Computational Analysis of Move Structures in Academic Abstracts

Jien-Chen Wu[1] **Yu-Chia Chang**[1] **Hsien-Chin Liou**[2] **Jason S. Chang**[1]
CS[1] and FLL[2], National Tsing Hua Univ.
{d928322,d948353}@oz.nthu.edu.tw, hcliu@mx.nthu.edu.tw, jason.jschang@gmail.com

Abstract

This paper introduces a method for computational analysis of move structures in abstracts of research articles. In our approach, sentences in a given abstract are analyzed and labeled with a specific move in light of various rhetorical functions. The method involves automatically gathering a large number of abstracts from the Web and building a language model of abstract moves. We also present a prototype concordancer, CARE, which exploits the move-tagged abstracts for digital learning. This system provides a promising approach to Web-based computer-assisted academic writing.

1 Introduction

In recent years, with the rapid development of globalization, English for Academic Purposes has drawn researchers' attention and become the mainstream of English for Specific Purposes, particularly in the field of English of Academic Writing (EAW). EAW deals mainly with genres, including research articles (RAs), reviews, experimental reports, and other types of academic writing. RAs play the most important role of offering researchers the access to actively participating in the academic and discourse community and sharing academic research information with one another.

Abstracts are constantly regarded as the first part of RAs and few scholarly RAs go without an abstract. “A well-prepared abstract enables readers to identify the basic content of a document quickly and accurately.” (American National Standards Institute, 1979) Therefore, RAs' abstracts are equally important to writers and readers.

Recent research on abstract requires manually analysis, which is time-consuming and labor-intensive. Moreover, with the rapid development of science and technology, learners are increasingly engaged in self-paced learning in a digital environment. Our study, therefore, attempts to investigate ways of automatically analyzing the move structure of English RAs' abstracts and develops an online learning system, CARE (Concordancer for Academic wRiting in English). It is expected that the automatic analytical tool for move structures will facilitate non-native speakers (NNS) or novice writers to be aware of appropriate move structures and internalize relevant knowledge to improve their writing.

2 Macrostructure of Information in RAs

Swales (1990) presented a simple and succinct picture of the organizational pattern for a RA—the IMRD structure (Introduction, Methods, Results, and Discussion). Additionally Swales (1981, 1990) introduced the theory of genre analysis of a RA and a four-move scheme, which was later refined as the "Create a Research Space" (CARS) model for analyzing a RA’s introduction section.

Even though Swales seemed to have overlooked the abstract section, in which he did not propose any move analysis, he himself plainly realized “abstracts continue to remain a neglected field among discourse analysts” (Swales, 1990, p. 181). Salager-Meyer (1992) also stated, “Abstracts play such a pivotal role in any professional reading” (p. 94). Seemingly researchers have perceived this view, so research has been expanded to concentrate on the abstract in recent years.

Anthony (2003) further pointed out, “research has shown that the study of rhetorical organization or structure of texts is particularly useful in the technical reading and writing classroom” (p. 185). Therefore, he utilized computational means to create a system, *Mover*, which could offer move analysis to assist abstract writing and reading.

3 CARE

Our system focuses on automatically computational analysis of move structures (i.e.

Background, **P**urpose, **M**ethod, **R**esult, and **C**onclusion) in RA abstracts. In particular, we investigate the feasibility of using a few manually labeled data as seeds to train a Markov model and to automatically acquire move-collocation relationships based on a large number of unlabeled data. These relationships are then used to analyze the rhetorical structure of abstracts. It is important that only a small number of manually labeled data are required while much of move tagging knowledge is learned from unlabeled data. We attempt to identify which rhetorical move is correspondent to a sentence in a given abstract by using features (e.g. collocations in the sentence). Our learning process is shown as follows:

(1)Automatically collect abstracts from the Web for training

(2)Manually label each sentence in a small set of given abstracts

(3)Automatically extract collocations from all abstracts

(4)Manually label one move for each distinct collocation

(5)Automatically expand collocations indicative of each move

(6)Develop a hidden Markov model for move tagging

Figure 1: Processes used to learn collocation classifiers

3.1 Collecting Training Data

In the first four processes, we collected data through a search engine to build the abstract corpus *A*. Three specialists in computer science tagged a small set of the qualified abstracts based on our coding scheme of moves. Meanwhile, we extracted the collocations (Jian et al., 2004) from the abstract corpus, and labeled these extracted collocations with the same coding scheme.

