasir, M. H., Ashley, K. D., Schunn, C. D., and Litman, D. J. (2014). Identifying thesis and conclusion statements in student essays to scaffold peer review. In *Intelligent Tutoring Systems*, pages 254–259. Springer.

Feng, V. W. and Hirst, G. (2011). Classifying arguments by scheme. In *Proceedings of the*

49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pages 987–996. Association for Computational Linguistics.

Feng, V. W. and Hirst, G. (2014). Two-pass discourse segmentation with pairing and global features. *CoRR*, abs/1407.8215.

Feng, V. W., Lin, Z., Hirst, G., and Holdings, S. P. (2014). The impact of deep hierarchical discourse structures in the evaluation of text coherence. In *Proceedings of International Conference on Computer Linguistics*, pages 940–949.

Ferschke, O., Zesch, T., and Gurevych, I. (2011). Wikipedia revision toolkit: efficiently accessing wikipedia's edit history. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations*, pages 97–102. Association for Computational Linguistics.

Forbes-Riley, K., Zhang, F., and Litman, D. (2016). Extracting pdtb discourse relations from student essays. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 117–127, Los Angeles. Association for Computational Linguistics.

Graesser, A., McNamara, D., Cooper, H., Camic, P., Gonzalez, R., Long, D., and Panter, A. (2012). Use of computers to analyze and score essays and open-ended verbal responses. *APA handbook of research methods in psychology. Washington, DC: American Psychological Association.*

Grammarly (2016). Grammarly. http://www.grammarly.come. [Online; accessed 01-08-2017].

Guo, Y., Korhonen, A., and Poibeau, T. (2011). A weakly-supervised approach to argumentative zoning of scientific documents. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 273–283. Association for Computational Linguistics.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Hashemi, H. B. and Schunn, C. D. (2014). A tool for summarizing students' shanges across drafts. In *International Conference on Intelligent Tutoring Systems(ITS)*.

Hearst, M. A. (1997). Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics*, 23(1):33–64.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Iida, R. and Tokunaga, T. (2014). Building a corpus of manually revised texts from discourse perspective. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 936–941, Reykjavik, Iceland. European Language Resources Association (ELRA).

Javanmardi, S., McDonald, D. W., and Lopes, C. V. (2011). Vandalism detection in wikipedia: a high-performing, feature-rich model and its reduction through lasso. In *Proceedings of the 7th International Symposium on Wikis and Open Collaboration*, pages 82–90. ACM.

Jing, H. (2002). Using hidden markov modeling to decompose human-written summaries. *Computational linguistics*, 28(4):527–543.

Jones, J. (2008). Patterns of revision in online writing a study of wikipedia's featured articles. *Written Communication*, 25(2):262–289.

Khan, Z., Balch, T., and Dellaert, F. (2004). An mcmc-based particle filter for tracking multiple interacting targets. In *European Conference on Computer Vision*, pages 279–290. Springer.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289.

Lawrence, J., Reed, C., Allen, C., McAlister, S., and Ravenscroft, A. (2014). Mining arguments from 19th century philosophical texts using topic based modelling. In *Proceedings of the First Workshop on Argumentation Mining*, pages 79–87, Baltimore, Maryland. Association for Computational Linguistics.

Lee, J., Yeung, C. Y., Zeldes, A., Reznicek, M., Lüdeling, A., and Webster, J. (2015). Cityu corpus of essay drafts of english language learners: a corpus of textual revision in second language writing. *Language Resources and Evaluation*, pages 1–25.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707.

Lin, Z., Ng, H. T., and Kan, M.-Y. (2014). A PDTB-styled end-to-end discourse parser. *Natural Language Engineering*, 20(02):151–184.

Liu, M., Calvo, R. A., and Pardo, A. (2013). Tracer: A tool to measure and visualize student engagement in writing activities. In *Advanced Learning Technologies (ICALT), 2013 IEEE 13th International Conference on*, pages 421–425. IEEE.

Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual*

Meeting of the Association for Computational Linguistics: System Demonstrations, pages 55–60.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mizumoto, T., Komachi, M., Nagata, M., and Matsumoto, Y. (2011). Mining revision log of language learning sns for automated japanese error correction of second language learners. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 147–155. Asian Federation of Natural Language Processing.

Moens, M.-F., Boiy, E., Palau, R. M., and Reed, C. (2007). Automatic detection of arguments in legal texts. In *Proceedings of the 11th international conference on Artificial intelligence and law*, pages 225–230. ACM.

Moore, R. C., Yih, W.-t., and Bode, A. (2006). Improved discriminative bilingual word alignment. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 513–520. Association for Computational Linguistics.

NCES (2011). National writing assessment of educational progress at grades 8 and 12.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453.

Nelken, R. and Shieber, S. M. (2006). Towards robust context-sensitive sentence alignment for monolingual corpora. In *EACL*.

Okazaki, N. (2007). Crfsuite: a fast implementation of conditional random fields (crfs).

Palau, R. M. and Moens, M.-F. (2009). Argumentation mining: the detection, classification

and structure of arguments in text. In *Proceedings of the 12th international conference on artificial intelligence and law*, pages 98–107. ACM.

