Learning Sentential Patterns of Various Rhetoric Moves for Assisted Academic Writing

Jim Chang[1], Hsiang-Ling Hsu[2], Joanne Boisson[2]
Hao-Chun Peng[2], Yu-Hsuan Wu[2], Jason S. Chang[2]
[1]Department of Computer Science
[2]Department of Information System and Application
National Tsing-Hua University
Hsinchu, Taiwan R.O.C. 30013
{jim, hsiang, joanne, henry.p, shanny, jason}@nlplab.cc

Abstract

We introduce a new method for extracting representative sentential patterns from a corpus for the purpose of assisting ESL learners in academic writing. In our approach, sentences are transformed into patterns for statistical analysis and filtering, and then are annotated with relevant rhetoric moves. The method involves annotating every sentence in a given corpus with part of speech and base phrase information, converting the sentence into formulaic patterns, and filtering salient patterns for key content words (verbs and nouns). We display the patterns in the interactive writing environment, *WriteAhead*, to prompt the user as they type away.

1 Introduction

The British Council estimated that roughly a billion people are learning and using English around the world (Graddol, 1997), mostly as a second language, and the number has been growing. For advanced learners in university, English for Academic Purposes (EAP) plays an important role in English Specific Purposes (ESP) study. More and more Computer Assisted Language Learning (CALL) systems have been developed to help learners in academic writing, including concordancers, grammar checkers, and essay raters. Typical CALL systems assist learners *before* and *after* the writing process by providing corpus-based reference services, or returning a grade and corrective feedback (e.g., Cambridge English *Write & Improve*).

However, researchers have shown that non-native student writers may have difficulties in composing sentences and lack knowledge at discourse level (Hinds, 1990; Swales, 1990 or Connor,1996) in academic writing. For example, (Antony, 2003) indicated that many Japanese scientists and engineers lack sufficient knowledge of commonly used structural patterns at the discourse level.

The rhetorical organization has been considered to be one of the most effective strategies of teaching technical writing and reading. The American National Standard Institute (ANSI) recommends editors or writers to state the *purpose, methods, results,* and *conclusions* in the document (Weil, 1970). That is, an article usually begins with a description of background information, and is followed by the target problem, solution to the problem, evaluation of the solution, and conclusion, by analyzing annotated dictionary examples and automatically tagged sentences in a corpus. As will be described in Section 4, we used the information on collocation and syntax (ICS) for example sentences from online *Macmillan English Dictionary*, as well as in the *Citeseer x* corpus, to develop *WriteAhead*. Along the same line, the second edition of the *Macmillan English Dictionary* provides a 29-page *Improve your Writing Skills* Writing Section with instruction on how to write fluently by : *adding information, comparing and contrasting, exemplifying, expressing cause and effect, expressing personal opinions, possibility and certainty, introducing a concession or introducing topics, listing items, paraphrasing or clarifying, quoting and reporting, summarizing and concluding.*

Although there are much information (such as dictionary examples) that could help to write a paper, learners may fail to generalize from examples

Figure 1: Example WriteAhead session where an user typed "pp".

and apply to their own situations. Often, there are too many examples to choose from and to adapt to match the need of the learner writers. Learners could be more effectively assisted with a tool that provides concise, relevant, genre-specific suggestions as learners type away, when writing a draft. In our research, we automatically extract rhetorical patterns to assist learners in academic writing. For example, in Figure 1, the learner has already typed a sentence describing the background and problem, and then the learner types the move tag *AIM*.

Figure 2 shows the implementation of *WriteAhead* in the Google Doc environment. With this Google Docs Add-on, the user can conveniently access the *WriteAhead* functionalities, as well as common editing functions. According to the information, *WriteAhead* displays the appropriate sentential patterns and examples for the "method" extracted from a corpus, to help the learner continue writing:

- **Our ALGORITHM be BASE on** (Our approach/method is based on),
- **We ILLUSTRATE the ALGORITHM** (We illustrate the approach/method/technique),
- **The ALGORITHM be BASE on** (The method/approach/model is base on).

