Computer Assisted
Language Instruction
Consortium

CALICO JOURNAL (ONLINE) ISSN 2056–9017

Article

Discourse Classification into Rhetorical Functions for AWE Feedback

Elena Cotos and Nick Pendar

Abstract

This paper reports on the development of an analysis engine for the Research Writing Tutor (RWT), an AWE program designed to provide genre and discipline-specific feedback on the functional units of research article discourse. Unlike traditional NLP-based applications that categorize complete documents, the analyzer categorizes every sentence in Introduction section texts as both a communicative move and a rhetorical step. We describe the construction of a cascade of two support vector machine classifiers trained on a multi-disciplinary corpus of annotated texts. This work not only demonstrates the usefulness of NLP for automated genre analysis, but also paves the road for future AWE endeavors and forms of automated feedback that could facilitate effective expression of functional meaning in writing.

KEYWORDS: AUTOMATED WRITING EVALUATION; GENRE; MACHINE LEARNING; MOVES; TEXT CATEGORIZATION

Introduction

Second language acquisition research, in consensus with views in psychology, has long argued that individuals learn languages differently depending on such influencing factors as idiosyncratic learning strategies, cognitive styles, and various affective factors (Dörnyei & Skehan, 2003). Since it is not in the power of a human teacher to adapt to all learner differences and to provide individualized instruction to groups of students, computers have been

Affiliation

Iowa State University, Ames, IA 50011-12-01, United States
email: ecotos@iastate.edu (corresponding author)

CALICO JOURNAL VOL 33.1 2016 92–116
doi : 10.1558/cj.v33i1.27047

proposed as a powerful and practical complementary alternative. In the past decades, Natural Language Processing (NLP), which is superior to the so-called pattern-markup and error-anticipation techniques used to generate conventional types of feedback (Garret, 1987), has been commonly employed for the purpose of identifying problematic aspects in learner language, diagnosing language errors, and providing detailed explanations about the nature of those errors. In particular, NLP techniques have been exploited for individual feedback generation by Intelligent Computer Assisted Language Learning (ICALL) systems, generally addressing the complexity of morphology, syntax, semantics, and pragmatics (see Gamper & Knapp, 2002). The domain of Automated Writing Evaluation (AWE) has also employed NLP in a variety of ways to provide learners with feedback on grammar, usage, mechanics, style, organization, coherence, content, etc. (see Dikli, 2006).

Despite considerable advances in NLP, analyzing the various aspects of natural language is still a challenging problem that remains to be solved in order to meet a wider range of learning needs. One such need is mastering the writing conventions of academic genres. Research in English for academic purposes has a well-established genre-analysis agenda (see Biber, Connor, & Upton, 2007; Hyland, 2000). This agenda needs to be extrapolated to interdisciplinary research involving applied and computational linguists and computer scientists, whose combined efforts would result in the creation of new needs-based intelligent feedback systems.

The study presented in this paper is beginning to fill this void. Our main goal was to develop an automated analysis engine for an AWE system that identifies the rhetorical structure of research articles (RA) in terms of communicative *moves* and functional *steps* (Swales, 1990) and provides feedback compared to the RA genre norms in learners' particular disciplines. Before describing this work, we review the approaches to and implementations of automated discourse categorization in order to provide a background for our automated genre analysis methodology. Focusing on RA Introduction sections, we employed a three-step process of discourse structure identification, which included: (1) feature selection from manually annotated text data; (2) sentence representation; and (3) training leading to sentence-level classification into moves and steps.

Automated text categorization

Automated text categorization in machine learning

Text categorization, also known as text classification and sometimes referred to as topic spotting, is a procedure by which natural language texts are labeled with thematic categories from an existing predefined set. Although text categorization work emerged in the early 1960s (Maron, 1961), it became prominent only in the 1990s due to the availability of digital text documents and the growing need to obtain easy access as well as selective information about them. A popular approach to text categorization is machine learning (ML). Some applications of ML include automatic document indexing for information retrieval systems (e.g., internet search engines), document organization (e.g., grouping of conference papers into sessions), text filtering (e.g., junk e-mail blocking), and word sense disambiguation (e.g., 'fan' as a person or as an air blowing object).

Sebastiani (2002) provides a comprehensive overview of the ML approach where 'a general inductive process builds an automatic text classifier by learning, from a set of preclassified documents, the characteristics of the categories of interest' (p. 2). More specifically, this inductive process, also referred to as the *learner*, automatically constructs a classifier by learning the characteristics of a corpus of human-labeled texts and then looking for the characteristics that new texts should have in order to be classified similarly to human coding. In this supervised learning process, thus, the corpus prepared by human experts is instrumental, for it is used not only for the purpose of training the classifier, but also for its testing and validation. Research has shown that classifiers developed with ML techniques are highly effective. Of a multitude of such techniques, (e.g., Naïve Bayes, Decision Tree, Rule-based, Neural Network, Regression, etc.), the Support Vector Machine (SVM) classifiers have been widely implemented (Cortes & Vapnik, 1995) and found to 'deliver top-notch performance' (Sebastiani, 2002: 39).

Automated categorization of discourse and genre

In text classification tasks, texts have traditionally been considered in terms of their structure and content. The most well recognized perspective to analyzing discourse is the rhetorical structure theory (Mann & Thompson, 1988), which has been applied to numerous computational applications (see Taboada & Mann, 2006). Generally, discourse has been analyzed based on discourse markers, which are viewed as indicators of rhetorical relations in the text (e.g., Schilder, 2002). Classifying texts based on their genre-specific functional roles is a territory yet to be explored. According to Kessler, Numberg, and Schutze (1997), one of the reasons why this particular task has been somewhat daunting is that it poses high-order theoretical and methodological questions:

> Is genre a single property or attribute that can be neatly laid out in some hierarchical structure? Or are we really talking about a multidimensional space of properties that have little more in common than that they are more or less orthogonal to topicality? And once we have the theoretical prerequisites in place, we have to ask whether genre can be reliably identified by means of computationally tractable cues. (Kessler *et al.*, 1997: 1)

The challenge is even greater considering Miller's (1994) view of genre as 'a rhetorical means for mediating private intention and social exigence; […] connecting […] the singular with the recurrent' (Miller, 1984: 163),[1] which highlights the discourse community dimension of genre extensively analyzed in Rhetoric and Composition/Writing Studies (e.g., Bazerman, Bonini, & Figueiredo, 2009).