Park, J., Katiyar, A., and Yang, B. (2015). Conditional random fields for identifying appropriate types of support for propositions in online user comments. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 39–44, Denver, CO. Association for Computational Linguistics.

Pfeil, U., Zaphiris, P., and Ang, C. S. (2006). Cultural differences in collaborative authoring of wikipedia. *Journal of Computer-Mediated Communication*, 12(1):88–113.

Pitler, E., Louis, A., and Nenkova, A. (2009). Automatic sense prediction for implicit discourse relations in text. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 683–691. Association for Computational Linguistics.

Pitler, E. and Nenkova, A. (2009). Using syntax to disambiguate explicit discourse connectives in text. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 13–16. Association for Computational Linguistics.

Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A. K., and Webber, B. L. (2008). The penn discourse treebank 2.0. In *LREC*. Citeseer.

Prasad, R., Webber, B., and Joshi, A. (2014). Reflections on the Penn Discourse Treebank, comparable corpora, and complementary annotation. *Computational Linguistics*.

Raghava, G., Searle, S. M., Audley, P. C., Barber, J. D., and Barton, G. J. (2003). Oxbench: a benchmark for evaluation of protein multiple sequence alignment accuracy. *BMC bioinformatics*, 4(1):1.

Silva, A. P., Silva, A., and Rodrigues, I. (2013). A new approach to the pos tagging problem using evolutionary computation. In *RANLP*, pages 619–625.

Southavilay, V., Yacef, K., Reimann, P., and Calvo, R. A. (2013). Analysis of collaborative writing processes using revision maps and probabilistic topic models. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, pages 38–47. ACM.

Stab, C. and Gurevych, I. (2014). Identifying argumentative discourse structures in persuasive essays. In *Conference on Empirical Methods in Natural Language Processing 2014, Association for Computational Linguistics, pages 4656., 2014.*, pages 46–56.

Stab, C., Kirschner, C., Eckle-Kohler, J., and Gurevych, I. (2014). Argumentation mining in persuasive essays and scientific articles from the discourse structure perspective. *Frontiers and Connections between Argumentation Theory and Natural Language Processing, Bertinoro, Italy.*

Tan, C. and Lee, L. (2014). A corpus of sentence-level revisions in academic writing: A step towards understanding statement strength in communication. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 403–408, Baltimore, Maryland. Association for Computational Linguistics.

Teufel, S. (2000). *Argumentative zoning: Information extraction from scientific text.* PhD thesis, Citeseer.

Toulmin, S. E. (2003). *The uses of argument.* Cambridge University Press.

Turnitin (2016). Turnitin. http://turnitin.com/en_us/what-we-offer/revision-assistant. [Online; accessed 01-22-2016].

Vaughan, M. M. and McDonald, D. D. (1986). A model of revision in natural language generation. In *Proceedings of the 24th annual meeting on Association for Computational Linguistics*, pages 90–96. Association for Computational Linguistics.

Verbert, K., Duval, E., Klerkx, J., Govaerts, S., and Santos, J. L. (2013). Learning analytics dashboard applications. *American Behavioral Scientist*, pages 1500–1509.

Wang, W., Su, J., and Tan, C. L. (2010). Kernel based discourse relation recognition with

temporal ordering information. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 710–719. Association for Computational Linguistics.

Webber, B. (2004). D-ltag: extending lexicalized tag to discourse. *Cognitive Science*, 28(5):751–779.

Wilson, T. (2008). Fine-grained subjectivity analysis. *Unpublished doctoral dissertation, University of Pittsburgh*.

Writelab (2015). WriteLab. http://home.writelab.com. [Online; accessed 10-03-2015].

Xu, W., Callison-Burch, C., and Dolan, W. B. (2015). Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*.

Xue, H. and Hwa, R. (2014). Improved correction detection in revised esl sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 599–604, Baltimore, Maryland. Association for Computational Linguistics.

Zeyrek, D., Demirşahin, I., Sevdik-Çallı, A., and Çakıcı, R. (2013). Turkish Discourse Bank: Porting a discourse annotation style to a morphologically rich language. *Dialogue and Discourse*, 4(2):174–184.

Zhang, F., B. Hashemi, H., Hwa, R., and Litman, D. (2017). A corpus of annotated revisions for studying argumentative writing. In *Proceedings of the 2017 Conference of the Association for Computational Linguistics*. Association for Computational Linguistics.

Zhang, F., Hwa, R., Litman, D., and B. Hashemi, H. (2016a). Argrewrite: A web-based revision assistant for argumentative writings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 37–41, San Diego, California. Association for Computational Linguistics.

Zhang, F. and Litman, D. (2014). Sentence-level rewriting detection. In *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 149–154, Baltimore, Maryland. Association for Computational Linguistics.

Zhang, F. and Litman, D. (2015). Annotation and classification of argumentative writing revisions. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 133–143, Denver, Colorado. Association for Computational Linguistics.

Zhang, F. and Litman, D. (2016). Using context to predict the purpose of argumentative writing revisions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1424–1430, San Diego, California. Association for Computational Linguistics.