In this paper, we present a prototype system, *WriteAhead*, that extracts patterns that cover extensively most semantic categories in academic writing (e.g. Teufel, 2000) from an academic corpus.

Writing suggestions are given to assist student writers. *WriteAhead* extracts these sentential patterns and examples automatically by tagging and analyzing sentences in a corpus. As will be described in Section 3, we used the *Citeseer^x* corpus as our source to extract sentential patterns.

These patterns are then used at run-time in *WriteAhead* for assisted writing. *WriteAhead* takes the move tag the user types in, and then retrieves, and displays patterns related to the tag to help the user write or edit a draft (Figure 1). We present a new methodology for automatically deriving patterns. WriteAhead is also the first interface that suggests patterns for learners while they type.

The rest of the paper is organized as follows. The related work is reviewed in the next section. Then we present our method for automatically extracting sentential patterns and examples (Section 3). As part of our evaluation, we measured the accuracy rate of suggestions generated by *WriteAhead* using published research papers unrelated to the training data (Section 4). Section 5 reports on the experiment results and we summarize our conclusion in Section 6.

2   Related Work

Researchers have shown that non-native student writers may have difficulties in composing sentences
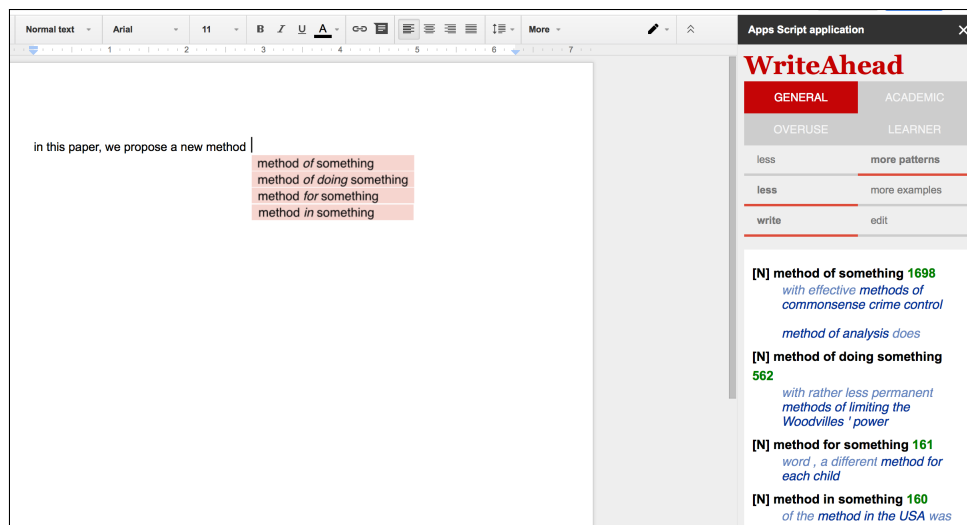
Figure 2: WriteAhead implementation for Google Doc.

and lack knowledge at discourse level (Connor, 1996 & 1999) in academic writing.

English for Academic Purposes (EAP) plays an important role in Specific Purposes (ESP) study, focusing on English of academic writing (EAW). EAW consists in numerous academic genres, including grant proposal (Connor, 1999), research articles (Swales, 1990), and reviews, which involve manual structure analysis in the academic texts for teaching academic writing. Among them, research articles (RAs) play the most significant role. In our work, we use RAs as our corpus to generate sentential patterns, which can assist learners in academic writing. We also adopt a set of semantic categories to generate patterns, which were manually identified in Teufel (1999).

2.1 Analysis of the Structure of Research Articles

More specifically, we focus on the structure of research articles, namely, automatically analyzing the abstracts based on series of moves. The sentences are classified to match the predefined structure. The most related body analyzing research article was Hill et al. (1982). The scheme he proposed was a starting-point for the analysis of the macrostructure of research articles. The graphical illustration of his proposed structure is like an hourglass.