Thus, it is important to consider definitions of genre that reflect both the socio-linguistic meaning of the concept and also a meaning that can be interpreted for applied ML purposes. The definition of genre proposed by Kessler *et al.* (1997) – 'any widely recognized class of texts defined by some common communicative purpose or other functional traits, provided the function is connected to some formal cues or commonalities and that the class is extensible' (p. 2) – embodies concepts that allow for such an interface. Key to this definition is the idea of formal generic cues, which are surface attributes that both distinguish classes of texts and possess 'a characteristic set of computable structural or linguistic properties, whether categorical or statistical' (Kessler *et al.*, 1997: 2). Toms and Campbell (1999) also refer to genres as possessing 'a parsimonious set of attributes' that 'determine a document's ability to be identified uniquely' (p. 1). Similarly, Stamatatos, Fakotakis, and Kokkinakis (2000) regard genre detection as the identification of the functional styles of texts, maintaining that the style markers are a set of pre-defined quantifiable measures (p. 472).

These definitions resonate with the perspective of applied linguists, who conceptualize genres as communicative events organized into a series of discourse units, *moves* and *steps* (Swales, 1990). The moves are communicative goals, and the steps are rhetorical functions that help achieve the goals of given moves. The rhetorical intent of the steps, in particular, is rendered through functional language, or linguistic cues that are indicative of specific genre elements. The assumption that lexical cues are often explicit realizations of rhetorical organization is adopted both in applied linguistics work that uses quantitative and qualitative methods to describe genres (see Biber *et al.*, 2007; Cortes, 2013; Upton & Connor, 2001) and in ML, where lexical cues are primarily applied to detect discourse structure (Kurohashi & Nagao, 1994).

With the notion of cues as observable properties of texts, Kessler *et al.* (1997) conducted a series of experiments with structural cues (e.g., passives, nominalizations, topicalized sentences), lexical cues (e.g., terms of address, Latinate affixes, time-related vocabulary), character-level cues (punctuation, capitalized and hyphenated words), and derivative cues (ratios and variation measures derived from lexical and character-level cues). They built different models including linear discrimination, linear regressions, logistic regression, and neural networks, and concluded that using cues, especially lexical cues, for genre categorization is a plausible approach leading to reasonable accuracy of classification. In turn, Stamatatos *et al.* (2000) proposed an approach to distinguishing between genres that is based on first extracting lexical style markers using NLP and multiple regression, and then conducting discriminant analysis for automatic categorization. Cues also proved to be a viable solution to the classification of discourse relations among different parts of texts (Litman, 1994).

Automated discourse and genre categorization in computer-assisted writing tools

A few works at the intersection of ML and applied linguistics made a step further, applying automated discourse categorization techniques to the development of intelligent writing tools. Current genre-based writing applications rely mostly on lexical approaches. For example, Yang and Akahori (1998) built a system to help learners write technical texts in Japanese, which could automatically detect micro-level and macro-level cues. Their system contained a set of simple pattern matching rules and three analyzers (morpheme, syntax, and discourse). The feedback it generated displayed sentences with cohesive expressions containing cue words, which corresponded to a chosen headline in a text that the learners clicked on. Both headline and cohesive expression extraction achieved high accuracy (99.2% and 92.7%, respectively).

Cue-phrased based discourse parsing is also at the core of *Criterion*® (Marcu, 2000), a complex learning platform developed by the Educational Testing Service. *Criterion*®'s essay discourse analyzer, *e-rater*®, uses a voting algorithm that makes decisions based on three classifiers – decision-based, probabilistic-based, and probabilistic-local, which together perform with 0.85 precision and 0.85 recall (Burstein, Tetreault, & Madnani, 2013). Trained using simple maximum-likelihood techniques and expectation maximization, it identifies words, terms, and structures that act as discourse markers as well as language characteristic of essay discourse. The feedback is based on the classification of each sentence as one of the following categories: title, introductory material, thesis, main idea, supporting idea, or conclusion.

Motivated by instructional needs, Anthony and Lashkia (2003) developed the genre-based *Mover* to be used by English language learners for academic reading and writing purposes. The name of their software reflects Swales' move terminology, and the output it generates presents the learners with the move structure of RA Abstracts. To develop the *Mover*, the authors conceptualized their approach based on Mitchell's (1997) task-experience-performance sequence, where the task was to automatically identify the rhetorical structure of Abstracts, the experience was a supervised learning approach, and the performance was the evaluation of accuracy. After experimenting with various algorithms, including Decision Tree and Neural Network, the Naïve Bayes classifier was chosen as the supervised learning approach; it performed better than others yielding an average first-order accuracy of 68%, which the authors claimed could be improved to over 86%.

Previously, we approached a very similar genre classification task when we developed a feedback application called *IADE* (Pendar & Cotos, 2008). Both the *Mover* and *IADE* used a classification schema based on Swales' (1990) Create a Research Space (CARS) framework for Introduction sections. Our text-categorization approach was similar to that of Anthony and Lashkia (2003) in that it was a lexical (*n*-gram) approach that involved the use of human-labeled texts. The choice of the supervised learning technique was different – we used an SVM classifier, which performed best with unigram and trigram models. Based on the output of the classifier, *IADE* operationalized feedback generation in two ways: as color-coded (representing the move of each sentence with a respective color) and as numerical (showing percentages that reflected a comparison of the distribution of the moves in learners' drafts versus a corpus in their discipline) (Cotos, 2011).