Zhang, F. and Litman, D. (2017). A joint identification approach for argumentative writing revisions. arXiv:1703.00089.

Zhang, F., Litman, D., and Forbes-Riley, K. (2016b). Inferring discourse relations from pdtb-style discourse labels for argumentative revision classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2615–2624, Osaka, Japan. The COLING 2016 Organizing Committee.

Zhou, Y. and Xue, N. (2015). The Chinese Discourse Treebank: A Chinese corpus annotated with discourse relations. *Language Resources and Evaluation*, 49(2):397–431.

Zhou, Z.-M., Xu, Y., Niu, Z.-Y., Lan, M., Su, J., and Tan, C. L. (2010). Predicting discourse connectives for implicit discourse relation recognition. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1507–1514. Association for Computational Linguistics.

APPENDIX A

ANNOTATION MANUAL

A.1 REVISION ANNOTATION CODING TABLE

Attribute	**Codes**	**Note**
Aligned Index (in Sheet 0)	The index of the aligned sentence in the new draft (Numerical value starting from 1) or "DELETE"	1) The aligned index should be the same number as the Sentence Index of the aligned sentence in sheet1 2) When a sentence could not be aligned to any sentences in the new draft, mark it as "DELETE"
Aligned Index (in Sheet 1)	The index of the aligned sentence in the original draft (Numerical value starting from 1) or "ADD"	1) The aligned index should be the same number as the "Sentence Index" of the aligned sentence 2) When a sentence could not be aligned to any sentences in the original draft, mark it as "ADD"
Original Paragraph Index (in Sheet 0)	The index of the paragraph the sentence is in in the original draft (Numerical value starting from 1) or blank	Should check the raw file for the annotation of this column

Original Paragraph Index (in Sheet 1)	The index of the paragraph the sentence is in in the original draft (Numerical value starting from 1) or blank	1) Should check the raw file for the annotation of this column 2) When a sentence is newly added, this cell is left blank
New Paragraph Index (in Sheet 0)	The index of the paragraph the sentence is in in the second draft (Numerical value starting from 1)	1) Should check the raw file for the annotation of this column 2) When the sentence is deleted in the second draft, this cell is left blank
New Paragraph Index (in Sheet 1)	The index of the paragraph the sentence is in in the second draft (Numerical value starting from 1)	Should check the raw file for the annotation of this column

Revision Purpose	Select from the following list • Claim/Ideas • Warrant/Reasoning/Backing • Rebuttal/Reservation • General Content • Evidence • Conventions/Grammar/Spelling • Word Usage/Clarity • Word Usage/Clarity - CASCADED	1) Claim/Ideas: the position or claim being argued for; the conclusion of the argument. Also described as "thesis" of a paragraph or essay. 2) Warrant/Reasoning/Backing: principle or reasoning of the claim and justification to the claims/ideas 3) Rebuttal/Reservation: exception to the claim/ideas 4) General Content: When the content-development is not explicitly related to the claim, mark it as general content. (i.e. the content is not claim, reasoning for the claim, rebuttal for the claim or evidence for the claim) 5) Evidence: evidence or example for the claim. Evidence has to be either examples/citations/theorems

		6) Conventions/Grammar/Spelling: changes to fix spelling or grammar errors, misusage of punctuations or to follow the organizational conventions of academic writing (If the previous organization could be considered as a mistake) 7) Word Usage/Clarity **Clarity**: changes of words or phrases for better representation of the authors ideas or to follow a specific requirement **Word Usage**: replacement of a specific word/phrase, the replacement can either be for the purpose of better word choosing or the word 8) Word Usage/Clarity-CASCADED: A specific category of Word Usage/Clarity for describing cascaded changes

Table A1: Coding table

A.2 ALIGNMENT ANNOTATION

A.2.1 Description

After processing the original documents, for each draft, the N sentences in the draft are assigned indexes from 1 to N according to their occurrence in the paper. For sentence alignment, each sentence in the revised draft is assigned the index of its aligned sentence in the original draft. Also, each sentence in the original draft is assigned the index of the aligned sentence in the revised draft. If a sentence is newly added, it will be annotated as

ADD. If a sentence is deleted from the old draft, it will be marked as DELETE.

A.2.2 Rules

- Every sentence should be either aligned or marked as ADD or DELETE. **Only** the alignment from the old draft to the new draft contains DELETE and **only** alignment from the new draft to the old draft contains ADD. We only allow the sentence alignments to be one-to-one, one-to-many and many-to-one cases.
- For one-to-one case, align the sentences if the sentence is either a replication of the other sentence or a modification of the other sentence with one or several of the following changes: a. addition/deletion of content b. modification of words, phrases c. restatement of the ideas. The sentences that are aligned should be either semantically close or syntactically close and within the same/similar context. (i.e. the paragraphs the sentences belong to should be similar). [1]
- For many-to-one and one-to-many cases, only align multiple sentences to one sentence (one-to-many and many-to-one) when it is explicit that the multiple sentences should be grouped together to be aligned to the one sentence. (i.e. For the group of multiple sentences, it is explicit that it is better to align the target sentence to the merged sentences than to align the target sentence to one or some of the sentences.)