Several research indicate that RAs, defined by Swales (1990), have a simpler and more clear picture of the organizational pattern – the IMRD structure: Introduction, Method, Results and Discussion. Additionally, Swales (1981, 1990) proposed the CARS model ("Create a Research Space") which describes the structure of the typical introductions of scientific articles according to prototypical rhetorical building plans.

The unit of analysis is the argumentative *move*, which represents "a semantic unit related to the writer's purpose", typically, one clause or sentence long. There is a finite number of such moves, and they are subdivided into "steps". The model is schematically depicted in Figure 3. The model has been used extensively by discourse analysis and researchers in the field of discourse structure. Many studies adopted his theory to analyze the introduction section (e.g. Cooper, 1985; Hopkins, 1985; Crookes, 1986; Samraj, 2002, 2005). Additionally, Thompson (1993) applied it to analyze the result section while others applied it to investigate the discussion section (e.g. Hopkins and Dudley-Evans, 1994). More researches have been done to study RAs in recent years.

2.2 Identifying Moves for Text Classification

In the search area of automatic analysis of the discourse structure of research articles, in recent years, much work has been done viewing the task as a text classification problem that determines a label (move name) for each sentence. Various classi-

Move 1: Establishing a territory	Step 1 Claiming centrality and/or
	Step 2 Making topic generalization(s) and/or
	Step 3 Reviewing items of previous research
Move 2: Establishing a niche	Step 1A Counter-claiming or
	Step 1B Indicating a gap or
	Step 1C Question-raising or
	Step 1D Continuing a tradition
Move 3: Occupying the niche	Step 1A Outlining purposes or
	Step 1B Announcing present research
	Step 2 Announcing principal findings
	Step 3 Indicating RA structure

Figure 3: Structure of research article introduction (Swales, 1990)

fiers were applied to text categorization, including Naive Bayesian Model (NBM) (Teufel and Moens, 2002, 2004, 2006; Anthony 2003), Support Vector Machines (SVM) (McKnight and Arinivasan, 2003; Shimbo et al., 2003; Yamamoto and Takagi, 2005), Hidden Markov Model (HMM) (Lin et al., 2006), and Conditional Random Fields (CRFs) (Hirohata et al. 2008). Table 1 summarizes these approaches.

Table 1 shows the set of labels commonly used in most studies: background (B), objective (O), purpose(P), gap (G), method (M), result (R),and conclusion (C). We did not compare directly the performances of these studies, which used a different set of classification labels and evaluation data.

Anthony (2003) has developed a system which could offer a *move* analysis to assist students in writing and reading. He used the CARS model to analyze the abstracts of RA, using hand tagged RA abstracts. Shimbo et al. (2003) presented an advanced text retrieval system for MEDLINE that provides zone search specific sections in abstracts. The system classifies sentences in each Medline abstract into four sections: objective, method, results, and conclusion. Each sentence is represented by words, word bigrams, and contextual information of the sentences (e.g., class of the previous sentence, relative location of the current sentence). They reported 91.9% accuracy (per-sentence basis) and 51.2% accuracy (per-abstract basis) for the classification with the best feature set for quadratic SVM.

Similarly, Yamamoto and Takagi (2005) developed a system to classify abstract sentences into five moves, background, purpose, method, result, and conclusion. They trained a linear-SVM classifier

with features of unigram, subject-verb, verb tense, relative sentence location, and sentence score. Hirohata et al (2008) presented another system for Medline abstracts into four moves. They trained a CRF classifier with features of n-grams, sentence location, and features from previous/next sentences.

3 Method

Writing academic paper by referencing examples (e.g., *We illustrate the method* ...) often does not work very well, because learners may fail to generalize from examples and apply them to their own situations. Often, there are too many examples to choose from and to adapt to match the need of the learner writers. To help the learner in writing, a promising approach is to extract a set of representative sentential patterns consisting of keywords and categories that are expected to assist learners to write better.