3.2 Automatically Expanding Collocations for Moves

To balance the distribution in the move-tagged collocation (*MTC*), we expand the collocation for certain moves in this stage. We use the one-move-per-collocation constraint to bootstrap, which mainly hinges on the feature redundancy of the given data, a situation where there is often evidence to indicate that a given should be annotated with a certain move. That is, given one collocation c_i is tagged with move m_i, all sentences *S* containing collocation c_i will be tagged with m_i as well; meanwhile, the other collocations in *S* are thus all tagged with m_i. For example:

Step 1. The collocation "paper address" extracted from corpus *A* is labeled with the "P" move. Then we use it to label other untagged sentences *US* (e.g. Examples (1) through (2)) containing "paper address" as "P" in *A*. As a result, these *US* become tagged sentences *TS* with "P" move.

(1)This **paper addresses** the state explosion problem in automata based ltl model checking. //P//
(2)This **paper addresses** the problem of fitting mixture densities to multivariate binned and truncated data. //P//

Step 2. We then look for other features (e.g. the collocation, "address problem") that occur in *TS* of *A* to discover new evidences of a "P" move (e.g. Examples (3) through (4)).

(3)This paper **addresses** the state explosion **problem** in automata based ltl model checking.
(4)This paper **addresses** the **problem** of fitting mixture densities to multivariate binned and truncated data.

Step 3. Subsequently, the feature "address problem" can be further exploited to tag sentences which realize the "P" move but do not contain the collocation "paper address", thus gradually expanding the scope of the annotations to *A*. For example, in the second iteration, Example (5) and (6) can be automatically tagged as indicating the "P" move.

(5)In this paper we **address** the **problem** of query answering using views for non-recursive data log queries embedded in a Description Logics knowledge base. //P//
(6)We **address** the **problem** of learning robust plans for robot navigation by observing particular robot behaviors. //P//

From these examples ((5) and (6)), we can extend to another feature "we address", which can be tagged as "P" move as well. The bootstrapping processes can be repeated until no new feature with high enough frequency is found (a sample of collocation expanded list is shown in Table1).

Type	Collocation	Move	Count of Collocation with m_i	Total of Collocation Occurrences
NV	we present	P	3,441	3,668
NV	we show	R	1,985	2,069
NV	we propose	P	1,722	1,787
NV	we describe	P	1,505	1,583
…	…	…	…	…

Table 1: The sample of the expanded collocation list

3.3 Building a HMM for Move Tagging

The move sequence probability $P(t_i+1 \mid t_i)$ is given as the following description:

We are given a corpus of unlabeled abstracts $A = \{A_1,\ldots, A_N\}$. We are also given a small labeled subset $S = \{L_1,\ldots, L_k\}$ of A, where each abstract L_i consists of a sequence of sentence and move $\{t_1, t_2,\ldots, t_k\}$. The moves t_i take out of a value from a set of possible move $M = \{m_1,m_2,\ldots,m_n\}$.

Then $P(t_{i+1} \mid t_i) = \left(\dfrac{N(t_i \mid t_{i+1})}{N(t_i)} \right)$

According to the bi-gram move sequence score (shown in Table 2), we can see move sequences follow a certain schematic pattern. For instance, the "B" move is usually directly followed by the "P" move or "B" move, but not by the "M" move. Also rarely will a "P" move occur before a "B" move. Furthermore, an abstract seldom have a move sequence wherein "P" move directly followed by the "R" move, which tends to be a bad move structure. In sum, the move progression generally follows the sequence of "B-P-M-R-C".