A.3 REVISION PURPOSE ANNOTATION

A.3.1 Rules

The order of revision purpose type importance

Importance Orders of Revision Purposes (Higher to lower):

[1]Semantically similar: The two sentences describes the same information, or the other sentence adds/deletes information on the basis of the other sentence; Syntactically similar: The two sentences look explicitly similar to each other. (i.e. the difference between the two sentences should be a small ratio of the whole sentence. For example, a normal sentence with less than 10 words should have at most 2 words (Does not count the change of words in the same stem, e.g. change-¿ changes is not counted in the number of differences) that are different).

$Claims/Ideas > Rebuttal/Reservation >= Warrant/Reasoning/Backing$
$>= Evidence > GeneralContent > Conventions/Grammar/Spelling$
$> WordUsage/Clarity$

As said above, except for the special case of Word Usage/Clarity, only one major revision purpose type should be selected for each revision unit. The importance of different revision purpose types are different, when there are multiple revision purpose types in one revision, make sure that the more important one is selected. The following sub-rules explains more specific details for cases where the decision of the appropriate revision purpose can be difficult.

a) Claim/Ideas vs. Warrant/Reasoning/Backing

One typical case is that a paper can have a major claim and several subclaims to support the major claims. These subclaims are usually in the form of reasoning to support the major claim. Thus the differentiation of claim and reasoning can be ambiguous. We ask the annotators to think of the Claim and Reasoning as a hierarchical tree structure. The leaves of the tree are marked as "Warrant/Reasoning/Backing" while the others are marked as "Claim/Ideas". In specific, if a sentence is further supported or objected by other sentences, it is considered as a claim. Meanwhile, if there are no other sentences (Reasoning or Evidence or Rebuttal) for or against this sentence, it is marked as "Warrant/Reasoning/Backing".

b) General Content vs. Warrant/Reasoning/Backing

Differentiating General Content and Reasoning can be difficult as they both often occur after the author proposes a claim. To differentiate the two categories, the annotator is required to distinguish whether the author is suggesting his position for his claim in the sentences or not. If the annotator senses the author's sentiment position towards his claim, then it should be "Warrant/Reasoning/Backing", whereas it should be "General Content".

c) Evidence vs. Warrant/Reasoning/Backing

These two categories are similar as they both provide support to the authors' claim. The annotators are required to distinguish these two categories according to whether the sentences are stating facts. The facts can be (1) Citation: the citation of papers, reports, news and books. (2) Example: facts of history or personal experiences. (3) Scientific proof. If there are facts involved, it is marked as Evidence, otherwise it is marked as Warrant.

d) Conventions/Grammar/Spelling vs. Word Usage/Clarity:

These two genres are similar as they don't change the content of the text and improve the quality of the text. The annotators are required to make the judgment according to the question: Are there spelling/grammar mistakes in the original draft and has this mistake been addressed in the new draft? If the mistake is addressed, it should be marked as "Grammar/Spelling".

Annotate according to what has the author changed rather than where the author changed

It is not necessarily that revisions made on the thesis of the paragraph are Claim/Idea changes, the type of the change should be determined according to what the author really has changed. For example, in a Claim sentence of a paragraph, if the author added a clause in the new sentence for reasoning the claim, the change would be a Warrant/Reasoning/Backing change; if the author only replaced some word with a more appropriate form of word, the annotator should mark it as Word usage change.

Handling the cascaded changes

The handling of the cascaded changes should follow Rule 2, for example, in the case below:

- Change 1: Sadam Hussein would be the perfect example here. $->$ Fidel Castro and Kim Joon En are perfect examples here.
- Change 2: He killed people who are against him in Iraq. $->$ They killed people who are against them in their countries.

In this example, the author changed the person from "Sadam Hussein" to "Fidel Castro and Kim Joon En" in Change 1. Depending on the topic of the paper, it can be either a Claim/Ideas change or an Evidence change. However, Change 2 is a cascaded change due to Change 1. While here there is the change from "he" to "they" indicating the change of the subjects, it should be marked as "Word Usage/Clarity".

In other words, the type of cascaded changes should be decided according to what really has been changed. In the example above, the changes in the subject resulted in the changes from "He" to "They". But as a reasoning sentence, the way the author reasons

about his claim/evidence is still the same. Thus the change should be categorized as "Word Usage/Clarity-CASCADED". In other cases, for example, if the author changed the claim of paper, and resulted the changes in the way he reasons about his claim, the changes would be categorized as "Warrant/Reasoning/Backing".

The special case of Word Usage/Clarity change

In the various types of Word Usage/Clarity change, simply replacing a word/phrase of the sentence is quite commonly seen. However, annotating them can be difficult. A simple word/phrase replacement can be just surface fixing or important content change.

For example, there is usually not a big difference between Alice is a good person to Alice is a great person. However, switching from This is 10 feets long to This is 10 inches long will be very important concept change. Even more, sometimes the author might even change the claim of a paragraph by simply replacing a word. For example, in the prompt where the student is required to provide an answer to the question. Changing from Species A would survive in this environment to Species B would survive in this environment would be a Claim change.