The genre-based systems described above have all been implemented in learning contexts. The amount of empirical evidence suggesting their effectiveness and potential benefits for language learners is accumulating (Attali, 2004; Chen & Cheng, 2008; Cotos, 2014) and is, therefore, motivating the development of new intelligent writing applications. An example of such applications is the *Research Writing Tutor (RWT)*, a scale-up from its *IADE* prototype which generates discourse-level feedback on all RA sections (Cotos, 2015). In this paper, we present only part of a bigger development project and focus on the ML approach employed to build the Introductions analysis and feedback engine of *RWT*.

Automated move and step identification

RWT is intended for use as a computer-assisted aid to academic writing instruction that focuses on the genre conventions of RAs, so our task was to build an analysis engine capable of classifying texts into moves and steps. As shown in Figure 1, we approached the identification of these discourse units as a supervised classification problem and employed a process of corpus data annotation, feature selection, sentence representation, and training leading to classification. Following this approach, we considered each sentence in a text as an independent unit of analysis to be classified into a move category and then into a step within the identified move.

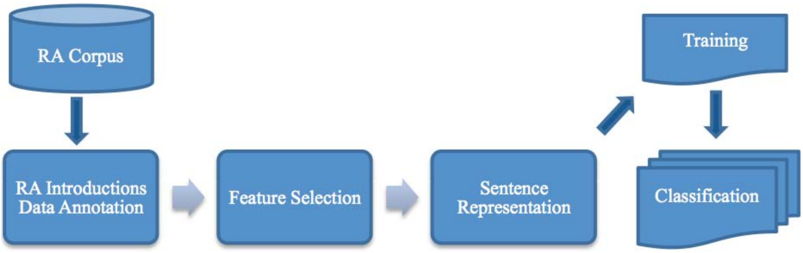


Figure 1: Move and step classification process.

RA Introduction training data

As any supervised classification, the move/step classification task required text data. We used the Introduction sections from a large specialized corpus of 1,020 research articles. The articles were obtained from reputable online academic journals in 51 disciplines, each discipline being represented by 20 articles.

The Introduction texts contained 1,322,089 words. They were converted to .TXT files and were manually annotated using a scheme of three moves and 17 steps (Table 1). Each sentence was tagged with a move and a step; multi-functional stretches of text were tagged as several steps, which could belong to the same move or to different moves. The annotation was completed using an XML-based markup created by the Callisto workbench, which allowed for nesting step XML tags inside move XML tags as well as for assigning multiple tags to sentences in order to capture multi-functionality at sentence-level (Appendix A).

Table 1: Move/step schema used for corpus annotation (based on Swales' CARS model)

Move	Step
Move 1. Establishing a territory	Step 1. Claiming centrality Step 2. Making topic generalizations Step 3. Reviewing previous research
Move 2. Identifying a niche	Step 4. Indicating a gap Step 5. Highlighting a problem Step 6. Raising general questions Step 7. Proposing general hypotheses Step 8. Presenting a justification
Move 3. Addressing the niche	Step 9. Introducing present research descriptively Step 10. Introducing present research purposefully Step 11. Presenting research questions Step 12. Presenting research hypotheses Step 13. Clarifying definitions Step 14. Summarizing methods Step 15. Announcing principal outcomes Step 16. Stating the value of the present research Step 17. Outlining the structure of the paper

Three experienced coders executed the annotation task.[2] Text annotation was accompanied by calculations of agreement on moves and steps in 30 texts from different disciplines; 18 texts were random and 12 were purposefully chosen, as they contained various instances of discourse whose rhetorical functions were likely to elicit different interpretations and were thus valuable material for 14 calibration meetings conducted weekly during three and a half months of annotation. This procedure helped to develop a comprehensive coding protocol to ensure reliability and consistency of annotation (as per Connor, Upton, & Kanoksilapatham, 2007) and to foster the adjudication of individual cases of disagreement. The intraclass correlation coefficient (ICC) (see Shrout & Fleiss, 1979) estimates are indicative of relatively high agreement among the three coders both for moves ($r= 0.86$, $p < 0.005$) and for steps ($r= 0.80$, $p < 0.005$) (Saricaoglu & Cotos, 2013).

For training we used a sub-corpus of 650 Introductions, which were extracted from the annotated corpus by means of stratified sampling (Appendix B). This sub-corpus contained a total of 15,460 sentences and 366,089 words.[3] The size of the sub-corpus both in terms of the number of texts (average 13) and of the number of words per discipline (average 7,000) can be considered adequate as per Stamatatos *et al.* (2000).[4]

Feature selection

Feature selection is an important step in a classification task and involves identifying the best features, or linguistic cues from the dataset that help reliably represent the data with respect to the target classes,[5] moves and steps in this study. The main features used for the identification of moves and steps were sets of word unigrams and trigrams (i.e., single words and three word sequences) from the annotated corpus. In our earlier work, we found that bigrams (two word sequences) had a negative effect on the classifier (Pendar & Cotos, 2008), which is why we did not experiment with the bigrams here. We also found that our extracted features were not discipline-dependent. To prepare the feature set for the classification task, the unigram and trigram data were preprocessed as follows:

1. The unigram and bigram tokens were stemmed to reduce the size of the feature set as well as the interdependence among features by representing lexically related items as the same, unified feature. Stemming was completed using NLTK3 port of the Porter Stemmer algorithm (Porter, 1980).
2. Four digit numbers denoting years in citations were replaced with __year__, and all other numerical tokens were normalized to the token __number__. The numbers were preprocessed in this way

because, even though they are not lexical realizations of rhetorical intent, they are indicative of the *Reviewing previous research* step of Move 1.

3. Three other types of substitutions based on the recurring patterns in the corpus data were made:
   a. HTML special characters (e.g., &quot, &amp) were replaced with __html__
   b. web-page links were replaced with __url__
   c. domain names were replaced with __domain__
4. *n*-grams with a frequency of less than 5 were excluded to avoid overfitting and to reduce the so-called noise, which could be created by the n-grams that would not contribute much to the learning process.