Move t_i	Move t_{i+1}	- log P (t_{i+1}|t_i)
$	B	0.7802
$	P	0.6131
B	B	0.9029
B	M	3.6109
B	P	0.5664
C	$	0.0000
M	$	4.4998
M	C	1.9349
M	M	0.7386
M	R	1.0033
P	M	0.4055
P	P	1.1431
P	R	4.2341
R	$	0.9410
R	C	0.8232
R	R	1.7677

Table 2: The score of bi-gram move sequence (Note that "$" denotes the beginning or the ending of a given abstract.)

Finally, we synchronize move sequence and one-move-per-collocation probabilities to train a language model to automatically learn the relationship between those extracted linguistic features based on a large number of unlabeled data. Meanwhile, we set some parameters of the proposed model, such as, the threshold of the number of collocation occurring in a given abstract, the weight of move sequence and collocation and smoothing. Based on these parameters, we implement the Hidden Markov Model (HMM). The algorithm is described as the following:

$$p(s_1,\ldots,s_n) = p(t_1)p(s_1 \mid t_1)\Pi p(t_i \mid t_{i-1})p(s_i \mid t_i)$$

The moves t_i take out of a value from a set of possible moves $M=\{m_1, m_2, \ldots, m_k\}$ (The following parameters θ_1 and θ_2 will be determined based on some heuristics).

$p(S_i \mid t_i = m_i)$
$= \theta_1$ if S_i contains a collocation in MTC_j $i = j$
$= \theta_2$ if S_i contains a collocation in MTC_j but $i \neq j$
$= \dfrac{1}{k}$ if S_i does not contain a collocation MTC_j

The optimal move sequence t* is

$$(t_1^*, t_2^*, \ldots, t_n^*) = \underset{t_1,t_2,\ldots,t_n}{\arg\max}\, p(s_1,\ldots,s_n \mid t_i,\ldots,t_n)$$

In summary, at the beginning of training time, we use a few human move-tagged sentences as seed data. Then, collocation-to-move and move-to-move probabilities are employed to build the HMM. This probabilistic model derived at the training stage will be applied at run time.

4 Evaluation

In terms of the training data, we retrieved abstracts from the search engine, *Citeseer*; a corpus of 20,306 abstracts (95,960 sentences) was generated. Also 106 abstracts composed of 709 sentences were manually move-tagged by four informants. Meanwhile, we extracted 72,708 collocation types and manually tagged 317 collocations with moves.

At run time, 115 abstracts containing 684 sentences were prepared to be the training data. We then used our proposed HMM to perform some experimentation with the different values of parameters: the frequency of collocation types, the number of sentences with collocation in each abstract, move sequence score and collocation score.

4.1 Performance of CARE

We investigated how well the HMM model performed the task of automatic move tagging under different values of parameters. The parameters involved included the weight of transitional probability function, the number of sentences in an abstract, the minimal number of instance for the applicable collocations. Figure 2 indicates the best precision of 80.54% when 627 sentences were qualified with the set of various parameters, including 0.7 as the weight of transitional probability function and a frequency threshold of 18 for a collocation to be applicable, and the minimally two sentences containing an applicable collocation. Although it is important to have many collocations, it is crucial that we set an appropriate frequency threshold of collocation so as not to include unreliable collocation and lower the precision rate.