Thus for the specific case of Word Usage/Clarity change, the annotator is required to annotator two revision purposes if necessary. In specific, for the case of word/phrase replacement, if the replacement DOES NOT involve an important concept change, just mark it as Word Usage/Clarity. If the replacement involves importance changes such as feets to inches in the example above, the annotator should also annotate the specific content change type IN ADDITION to Word Usage/Clarity. For example, changing from This is 10 feets long to This is 10 inches long would typically be Word Usage/Clarity + General Content change.

Read and understand the prompt before the annotation

Sometimes the annotation of revision purpose could be different according to what the author is really targeting. So it is critically important that the annotator read and understand the prompt before the annotation. For example, in a regular essay, a sentence change from Fidel Castro would be a good example for this case to Sadam Hussein would be a good example for this case would typically be Word Usage/Clarity + Evidence. However, if the prompt of the essay writing assignment is Put the contemporaries at different levels of Hell, then the annotation would be Word Usage/Clarity + Claim/Ideas.

APPENDIX B

ARGREWRITE STUDY MATERIALS

B.1 PRESTUDY SURVEY QUESTIONS

The prestudy survey questions involves questions about the demographics of the participant and the participant's previous writing behaviors.

For the demographic questions, the participants have to select from the given options.

- What is your major? (Natural Sciences vs. Social Sciences vs. Humanities)
- Are you an undergraduate or graduate student? (Undergradudate vs. gradudate vs. None)
- What is your current year of study? (1st year vs. 2nd year vs. 3rd year vs. 4th year vs. 5th year or above)
- Is English your native language? (Yes or No)

The writing behavior questions involve the following:

- What are some of your recent classes that have an intensive writing component to them? How did you do in these classes?
- When writing a paper for a class, how many drafts of major revisions do you typically make? (0 vs. 1 vs. 2 vs. 3 vs. 4 or above)
- Overall, how confident are you with your writings (1 to 5 from Not at all confident to Extremely confident)?

- What aspects of writing do you think you are good at? e.g. vocabulary choice, clear sentences, writing organization.
- What aspects of writing do you think you can improve?

B.2 POSTSTUDY SURVEY QUESTIONS

For users in the experiment and the control group, they all have to rate the following questions from 1 to 5 (Strongly Disagree to Strongly Agree):

- The system allows me to have a better understanding of my previous revision efforts.
- It is convenient to view my previous revisions with the system.
- The system helps me to recognize the weakness of my essay.
- The system encourages me to make more revisions than I usually make.
- The system encourages me to think more about making more meaningful changes.
- Overall the system is helpful to my writing.

They are also instructed to answer a subjective question: "How would you expect the system to be more helpful / what other designs of system is helpful to you?".

For the experiment group users using the ArgRewrite system, they have to rate 4 additional questions from 1 to 5:

- Taxonomy of revisions inspires me to make more changes
- Listing the changes for me inspires me to make more changes
- Visualization of revision distribution inspires me to make more changes
- Difference between system recognition and self recognition inspires me to make more changes

The control group users have 1 additional question to rate from 1 to 5: "The presentation of "Diff" inspires me to make more changes"

B.3 TUTORIALS BEFORE DRAFT3 WRITING

ArgRewrite interface tutorial

Brief introduction to the key functions of our system

1. The revisions to your essay that were identified by our system are color coded to distinguish between two major categories of revision type: Content and Surface. Content revisions change the **information/content** of the essay (for example, by modifying the thesis or adding evidence). Surface revisions change the **surface** details of the essay (for example, by correcting grammar mistakes or making the language more fluent). Hover your mouse on the specific items on the left and you will be given a detailed definition of each category. The **content** changes are colored in **"warm"** colors while the **surface** changes are colored in **"cold"** colors.
2. On the right you were given a **"revision map"** to view the changes you made. In the revision map, each sentence is represented as a tile and each paragraph is composed of a segment of tiles. The tile is pale if there are no changes from draft 1 to draft 2 on the sentence. If the tile is colored, the color represents the revision type that best categorizes the purpose of the change for that sentence. Click on the tile to view the details of the revision.

Hints to rewriting with our system

1. **More** "**warm**" color goal.
   Research on revisions indicates that making content changes will most improve your writings. So you should aim to see **more** **"warm"** colors than **"cold"** colors in the interface.
2. Expect **multiple** warm colors in new added paragraphs.
   In good writing, each paragraph is well structured. In the writing of persuasive essays, we expect each paragraph to have one or two sentences as the thesis or rebuttal, at least one sentence to reason for the thesis and one sentence to provide evidence for the thesis. So if a new paragraph only contains one warm color or no warm colors, try to add more content to make the paragraph better.
3. Reflect on all revisions where the mapped color does **not** agree with your expectation.
   Your revision is labeled by a human expert based on what he or she thinks your purpose was for making the revision. It is possible that the human expert made a mistake when labeling your revision, but it is also likely that your change did not always successfully achieve your desired rewriting goal. So whenever the color of the revision type does not agree with your intended revision type, think about how to change the sentence and try to revise it to better implement your revision intention.
4. Check whether your revisions agree with the **prompt**
   You were given a prompt when you revised your essay from draft 1 to draft 2, and in the prompt you were required to add new content such as rebuttals. Check the revision map to confirm that your second draft contains what you were required to write.