3.1 Problem Statement

We focus on the extracting process of sentential patterns for various rhetoric functions: identifying a set of candidate patterns with keywords and categories. These candidate patterns are then statistically analyzed, filtered and finally returned as the output of the system. The returned patterns can be directly examined by the learner, alternatively they can be used to annotate rhetoric moves. Thus, it is crucial that the extracted patterns cover all semantic categories of interest. At the same time, the set of extracted patterns of a semantic category cannot be too large that it overwhelms the writer or the tagging process of the subsequent move. Therefore, our goal is to return a reasonable-sized set of sentential patterns that, at the same time, must cover all rhetoric moves. We now formally state the problem that we are addressing.

Problem Statement: We are given a raw corpus *CORP* (e.g., *Citeseer^x*) as well as an annotated corpus *TAGGED-CORP* in a specific genre and domain, and a list of semantic categories (e.g., *PAPER* = { *paper, article*}, *PRESENT* = { *present, describe, introduce* }). Our goal is to retrieve a set of tagged sentential patterns, p_1 , ... , p_m, consisting of keywords and categories from *CORP*. For this, we convert all sentences in *CORP* and *TAGGED-CORP* into candidate patterns (e.g., *In this PAPER, we PRESENT*

Table 1: Approaches of previous studies

Researchers	Model	Moves	Data
Macmillan English Dictionary	—	12 moves$^+$	general writing
Teufel and Moens (2002, 2004, 2006)	Naive Bayes	7 moves$^+$	scientific papers
Anthony (2003)	Naive Bayes	BPGMRC	scientific papers
McKnight and Srinivasan (2003)	Support Vector Machine	OMRC	MEDLINE
Shimbo et al. (2003)	Support Vector Machine	OMRC	MEDLINE
Yamamoto and Takagi (2005)	Support Vector Machine	BPMRC	MEDLINE
Wu et al. (2006)	Hidden Markov Model	BPMRC	Citeseer
Lin et al. (2006)	Hidden Markov Model	OMRC	MEDLINE
Hirohata et al. (2008)	Conditional Random Field	OMRC	MEDLINE

* ADD, COMPARE, EXAM, CAUSE, OPIONION, HEDGE, TOPIC, LIST, REPHRASE, REPORT, SUM-UP
+ AIM, TEXTUAL, OWN, BACKGROUND, CONTRAST, BASIS and OTHER

- **Procedure ExtractPatterns**(*Sent*, *Categories*, *Corpus*)

   1. Extract candidate patterns from sentences in *CORP* (Section 3.2.1)

   2. Group patterns by semantic categories in the given corpus (Section 3.2.2)

   3. Generate sentential patterns by statistically analyzing and filtering candidate patterns (Section 3.2.3)

   4. Output characteristic patterns for each category

Figure 4: Outline of the pattern extraction process

WORK), such that these candidates can be statistically analyzed and filtered to generate common and representative patterns.

In the rest of this section, we describe our solution to this problem. First, we define a strategy for transforming sentences from academic corpora into candidate patterns (Section 3.2.1). This strategy relies on a set of candidate patterns derived from sentences of patterns (which we will describe in detail in Section 3.2.3). In this section, we also describe our method for extracting the most representative of the candidate patterns for each semantic category of interest. Finally, we show how *WriteAhead* displays patterns at run-time (Section 3.3).

3.2 Transforming Sentences into Patterns

We attempt to find transformations from sentences into patterns, consisting of keywords and categories expected to characterize rhetoric moves in academic writings. Our learning process is shown in Figure 4.