Next, to identify the features that are most indicative of a given move and step, odds ratios[6] of the *n*-grams were calculated against each move and step using the following formulas:

$$OR(f_i, m_j) = \frac{p(f_i|m_j).\left(1 - p(f_i|\overline{m}_j)\right)}{(1 - p(f_i|m_j)).p(f_i|\overline{m}_j)}$$

where $OR(f_i, m_j)$ is the odds ratio of feature f_i (or *n*-gram$_i$) occurring in move m_j; $p(f_i|m_j)$ is the probability of occurrence of feature f_i given move m_j; and $p(f_i|\overline{m}_j)$ is the probability of occurrence of feature f_i given a move that is not m_j.

$$OR(f_i, s_j) = \frac{p(f_i|s_j).\left(1 - p(f_i|\bar{s}_j)\right)}{(1 - p(f_i|s_j)).p(f_i|\bar{s}_j)}$$

Similarly, $OR(f_i, s_j)$ is the odds ratio of feature f_i occurring in step s_j; $p(f_i|s_j)$ is the probability of occurrence of feature f_i given step s_j; and $p(f_i|\bar{s}_j)$ is the probability of occurrence of feature f_i given a step other than s_j. The conditional probabilities in these formulas were calculated as maximum likelihood estimates $p(f_j|m_j)$ and $p(f_j|s_j)$, respectively, where N is the total number of *n*-grams in the corpus of sentences C:

$$p(f_i|m_j) = \frac{count(f_i \mid m_j)}{\sum_{k=1}^{N} count(f_k \mid m_j)} \quad \text{and} \quad p(f_i|s_j) = \frac{count(f_i \mid s_j)}{\sum_{k=1}^{N} count(f_k \mid s_j)}$$

The features, or *n*-grams exhibiting high odds ratios were selected as features indicative of a given move and step; the features with odds ratios less than 5 were removed. The final *n*-gram feature set contained 5,825 unigrams and 11,630 trigrams for moves, and 27,689 unigrams and 27,160 trigrams for steps.

Sentence representation

We considered each sentence as an item to be classified into a move and a step; hence, it is represented as an *n*-dimensional vector in the R^n Euclidean space. Formally, each sentence c_i is represented as $c_i = <f_1, f_2, f_3,\ldots,f_n>$ where each f_j measures feature *j* in sentence c_i. Thus, the learning algorithm attempts to learn a functional mapping that maps each sentence in the corpus *C* to a move *m*, and then using this move *m* to map each sentence to a step *s*. Here $M = \{m_1, m_{2,} m_3\}$ and $S = \{s_1, s_2, s_{3,}..,s_{17}\}$. Mathematically, the learning algorithm tries to predict functions *F* and *G* such that

$$\{F: C \rightarrow M\} \text{ and } \{G: C, M \rightarrow S\}.$$

In other words, function *F* would map the sentences in the corpus to one of the three move classes in *M*, and function *G* would map those sentences to one of the 17 step classes in *S*. Although it would be ideal to accomplish many-to-many mappings (which would be similar to the coders' multi-level annotation of the corpus), at this point, both for simplicity and practicality, we assumed both *F* and *G* functions as many-to-one mappings.

Given that our units of analysis were individual sentences, which are very small documents and therefore inappropriate to use measures of the importance of a term in a document,[7] we resorted to Boolean representation in order to indicate the presence or absence of a particular feature. In other words, we used binary coding such that if an *n*-gram feature *j* is present in sentence c_i, f_j equals 1; if an *n*-gram feature *j* is absent in sentence c_i, f_j equals 0. For example, for move classification the representation of a sentence may be:

$$c_i = < mf_1{:}1,\ mf_2{:}1,\ mf_3{:}0,\ldots,mf_n{:}0>$$

where c_i is a sentence from the annotated sub-corpus *C*, and $mf_1, mf_2, mf_3,\ldots, mf_n$ are the features representing the move to which a sentence belongs. Sentence representation for step classification is similar, but includes an additional feature that specifies the step to which the sentence belongs:

$$c_i = < m_1{:}1, m_2{:}0,\ m_3{:}0\ ,\ sf_1{:}1,\ sf_2{:}1,\ sf_3{:}0,\ldots,\ sf_n{:}0>$$

where c_i is a sentence from the annotated sub-corpus *C*, and $sf_1, sf_2, sf_3,\ldots,sf_n$ are the features representing the step of the sentence. Also $m_1{:}1$, $m_2{:}0$, $m_3{:}0$ in the example above implies that sentence c_i belongs to move 1 and not to move 2 or move 3. Thus, in this representation, the move predicted for a sentence is passed as an input to predict a step. Figure 2 provides an example of how a new sentence is processed and represented for move classification. First, the sentence is divided into unigrams and trigrams and matched with the existing feature set. Then, the unigrams and trigrams are represented as Boolean values: 1 if the *n*-gram was found in the feature set and 0 if it was not found. Based on the features represented as 1, the classifier makes a decision as to which move the sentence belongs to.

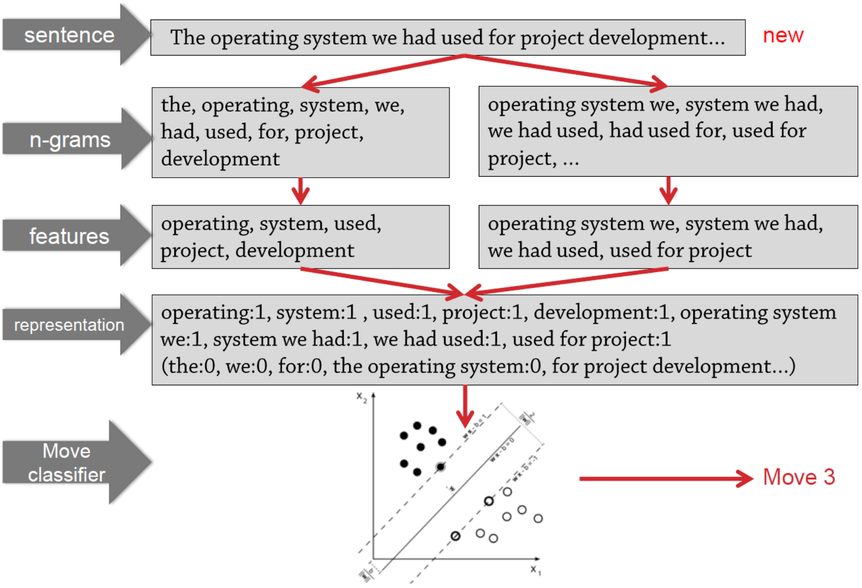

sentence
The operating system we had used for project development...
new
n-grams
the, operating, system, we, had, used, for, project, development
operating system we, system we had, we had used, had used for, used for project, ...
features
operating, system, used, project, development
operating system we, system we had, we had used, used for project
representation
operating:1, system:1 , used:1, project:1, development:1, operating system we:1, system we had:1, we had used:1, used for project:1
(the:0, we:0, for:0, the operating system:0, for project development...)
Move classifier
Move 3