Revision examples (Ignore Unknown at this moment)

The content changes can be sentences added/deleted/modified

1. Additions/Deletions

When there is an added/Deleted paragraph, set the revision type to the argument type of the sentence

Example 1

Topic: Arguing what rhetoric strategy Richard Louv uses in his argument

Draft 1:	Draft 2:
Empty	**First and foremost, Richard Louv uses an anecdote to show that children are actually being encouraged to use electronics rather than enjoy nature.** Louv mentions someone who is buying a new car, when the buyer says no to adding "backseat entertainment," the salesman is shocked. This anecdote explains that electronic entertainment for children is becoming more and more common, almost to the point where it is a standard. Louv believes that this "standard" is taking away the knowledge that children gain by looking out the window of a car. **Ideas Reasoning Other**

Example 2

Topic: Arguing how does electronic communication impacts social relationships

Draft 1:	Draft 2:
Empty	**A final example of why electronic communication takes away from interpersonal relationships is the case of the Republic of Zimbabwe.** In this southern African nation, telecommunications are scarce and communication primarily relies on neighbors talking to one another, or families writing letters to one another. This type of verbal and non-verbal communication have helped to develop strong interpersonal relationships with family members and friends. This example proves how this type on non-electronic communication can alternatively be more powerful at establishing inter-personal relationships. **In contrast, I acknowledge there is one advantage to using electronic communications to establish interpersonal relationships**. If a natural disaster such as a hurricane or tornado were to occur, I acknowledge that using electronic forms of communication such as texting or social media are critical to reach out to your relatives and friends, and let them know that you and your family are all safe. **Evidence Reasoning Rebuttal**

2. Modifications

Content modifications modifies the corresponding argument role, and changes the information contained in the content. The biggest difference with surface modifications is that the content modifications **changes the information** of the essay

Example 1

Topic: Arguing the rhetoric strategies used by Florence Kelly in her speech.

Draft 1:	Draft 2:
Child labor was not so much of an issue to onlookers as it was to the actual children who partook in breadwinning at ages as young as six years old. In her attempt to enlighten the public about the severe injustices surrounding child labor laws, Florence Kelley delivered a powerful speech before the convention of the National American Woman Suffrage Association on July 22, 1905. Kelley's use of conspicuous repetition, cheeky sarcasm, and a thought inducing oxymoron in her speech helped emphasize the crime in the practice of child labor.	Child labor was not so much of an issue to onlookers as it was to the actual children who partook in breadwinning at ages as young as six years old. In her attempt to enlighten the public about the severe injustices surrounding child labor laws, Florence Kelley delivered a powerful speech before the convention of the National American Woman Suffrage Association on July 22, 1905. Kelley's use of conspicuous repetition, cheeky sarcasm, **pathos,** and an oxymoron in her speech helped emphasize the crime in the practice of child labor and the need for reformation. **Ideas**

NOTE: While there is only one "pathos" added here, this is a change to the major claim of the essay.

Example 2

Topic: Arguing how does electronic communication impacts social relationships

Draft 1:	Draft 2:
An example for the case where the electronic communication is limited would be China.	An example for the case where the electronic communication is limited would be **North Korea.** **Evidence**

Example 3

Draft 1:	Draft 2:
He is tall. An example for the case where the electronic communication is limited would be China.	He is **6 feet** tall. An example for the case where the electronic communication is limited would be **mainland** China. **Precision**

Surface changes can only be sentences reordered or modified

Example 1

Topic: Argue the type of person that should be put into certain levels of Hell

Draft 1:	Draft 2:
The hoarders and the spendthrifts linger in the fourth level of Hell. The first example would be Donald Trump. The second example would be Candy Spelling.	The spendthrifts and hoarders linger in the fourth level of Hell. The first example would be Candy Spelling. The second example would be Donald Trump. **Reordering**

Example 2

Draft 1:	Draft 2:
He is tall. He runs fast. These technologies have been beneficial for many reasons. Although, people can use it in positive ways such as the case mentioned above, it can be harmful for relationships.	He is a tall **person**. He is a fast runner. **Many, if not all, of the people agree on this, that** these technologies are beneficial from many different aspects. Although, people can use **these technologies** in positive ways such as the cases mentioned above, **they** can be harmful for relationships. **Fluency**

Note: Surface revisions either make no changes or make very limited changes on the information of sentences.

Example 3

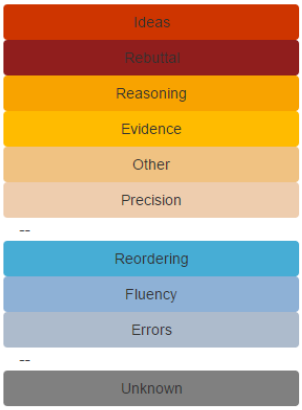
Draft 1:	Draft 2:
He like oranges. He is a good person, He loves to help people.	He likes oranges. He is a good person. He loves to help people. Errors

Revision Reasons

Besides self-motivation, we listed four possible revision motivations from our system.