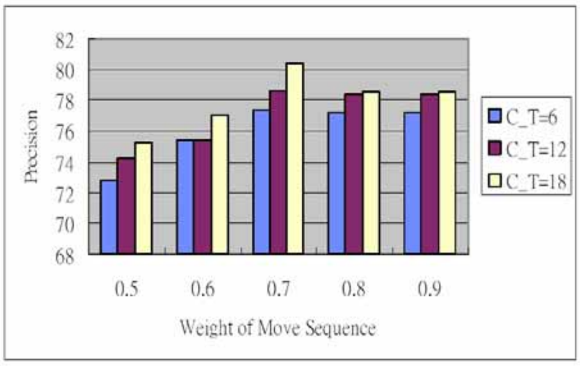


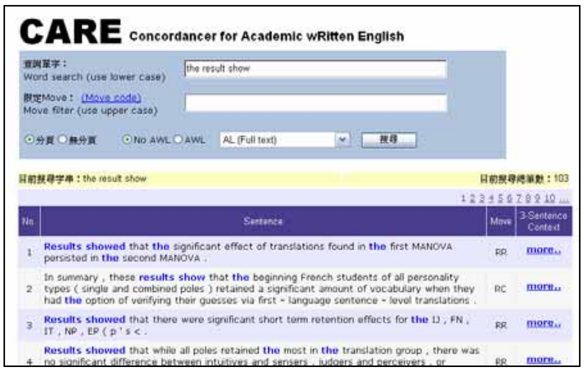
Figure2: The results of tagging performance with different setting of weight and threshold for applicable collocations (Note that C_T denotes the frequency threshold of collocation)

5 System Interface

The goal of the CARE System is to allow a learner to look for instances of sentences labeled with moves. For this purpose, the system is developed with three text boxes for learners to enter queries in English (as shown in Figure3.):

- Single word query (i.e. directly input one word to query)
- Multi-word query (i.e. enter *the result show* to find citations that contain the three words, "*the*", "*paper*" and "*show*" and all the derivatives)
- Corpus selection (i.e. learners can focus on a corpus in a specific domain)

Once a query is submitted, CARE displays the results in returned Web pages. Each result consists of a sentence with its move annotation. The words matching the query are highlighted.



CARE Concordancer for Academic wRitten English

查詢單字：
Word search (use lower case): the result show

限定Move： (Move code)
Move filter (use upper case):

[✓ 分頁] 無分頁 [✓ No AWL] AWL AL (Full text) 搜尋

目前搜尋字串：the result show 目前搜尋總筆數：103

1 2 3 4 5 6 7 8 9 10 ...

No	Sentence	Move	3-Sentence Context
1	**Results showed** that **the** significant effect of translations found in **the** first MANOVA persisted in **the** second MANOVA .	RR	more...
2	In summary , these **results show** that **the** beginning French students of all personality types (single and combined poles) retained a significant amount of vocabulary when they had **the** option of verifying their guesses via first - language sentence - level translations .	RC	more...
3	**Results showed** that there were significant short term retention effects for **the** IJ , FN , IT , NP , EP (p ' s < .	RR	more...
4	**Results showed** that while all poles retained **the** most in **the** translation group , there was no significant difference between intuitives and sensers , judgers and perceivers , or	RR	more...

Figure 3: The sample of searching result with the phrase "the result show"

6 Conclusion

In this paper, we have presented a method for computational analysis of move structures in RAs' abstracts and addressed its pedagogical applications. The method involves learning the inter-move relationships, and some labeling rules we proposed. We used a large number of abstracts automatically acquired from the Web for training, and exploited the HMM to tag sentences with the move of a given abstract. Evaluation shows that the proposed method outperforms previous work with higher precision. Using the processed result, we built a prototype concordance, CARE, enriched with words, phrases and moves. It is expected that NNS can benefit from such a system in learning how to write an abstract for a research article.

References

Anthony, L. and Lashkia, G. V. 2003. Mover: A machine learning tool to assist in the reading and writing of technical papers. *IEEE Trans. Prof. Communication*, 46:185-193.

American National Standards Institute. 1979. *American national standard for writing abstracts*. ANSI Z39, 14-1979. New York: Author.

Jian, J. Y., Chang, Y. C., and Chang, J. S. 2004. TANGO: Bilingual Collocational Concordancer, Post & demo in ACL 2004, Barcelona.

Salager-Meyer, F. S. 1992. A text-type and move analysis study of verb tense and modality distribution in medical English abstracts. *English for Specific Purposes*, 11:93-113.

Swales, J.M. 1981. *Aspects of article introductions. Birmingham*, UK: The University of Aston, Language Studies Unit.

Swales, J.M. 1990. *Genre analysis: English in Academic and Research Settings*. Cambridge University Press.