Procedure SPs&Examples (*Sentences*, *Templates*)

(1) *taggedCorpus* = ChunkParser(*Sentences*)

(2) *candidates* = GenPatternCandidate(*taggedCorpus*)

(3) *patternInstances* = ReplaceTeufel(*candidates*)

(4) *Pats*, *Categories*, *Instances* = GroupByCategory(*patternInstances*)

(5) *Pats*, *Counts* = Counter(*Patterns*, *Instances*)

(6) *Avg*, *STD* = CalStatics(*Pats*, *Counts*)

   For each *Pat*, *Count* pair in (*Pats*, *Counts*)

      If *Count* > *Avg* + *MinSTDThreshold* × *STD*

(7)   Emit *Tuple* = (*Word*, *Pat*, *PatTuples*)

(8) *Pats* = Annotate(*Word*, *Pat*, *PatTuples*)

Figure 5: Process for extracting SPs and examples.

3.2.1 *Extracting Candidate Patterns.*

In the first stage of the extracting process (Step (1) in Figure 5), we tokenize sentences in the given corpus, and assign to each word its syntactic information including lemma, part of speech, and phrase group (represented using the B-I-O notation to mark the beginning, inside, and outside of some phrase group).

See Table 2 for an example of tagged sentence. In order to identify the head of a phrase, we convert the B-I-O notation to I-H-O notation with *H* denoting the headword of a phrase. Using the I-H-O notation allows us to directly identify the headword of a phrase chunk. Then, we convert every word in a sentence into elements of a candidate pattern. The

Table 2: A tagged sentence, pattern elements for each word, and anchored pattern candidates

Word	Lemma	POS	B-I-O	I-H-O	Element	Pattern candidate anchored at each word
In	In	IN	B-PP	I-PP	in	(In)
this	this	DT	B-NP	H-NP	this	(In this)
report	report	NN	I-NP	H-NP	PAPER	(In this PAPER)
,	,	,	O	O	,	(In this PAPER ,)
we	we	PRP	B-NP	I-NP	we	(In this PAPER , we)
propose	propose	VBP	B-VP	H-VP	PRESENT	(In this PAPER , we PRESENT)
a	a	DT	B-NP	I-NP	(_)	(In this PAPER , we PRESENT _)
method	method	NN	I-NP	H-NP	WORK	(In this PAPER , we PRESENT _ WORK)

Table 3: Example Teufel category of sentential patterns

Teufel category	Gloss and examples
PAPER	article, draft, paper, project, report, ...
WORK	analysis, approach, method, ...
PURPOSE	aim, goal, purpose, task, theme, topic, ...
DEVELOP	accomplish, achieve, answer, prove, ...
PRESENT	describe, introduce, present, propose, ...
EVALUATE	compare, compete, evaluate, test, ...

elements consist of three types of information:

- **Semantic categories** (See Table 3) : typical domain specific concepts and words,
- **Lexical symbols**: a list of common prepositions, pronouns, adverbs, and determinants,
- **Noun phrase** and **verb phrase**: head words that are not classified in a category are represented as *something* or *do*.

Note that determinants (e.g., the, an, a) or adjectives need to be represented in a pattern. For those words, we add "_" (*ignored*) to the element list. The *ignored* elements will be deleted before patterns are analyzed and filtered (as shown in Table 2). We design the scope of extracted patterns, as from the beginning of the sentence, to the object phrase after the main verb.

Finally, we combine elements for a sentence to generate pattern candidates (Step (2) in Figure 5). Table 2 shows those elements associated with words and how they combine to form pattern candidates.

In Step (3), we use semantic categories to generalize words and generate formulaic patterns. As will be described in Section 4, we used a Teufel manually analyzed research article to device a set of categories of words (Teufel, 1999).

For example, in Table 2, the sentence "*In this paper, we propose a method that accurately reports timing information by accounting for intrusion introduced by monitoring*." will be transformed into the candidate pattern "*In this PAPER, we PRESENT WORK*".

The input to this state is a set of sentences. These sentences constitute the data for generating the candidate patterns, that can be used in the next step.

The output of this stage is a set of candidate patterns that can be statistically analyzed and filter in a later step. See Table 4 for example candidate patterns extracted from some sentences.

3.2.2 *Grouping Patterns by Categories.*

In the second stage of the process (Step (4) in Figure 5), we filter candidate patterns to generate representative patterns. Once patterns and instances are generated, they are sorted and grouped by category.