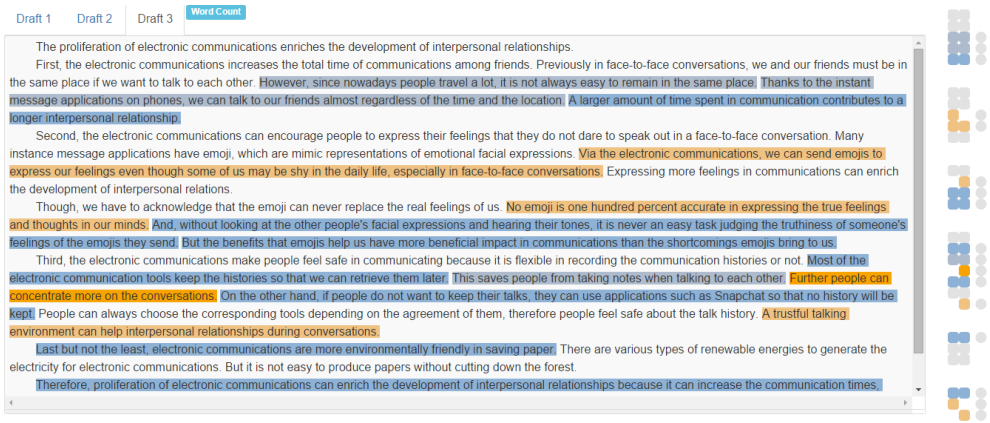
Taxonomy of revisions

The types of revisions defined in the system inspires you on what you can do to improve the essay. Basically, this following list of revision categories is being helpful.



Listing the changes for me

The fact that the system marks every changes you make is helpful. Basically, the revision map and the highlights on the text makes you aware of where you have changed and reminds you what to change next.



Visualization of revision distribution (Warm/Cold)

Cold colors, only surface revisions

Warm colors, content revisions

You make the revisions not only because the system tells you where you have changed, but also because the system gives you the information of Content/Surface changes. The visualization of warm color content changes and cold color surface changes help you to recognize which part of the essay is only containing surface changes and which part of the essay is containing idea/content changes.

Difference between system recognition and self-recognition

Difference between system recognition and self-recognition

The exact type of the revision recognized by the system is not the same as the revision type I thought I made. This difference reminds me that maybe my revision is not clear enough to be recognized and inspires me to further revise.

Control interface tutorial

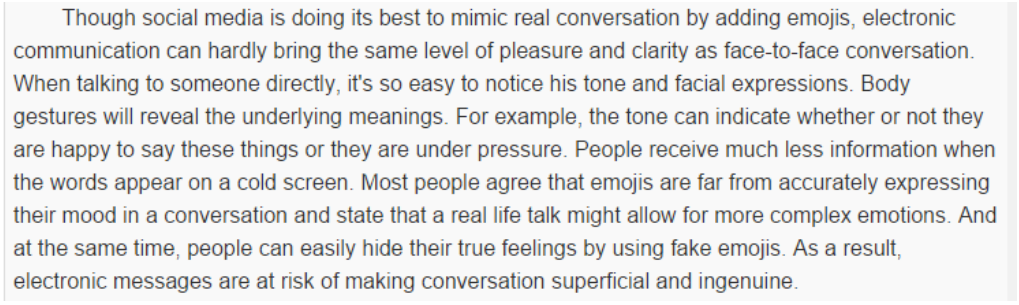
Brief Introduction to the system

The system incorporates a word-level diff algorithm to help you locate your revisions. In Draft1, the deleted words/sentences are shown in red with strike. In Draft 2, the added words/sentences are shown in green. Notice that there is no "Modification" here, which means that the modification from one word/sentence to another word/sentence will be viewed as a deleted word/sentence in Draft 1 and an added word/sentence in Draft 2.