Figure 2: Example of sentence classification as a move

Move and step classifiers

SVM learning has been traditionally exploited in text categorization problems. It is a supervised learning technique that uses an algorithm to analyze data and identify patterns, which are then used for classification. Provided with an input of a set of labeled training data, the SVM model represents the training examples as points in an *N*-dimensional space that are mapped such that the labeled classes are optimally separated by hyperplanes of maximal margin, or clear gaps. Once the SVM learns the hyperplanes, it can classify unseen data into one of the learned labeled classes. Fed with a new example, the model maps it into the same space and makes a prediction as to which class it belongs to based on which side of the hyperplane it is on.

In our case, the labeled training data set was the annotated Introductions corpus, the training examples were the annotated sentences, the classes were the move and step categories, and the hyperplanes separated the move or the step classes. Figure 3 depicts this SVM learning trajectory for move identification.[8] We chose SVM not only because it generally yields better performance, but also because it performs well in a high dimensional space even with sparse values (Kivinen, Warmuth, & Auer, 1997), that is, when most of the values in a large vector are zero. This type of sparse representation is common in natural language analysis because in any given excerpts of text (sentences here) only a handful of items from the feature set are observed.

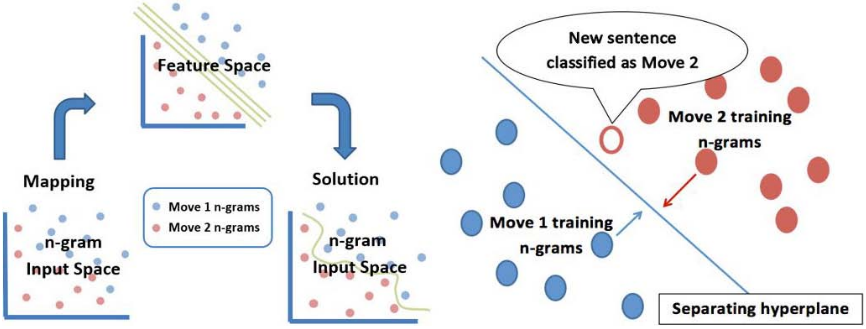


Figure 3: SVM move learning and classification trajectory

Further, the accuracy of classification depends on careful selection of parameters that are fed to the SVM model. Considering our task of building a predictive classifier, we employed a common technique known as *k*-fold cross-validation in order to estimate how well the model would perform when given completely new data. This procedure involved the application of the remaining 370 annotated Introduction texts not used for model training, which were randomly partitioned into 10 equal size subsets and used for 10-fold cross validation, a common technique for this type of evaluations (McLachlan, Do, & Ambroise, 2004). Specifically, we fed the learned model with one of the 10 subsets of unseen labeled data at a time and compared the move and step classes it generated with the move and step labels assigned by the coders.

We experimented with different feature sets for both move and step classification tasks (Tables 2 and 3). For evaluating the performance of the classifier on these feature sets, we used measures of accuracy on each of the models built. Accuracy measures the proportion of correctly classified instances to the total number of classified instances. Other standard metrics used for evaluating the model performance are precision and recall. Precision measures the proportion of items assigned to a category that actually belong to that category, whereas recall measures the proportion of items belonging to a category that were classified correctly. In the formulas below, *TP* is the number of true positives; *FP* is the number of false positives, *TN* is the number of true negatives, and *FN* is the number of false negatives, and all these terms indicate a comparison of the results of the classifier with expert judgments. True and false indicate whether the classifier's prediction corresponds to the expert judgment (in our case the coders' move/step label), while positive and negative refers to the expected prediction by the classifier.

1. Accuracy: $$a = \frac{TP+TN}{TP+FP+TN+FN}$$
2. Precision: $$p = \frac{TP}{TP+FP}$$
3. Recall: $$r = \frac{TP}{TP+FN}$$

Table 2: Feature set for move classification

N-gram features	
# Unigrams	*# Trigrams*
1,000	0
2,000	0
3,000	0
0	1,000
0	2,000
0	3,000
1,000	1,000
2,000	2,000
3,000	3,000
5,825	11,630

Table 3: Feature set for step classification

N-gram features	
# Unigrams	*# Trigrams*
1,000	0
5,000	0
6,334	0
10,000	0
26,789	0
0	1,000
0	5,000
0	5,986
0	10,000
1,000	1,000
5,000	5,000
10,000	10,000
27,689	27,160

Figures 4 and 5 report the performance of the classifier on cross-validation data, showing that the accuracy of the move classifier increases as the feature set increases in size. Also, accuracy is slightly higher when the feature set contains both unigrams and trigrams than when unigrams or trigrams are used separately. The accuracy of the step classifier exhibits a comparable trend.