Then, we count the number of instances of each pattern within the category (in Step (5)), and the average and standard deviation of these counts for each category (in Step (6)).

In Step (7), we select patterns with an instance count exceeding the average count by *MinSTDThreshold* standard deviation.

Consider the partial sentence "*In this paper, we propose a method*" Table 2 shows elements of each word, pattern candidates anchored each word. Note that the candidate (e.g., **In this PAPER, we PRESENT WORK** associated with the instance of *In this paper, we propose a method*) are valid patterns.

Table 4: Ranked patterns based on *CORP* and *TAGGED-CORP* statistics

AZ	Pattern	Count	Example
AIM	In this PAPER , we PROPOSE	100,607	In this paper , we present/propose/introduce In this paper , we describe
AIM	In this PAPER we PROPOSE	48,354	In this paper we present/propose/introduce In this paper we describe/discuss
AIM	The AIM of this PAPER	20,725	The purpose/aim/goal of this paper The aim/purpose of this study
AIM	In this PAPER , we INVESTIGATE	16,872	In this paper , we investigate/examine/identify In this paper , we analyze

3.2.3 *Ranking and Annotating Patterns.*

In the third and final stage (Step (8) in Figure 5), we count, sort, and filter patterns, essentially using the frequency counts from *CORP* with the tags in *TAGGED-CORP* (See Tables 4). Figure 5 shows the algorithm for ranking a set of corresponding sentential patterns for all semantic categories. See Table 6 for an example of the move tag *AIM* and its corresponding sentential patterns.

3.3 Run-Time

Once the patterns and examples are automatically extracted for each category in the given corpus, they are stored and indexed by category that can be annotated with corresponding rhetoric moves. At run-time in a writing session, *WriteAhead* detects a rhetoric move tag *Move* in the text box. With the tag as a query, *WriteAhead* retrieves and sorts all relevant patterns and examples (*Pattern* and *Example*) by frequency, aiming to display the most common information toward the top.

4 Implementation and Setting

In this section, we describe the implementation and experiments of the method presented in Section 3. First, we retrieved computer science abstracts from the digital library website, *CiteSeer* (`citeseerx. ist.psu.edu`, a collection of bibliographies of scientific literature in computer science from various sources). We obtained about four million computer science abstracts. Before extracting patterns, we use GeniaTagger as a simple toolkit for analyzing and parsing English sentences and outputting the base forms, part-of-speech tags, chunk tags and named

entity tags. Tsuruoka et al. (2005) develop this tools, specifically tuned for biomedical text. Then, we use this tagger for tagging sentences to obtain the part-of-speech tags, lemma, chunks for training the model. After tagging, we applied our method to extract patterns from *CiteSeer x*.

We used manually compiled semantic categories and words (Teufel, 1999) to generalize word and generate formulaic patterns. There are 66 categories with some 900 nouns, verbs, and adjectives.

For example, in Table 2, the sentence "*In this paper, we propose a method that accurately reports timing information by accounting for intrusion introduced by monitoring.*" will be transformed into the candidate pattern "*In this PAPER, we PRESENT WORK*". As will be described in Section 4, we used a Teufel manually analyzed research article to derive a set of categories of words (Teufel, 1999).

We also used the Argumentative Zoning corpus (available at `www.cl.cam.ac.uk/~sht25/ AZ_corpus.html`), created and annotated by Teufel (2010) and collaborators. The dataset consists of 80 AZ-annotated conference articles in the research area of computational linguistics, hosted in the academic archive of Cmplg arXiv. The set of AZ tags include AIM (purpose), BAS (basis), BKG (background), CTR (contrast), OTH (previous work), OWN (own method and results), TXT (textual references).