Examples

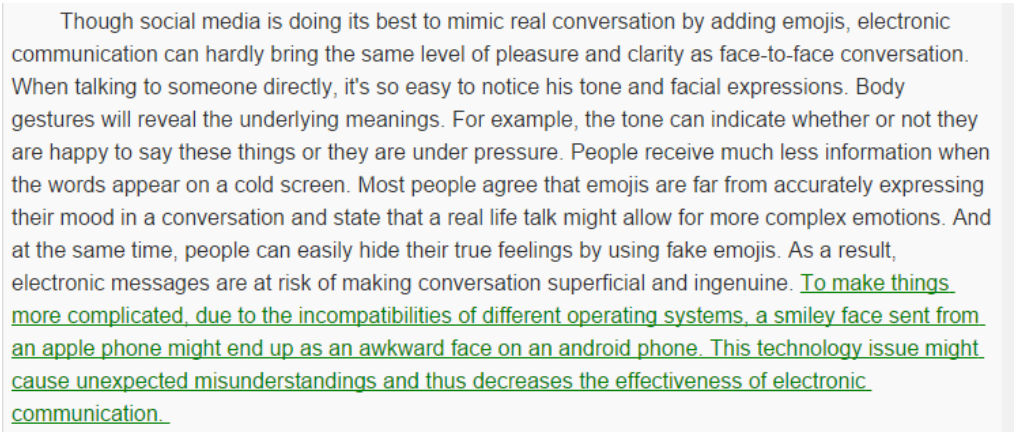
Add shown as green underline in Draft 2

Draft 1



Though social media is doing its best to mimic real conversation by adding emojis, electronic communication can hardly bring the same level of pleasure and clarity as face-to-face conversation. When talking to someone directly, it's so easy to notice his tone and facial expressions. Body gestures will reveal the underlying meanings. For example, the tone can indicate whether or not they are happy to say these things or they are under pressure. People receive much less information when the words appear on a cold screen. Most people agree that emojis are far from accurately expressing their mood in a conversation and state that a real life talk might allow for more complex emotions. And at the same time, people can easily hide their true feelings by using fake emojis. As a result, electronic messages are at risk of making conversation superficial and ingenuine.

Draft 2



Though social media is doing its best to mimic real conversation by adding emojis, electronic communication can hardly bring the same level of pleasure and clarity as face-to-face conversation. When talking to someone directly, it's so easy to notice his tone and facial expressions. Body gestures will reveal the underlying meanings. For example, the tone can indicate whether or not they are happy to say these things or they are under pressure. People receive much less information when the words appear on a cold screen. Most people agree that emojis are far from accurately expressing their mood in a conversation and state that a real life talk might allow for more complex emotions. And at the same time, people can easily hide their true feelings by using fake emojis. As a result, electronic messages are at risk of making conversation superficial and ingenuine. To make things more complicated, due to the incompatibilities of different operating systems, a smiley face sent from an apple phone might end up as an awkward face on an android phone. This technology issue might cause unexpected misunderstandings and thus decreases the effectiveness of electronic communication.

Delete shown as red strike section in Draft 1

Draft 1

For too long humanity had been beholden to the limitations of physical proximity for the allowance of clear and open communication. How many tragedies could have been prevented with faster communication? How many projects more quickly completed without waste? How many wars concluded without needless bloody sacrifice? More than a century ago, this began to change. The telegraph and telephone, then radio, television, and now the ubiquitous Internet connectivity we all enjoy has fundamentally changed the way humans engage with each other. Humanity has never been better off.

~~Some argue that the so-called "Millennial" generation is the first to fully encapsulate themselves in new technology in a way that actually prevents the fomenting of real interpersonal relationships. Images of young adults on the bus or subway, each with nose firmly planed in the glowing rectangular screen of their mobile devices are meant to have us think "what a tragedy!". Dig just a little deeper and you will find similar images seventy years older of young adults firmly ignoring each other with newspapers or books in the same settings. The young adults of the present merely have the luxury of selecting whatever it is they wish to read or do in their personal time, with zero hassle.~~

It may be stated that a relationship that exists purely over an Internet connection is not real. How can two people who met briefly perhaps once continue to call themselves "friends" on Facebook years afterward? But how is this different than two people staying in touch by written letters as so many have done since the emergence of a reliable postal service? It may well be true that those who cite this as an example of a non-relationship have never had need of such a thing, and well cannot conceive of it.

Draft 2

For too long humanity had been beholden to the limitations of physical proximity for the allowance of clear and open communication. How many tragedies could have been prevented with faster communication? How many projects more quickly completed without waste? How many wars concluded without needless bloody sacrifice? More than a century ago, this began to change. The telegraph and telephone, then radio, television, and now the ubiquitous Internet connectivity we all enjoy has fundamentally changed the way humans engage with each other. Humanity has never been better off.

It may be stated that a relationship that exists purely over an Internet connection is not real. How can two people who met briefly perhaps once continue to call themselves "friends" on Facebook years afterward? But how is this different than two people staying in touch by written letters as so many have done since the emergence of a reliable postal service? It may well be true that those who cite this as an example of a non-relationship have never had need of such a thing, and well cannot conceive of it.

Modify shown as red strike in Draft 1 and green underline in Draft 2

Draft 1:

While others may say that electronic communication is decreasing the amount of ~~actual human~~ interaction, ~~I believe that it is just~~ provid~~ing~~ another method ~~of human interaction. Some say that we are not going outside to actually talk to people anymore, but electronic~~ communication allows for us to plan meeting up with others more efficiently. Otherwise, ~~you~~ would~~n't~~ know exactly when ~~your neighbor was free to meet for dinner without waiting for them to come home and then ringing their doorbell and asking.~~

Draft 2:

While others may say that electronic communication is decreasing the amount of in-person interaction, electronic communication truly helps to provide another method for people to interact. With more methods of interaction, the quantity of interaction increases, which helps foster long-term relationships. The additional amount of communication methods allows for us to plan meeting up with others more efficiently. Otherwise, it would be impossible to know exactly when friends or family were available.

Hints to rewrite

Our previous findings indicate that the revisions are only helpful when they contain more meaningful changes. This means that the change of one single word/phrase might not be enough. We encourage more added words/phrases in the revision to improve the paper.

Check whether your revisions agree with the prompt. You were given a prompt when you revised your essay from draft 1 to draft 2, and in the prompt you were required to add new content such as rebuttals. Check whether the revisions you have made contains what you were required to write.