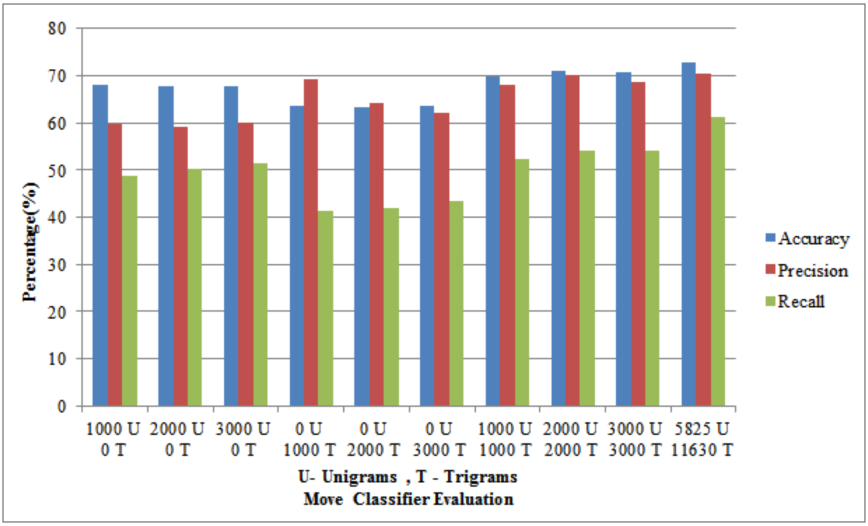


Figure 4: SVM performance on move classification

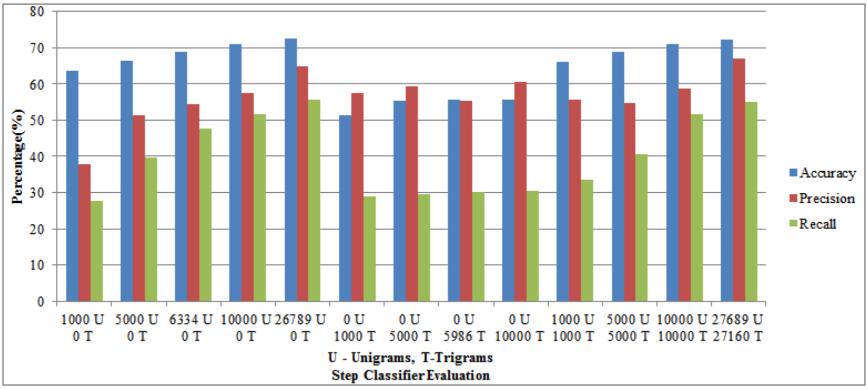


Figure 5: SVM performance on step classification

Similarly, in terms of precision and recall, the feature sets containing only unigrams or trigrams have lower precision and recall for both move and step classifiers. The move and step classifier models show an increase in precision and recall as more unigrams and trigrams are added into the feature set. A high precision and recall for both move and step classifiers is evident with the feature set containing most unigrams and trigrams taken together. It is also noticeable that the combined unigram and trigram feature sets yield precision figures that are higher than recall – 70.3% versus 61.2% for the move classifier and 68.6% versus 55% for the step classifier. This may be preferable when it comes to classification for error feedback generation. The developers of *Criterion*® opted for maximizing precision even if it was at the expense of recall; for example, precision for article and preposition error detection is 90% and 80% while recall is 40% and 25%, respectively (Chodorow, Gamon, & Tetreault, 2010). Nagata and Nakatani (2010) also hypothesized that feedback based on precision-oriented error-detection is likely to have a stronger learning effect than the recall-oriented feedback. For us however, tuning for precision is not advisable. Since we are classifying every single sentence, high precision in one category necessarily leads to low precision in another category. Therefore, our ultimate objective is to maximize accuracy.

Having found which model performed best, we built a cascade of two SVM classifiers. When a new input sentence is passed, it goes through the move classifier, which predicts its move, and then it is passed on to the step classifier, which predicts its step within the assigned move (Figure 6).

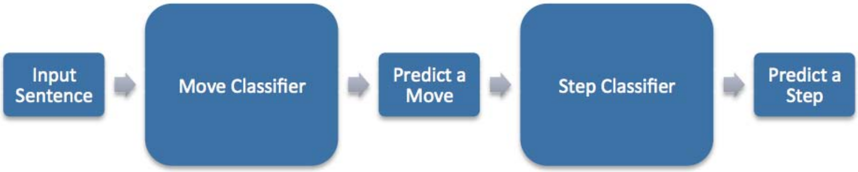


Figure 6: Sentence classification process by *RWT* analyzer

Evaluation and discussion

It is important to consider the classifiers' performance at the level of individual moves and steps. Like other systems, both our classifiers can predict some discourse elements better than others. In the following discussion, we compare the precision, recall and F1 scores obtained for each move/step. The F1 score, or the harmonic mean of precision and recall, measures the overall performance of the system for a category (Van Rijsbergen, 1979) and is calculated as:

$$F1 = 2\frac{PR}{P+R}$$

Table 4 shows that the move classifier predicted Move 1 and Move 3 with higher precision than Move 2. This result is in agreement with our earlier experimentation where we found that Move 2 is most difficult to identify and that it tends to be misclassified as Move 1 (Pendar & Cotos, 2008). This is not surprising since this time the training data for Move 2 was also considerably sparser than the data for the other two moves (6,039 sentences for Move 1; 1,609 for Move 2; and 2,352 for Move 3). In our testing dataset, the moves were not equally distributed either, with Move 2 being least represented (3,233 sentences for Move 1; 926 for Move 2; and 1,301 for Move 3). It is worth noting that the system obtained the best recall on Move 1, which combined with relatively high precision on that category results in the highest F1 score. While this may be attributed to the larger amount of features in the dataset, this move may also contain less ambiguous and/or more overt linguistic cues.