To provide user interface for accessing *WriteAhead*, we have implemented two versions of the system: (1) browser-based proof-of-concept (POC) interface and a Google Docs add-on. (2) To prove the feasibility of the concept, the browser version (with-

out the common editing functions) was implemented in Python and within the Flask Web framework. We stored the suggestions in JSON format using PostgreSQL for a faster access. The *WriteAhead* server obtains client input from a popular browser (Safari, Chrome, or Firefox) dynamically with AJAX techniques. For uninterrupted service and ease of scaling up, we chose to host *WriteAhead* on Heroku, a cloud-platform-as-a-service (PaaS) site.

The *WriteAhead* add-on for Google Docs was implemented in *Google App Script (GAS) with HTML and JavaScript. WriteAhead* add-on obtains client input from documents using built-in methods in GAS, and obtains the suggestions by sending requests to the *WriteAhead* server through our API. To start the add-on, the user click "Add-ons > WriteAhead > Start" after installation, and a sidebar will appear. The suggested patterns and examples are shown in the sidebar.

5   Evaluation

The proposed system was designed to automatically extract patterns and examples for each corresponding rhetoric move. A preliminary evaluation was done on a set of real formula labeled with moves. We compared our patterns with the Teufel formula and evaluated the precision in different experimental settings. In this Section, we first describe how we compared patterns (Section 4). Then, Section 5.1 introduces the evaluation metrics for evaluating the experimental results.

5.1   Evaluation Metrics

The output of our method is an automatically tagged pattern, which can either be shown to the user directly, or be used in academic writing, e.g., in teaching academic writing, teachers can use those tagged patterns to help tag the sentences in a given RA with moves.

To evaluate our approach, we compare our sentential patterns with Teufel Formula. We extract three categories PAPER, WORK, and PRESENT to inspect that if there are some patterns in common.

We compare Teufel formulas and our extracted sentential patterns under three specific categories, PAPER, WORK, and PRESENT. In PAPER category, there are 76 formula and 5 patterns, among

these, 18 formula and 5 patterns are in common. Similar in WORK category, there are 153 formula and 16 patterns, among these, 30 formula and 16 patterns are in common. And in PRESENT category, there are 15 formula and 13 patterns, among these, 7 formula and 5 patterns are in common.

In comparison, our sentential patterns are longer and more complete than Teufel's original formulas serving as features for classification. It is clear that, our sentential patterns tend to be more complete and indicative of rhetoric moves, since we use the stopping condition to extract patterns all the way up to the object of the main verb. As it turns out, the patterns generate by a computer using a very large scale academic corpus are more consistent, complete, and relevant to academic writing.

6   Conclusion and Future Work

Many avenues exist for future research and improvement of our system. For example, the method for extracting patterns could be discussed further and be evaluated separately, using different formula to calculate a threshold, and generate different patterns. Additionally, an interesting direction to explore is to expand the word categories and obtain more fine-grain patterns. Yet another direction of this research would be applied to the model of different sections of RAs, and more disciplines.

In summary, we have presented a new method for extracting patterns in a scholar big dataset for various moves in academic writing. The method involves patterns with all semantic categories of interest. The experimental results show that our automatically extracted patterns reflect different rhetoric moves and purposes.

References

Anthony, Laurence and Lashkia, George V. 2003. Mover: A machine learning tool to assist in the reading and writing of technical papers. *Transactions on Professional Communication, IEEE.* Vol 46,3, pages 185–193, IEEE.

Connor, Ulla. 1996. Contrastive rhetoric: Cross-cultural aspects of second language writing. *Cambridge University Press.*

Connor, Ulla and Mauranen, Anna. 1999. Linguistic analysis of grant proposals: European Union research

grants. *English for specific purposes.* Vol. 18, 1, pages 47–62, Elsevier.

Cooper, Catherine. 1985. Aspects of article introductions in IEEE publications. *Unpublished master's thesis, University of Aston, Birmingham, England.*

Crookes, Graham. 1986. Towards a validated analysis of scientific text structure. *Applied linguistics, Am Association of Applied Linguistics.* Vol. 7, 1, pages 57–70.