Table 4: Precision and recall for the move classifier

Move #	Move name	Precision (%)	Recall (%)	F1 Score (%)
1	Establishing a territory	73.3	**89.0**	**80.4**
2	Identifying a niche	59.2	37.3	45.8
3	Addressing the niche	**78.4**	57.2	66.1
Average		70.3	61.2	65.4

Table 5 shows that 10 out of 17 steps were predicted quite well by the step classifier. A few steps, in particular, had very high precision: *Clarifying definitions* – 100%, *Outlining the structure of the paper* – 92%, *Reviewing previous research* – 86.7%, and *Presenting research questions* – 84.6%. Table 5 also lists the steps that had a precision below the 68% average, three of which belong to Move 2 (*Highlighting a problem*, *Raising general questions*, *Proposing general hypotheses*) and four to Move 3 (*Introducing present research descriptively*, *Summarizing methods*, *Announcing principal outcomes*, and *Stating the value of the present research*). The steps of Move 1 were identified relatively well, as were many of the Move 3 steps, especially considering that Move 3 has the highest number of steps. The steps of Move 2, on the other hand, appear to be more problematic for classification – just like Move 2 itself. Overall performance is best on Step 3, *Reviewing previous research*, Step 5, *Highlighting a problem*, and Step 17, *Outlining the structure of the paper*, suggesting that these categories are signaled by relatively unambiguous lexical cues. The system, however, appears to struggle with Step 6, *Raising general questions,* Step 13, *Clarifying definitions* (despite high precision on this category), and Step 16, *Stating the value of the present research.*

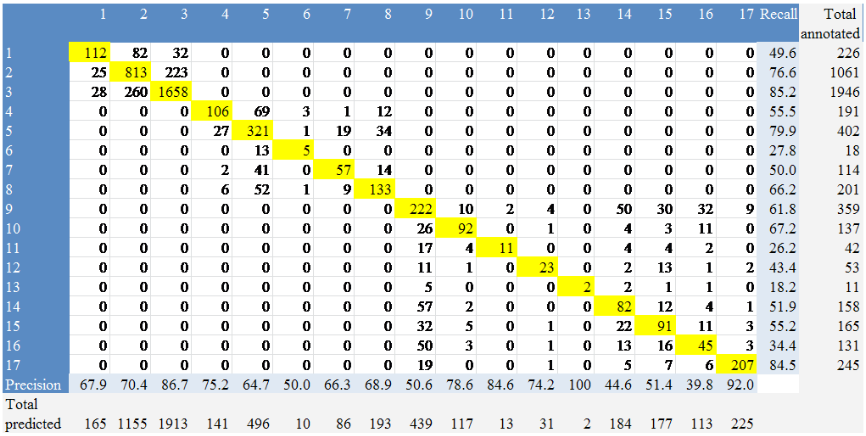
Table 5: Precision and recall for the step classifier

Step #	Step name	Precision (%)	Recall (%)	F1 Score (%)
1 (Move1)	Claiming centrality	67.9	49.6	57.3
2 (Move1)	Making topic generalizations	70.4	76.6	73.4
3 (Move1)	Reviewing previous research	**86.7**	**85.2**	**85.9**
4 (Move2)	Indicating a gap	**75.2**	55.5	63.9
5 (Move2)	Highlighting a problem	64.7	**79.9**	**71.5**

6 (Move2)	Raising general questions	50.0	27.8	35.7
7 (Move2)	Proposing general hypotheses	66.3	50.0	57.0
8 (Move2)	Presenting a justification	68.9	66.2	67.5
9 (Move3)	Introducing present research descriptively	50.6	61.9	55.7
10 (Move3)	Introducing present research purposefully	78.6	67.2	72.5
11 (Move3)	Presenting research questions	**84.6**	26.2	40.0
12 (Move3)	Presenting research hypotheses	74.2	43.4	54.8
13 (Move3)	Clarifying definitions	**100.0**	18.2	30.8
14 (Move3)	Summarizing methods	44.6	51.9	48.0
15 (Move3)	Announcing principal outcomes	51.4	55.2	53.2
16 (Move3)	Stating the value of the present research	39.8	34.4	36.9
17 (Move3)	Outlining the structure of the paper	**92.0**	**84.5**	**88.1**
Average		68.6	54.9	61.0

Our performance evaluation measures are slightly lower than *Criterion*'s overall precision of classification into discourse elements by best single system (81%) and by the voting system (85%) (Burstein, Marcu, & Knight, 2003). However, this is not at all discouraging given the increased complexity of our categorization task. Compared with Anthony and Lashkia (2003), our SVM model performs better when identifying *Claiming centrality* and *Highlighting a gap*. Their Naïve Bayes model classified statements of announcing research with higher accuracy than our step SVM; however, in *Mover* this category combined five steps that our classifier identifies separately (*Introducing present research purposefully*, *Introducing present research descriptively*, *Presenting research questions*, *Presenting research hypotheses*, and *Summarizing methods*). Principal outcomes and value statements are problematic for both *Mover* and *RWT*.