Dudley-Evans, Tony. 1994. Genre analysis: An approach to text analysis for ESP. *Advances in written text analysis.* Vol. 219, page 228.

Graddol, David. 1997. The future of English?: A guide to forecasting the popularity of the English language in the 21st century. *British Council.*

Hill, Susan S and Soppelsa, Betty F and West, Gregory K. 1982. Teaching ESL Students to Read and Write Experimental-Research Papers. *TESOL quarterly.* Vol. 16, 3, pages 333–347, Wiley Online Library.

Hinds, John. 1990. Inductive, deductive, quasi-inductive: Expository writing in Japanese, Korean, Chinese, and Thai. *Coherence in writing: Research and pedagogical perspectives,* pages 87–110.

Hirohata, Kenji and Okazaki, Naoaki and Ananiadou, Sophia and Ishizuka, Mitsuru and Biocentre, Manchester Interdisciplinary. 2008. Identifying Sections in Scientific Abstracts using Conditional Random Fields. *IJCNLP,* pages 381–388.

Hopkins, Andy. 1985. An investigation into the organizing and organizational features of published conference papers. *Unpublished MA dissertation, University of Birmingham.*

opkins, Andy and Dudley-Evans, Tony. 1988. A genre-based investigation of the discussion sections in articles and dissertations. *English for Specific Purposes.* Vol. 7, 2, pages 113–121, Elsevier.

Lin, Jimmy and Karakos, Damianos and Demner-Fushman, Dina and Khudanpur, Sanjeev. 2006. lin2006generative. *Association for Computational Linguistics,* pages 65–72.

McKnight, Larry and Srinivasan, Padmini. 2003. Categorization of sentence types in medical abstracts. *American Medical Informatics Association Annual Symposium Proceedings 2003,* page 440.

Ruiying, Yang and Allison, Desmond. 2003. Research articles in applied linguistics: Moving from results to conclusions. *English for Specific Purposes.* Vol. 22, 4, pages 365–385.

Samraj, Betty. 2002. Introductions in research articles: Variations across disciplines. *English for specific purposes.* Vol. 21, 1, pages 1–17, 2002. Elsevier.

Shimbo, Masashi and Yamasaki, Takahiro and Matsumoto, Yuji. 2003. Using sectioning information for text retrieval: a case study with the medline abstracts.

Proceedings of Second International Workshop on Active Mining (AM'03).

Swales, John. 1990. Genre analysis: English in academic and research settings. *Cambridge University Press.*

Teufel, S., Carletta, J. & Moens, M. 1999. An annotation scheme for discourse-level argumentation in research articles. *EACL.*

Teufel, Simone. 2000. Argumentative zoning: Information extraction from scientific text. *Diss. University of Edinburgh.*

Teufel, Simone and Moens, Marc. 2002. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational linguistics.* Vol.28, pages 409–445. MIT Press.

Teufel, Simone. 2010. The Structure of Scientific Articles: Applications to Citation Indexing and Summarization (Center for the Study of Language and Information-Lecture Notes). *Center for the Study of Language and Inf.*

Thompson, Dorothea K. 1993. Arguing for Experimental Facts in Science A Study of Research Article Results Sections in Biochemistry. *Written communication.* Vol. 10, 1, pages 106–128, Sage Publications.

Tsuruoka, Yoshimasa and Tateishi, Yuka and Kim, Jin-Dong and Ohta, Tomoko and McNaught, John and Ananiadou, Sophia and Tsujii, Junichi. 2005. Developing a robust part-of-speech tagger for biomedical text. *Advances in informatics,* pages 382–392, Springer.

Weil, Ben H. 1970. Standards for writing abstracts. *Journal of the American Society for Information Science.* Vol. 1, 5,pages 351–357, Wiley Online Library.

Yamamoto, Yasunori and Takagi, Toshihisa. 2005. A sentence classification system for multi biomedical literature summarization. *Data Engineering Workshops, 2005. 21st International Conference on Biomedical Data Engineering,* pages 1163–1163, IEEE.