To better understand why and how misclassification occurs, we computed a confusion matrix comparing the categories predicted by the step classifier with the coders' annotation using the training dataset (Figure 7). The columns in the matrix represent the steps predicted by the step classifier, and the rows represent the primary step labels assigned by the coders. The highlighted diagonal line shows the number of correct predictions, and the off-diagonal counts represent the classifications that are different from human annotation. The calculations are based on the final SVM model of 27,689 unigrams 27,160 trigrams. The matrix reveals that the steps with the precision below the 68.6% average were confused with other steps. Additionally, it indicates that when misclassifications occurred, the misclassified step was still in the realm of the correct move. The classifier had lower performance when distinguishing between the steps of Move 1, in particular getting confused about Step 1 (*Claiming centrality*) and Step 2 (*Making topic generalizations*). In Move 2, it tended to classify sentences as Step 8 (*Presenting a justification*) instead of Step 5 (*Highlighting a problem*), and Step 5 instead of Step 6 (*Raising general questions*), Step 7 (*Proposing general hypotheses*) and Step 8. In Move 3, Step 9 (*Introducing present research descriptively*) appears to be most challenging – it was misclassified as Steps 14 (*Summarizing methods*), 15 (*Announcing principal outcomes*), and 16 (*Stating the value of present research*); Step 14 – as Steps 9; Step 15 – as Steps 9 and 14; and Step 16 – as Step 9.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	Recall	Total annotated
1	112	82	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	49.6	226
2	25	813	223	0	0	0	0	0	0	0	0	0	0	0	0	0	0	76.6	1061
3	28	260	1658	0	0	0	0	0	0	0	0	0	0	0	0	0	0	85.2	1946
4	0	0	0	106	69	3	1	12	0	0	0	0	0	0	0	0	0	55.5	191
5	0	0	0	27	321	1	19	34	0	0	0	0	0	0	0	0	0	79.9	402
6	0	0	0	0	13	5	0	0	0	0	0	0	0	0	0	0	0	27.8	18
7	0	0	0	2	41	0	57	14	0	0	0	0	0	0	0	0	0	50.0	114
8	0	0	0	6	52	1	9	133	0	0	0	0	0	0	0	0	0	66.2	201
9	0	0	0	0	0	0	0	0	222	10	2	4	0	50	30	32	9	61.8	359
10	0	0	0	0	0	0	0	0	26	92	0	1	0	4	3	11	0	67.2	137
11	0	0	0	0	0	0	0	0	17	4	11	0	0	4	4	2	0	26.2	42
12	0	0	0	0	0	0	0	0	11	1	0	23	0	2	13	1	2	43.4	53
13	0	0	0	0	0	0	0	0	5	0	0	0	2	2	1	1	0	18.2	11
14	0	0	0	0	0	0	0	0	57	2	0	0	0	82	12	4	1	51.9	158
15	0	0	0	0	0	0	0	0	32	5	0	1	0	22	91	11	3	55.2	165
16	0	0	0	0	0	0	0	0	50	3	0	1	0	13	16	45	3	34.4	131
17	0	0	0	0	0	0	0	0	19	0	0	1	0	5	7	6	207	84.5	245
Precision	67.9	70.4	86.7	75.2	64.7	50.0	66.3	68.9	50.6	78.6	84.6	74.2	100	44.6	51.4	39.8	92.0		
Total predicted	165	1155	1913	141	496	10	86	193	439	117	13	31	2	184	177	113	225		

Figure 7: Confusion matrix for steps predicted by the step classifier and annotated by coders

These misclassifications by the step classifier are not surprising. Sparseness of training data, a major reason often mentioned in previous research, accounts for the lower performance in our study as well. In addition, there are a number of other factors that can help explain our SVM performance results. For instance, some steps are more challenging for automated identification because their rhetorical meaning is not as clearly encoded in functional language and is, therefore, difficult to operationalize by a learning model. Another reason is that a sentence can carry multiple rhetorical functions and thus belong to more than one step. While the coders were able to capture this phenomenon when annotating the corpus, the classifiers were only capable of predicting one move and one step category. We will further qualitatively analyze the classifiers' output to see whether the misclassifications are indeed inaccurate or whether they are capturing secondary functions. An equally important factor is meaning ambiguity; in the absence of lexical signals of functional meaning the coders were often confused as well.

Conclusions and future work

In this study, we developed a cascade of two SVM move and step classifiers that are at the core of *RWT*'s Introduction discourse analyzer. For that, we combined work in genre analysis and ML, relying on linguistic cues indicative of rhetorical functions. Our evaluation results are in agreement with previous research on classification of discourse elements and, in some aspects, outperform existing automated classification systems (e.g., Anthony & Lashkia, 2003). The analyzer classifies new input sentences with an overall move accuracy of 72.6% and step accuracy of 72.9%, the latter being slightly higher likely due to the preceding move classification in the sentence classification sequence.

Up to this point, we have been treating each sentence as an independent random variable; that is, we were assuming that the move/step represented by each sentence is independent of its context. This is a useful, yet not a definitive assumption. It is useful in that it allows us to understand how much the linguistic information contained within a sentence contributes to its move/step classification. It seems that we are reaching the limits of this approach, and it is now prudent to investigate the influence of the context. In further work, we are planning to incorporate context information and the sequencing of moves/steps in our predictive models. Additionally, we are planning to implement a ranking of classification decisions based on higher probabilities to be able to distinguish between primary and secondary step functions. For steps that are most difficult to detect, we will take a knowledge-based approach (as in Madnani, Heilman, Tetreault, & Chodorow, 2012) and experiment with a set of hand-written rules to recognize the functional language and, perhaps, the lexico-grammatical patterns that are identifiable in the annotated corpus but not frequent enough to appear in our current set of *n*-gram features. With new results from these additional approaches, we may develop a voting algorithm that would pass final classification decisions considering the output of a number of independent analyzers, similar to Burstein *et al.* (2003). With this work, we not only demonstrate the usefulness of ML and NLP for automated genre analysis, but also pave the road for future endeavors that will lead to the development of AWE and ICALL systems with meaning-oriented feedback.

Acknowledgements

We thank all the members of the *RWT* research and development team, giving special acknowledgement to the work of our research assistants Deepan Prabhu Babu, Sarah Huffman, Stephanie Link, Todd Paben, and Aysel Saricaoglu. We also express gratitude to the guest editors of this special issue and the anonymous reviewers.

This material is based upon work supported by the Senior Vice President and Provost/Computation Advisory Committee, Graduate College, College of Engineering, and College of Liberal Arts and Sciences at Iowa State University.

Notes

1. Miller's (1994) definition emphasizes the importance of genre in providing insight about discourse communities, which is particularly relevant given our end-goal to develop an AWE tool for the analysis of disciplinary RA genre discourse.

2. The coders acquired the needed expertise through a focused four-week training that involved guided identification, analysis, and discussion of moves and steps in published Introductions.

4. Stamatatos *et al.* (2000) recommend at least 10 texts per category and an average text length no shorter than 1,000 words.

5. See Sebastiani (2002) for an overview of feature selection techniques in text categorization.

6. Based on literature reporting feature selection experiments in ML (e.g., Mladenic, 1998; Sebastiani, 2002), from different possible options – maximum values, information gain, and odds ratios – we chose the latter because it was found to result in the highest classification accuracy.

7. In text categorization, term frequency times the inverse document frequency (tf.idf) is used to measure the importance of a term in a document.

8. LIBSVM (http://www.csie.ntu.edu.tw/~cjlin/libsvm/) was used to construct the move and step classifiers.

About the authors

Elena Cotos is an Assistant Professor in the Department of English, Applied Linguistics and Technology Program, and the Director of the Center for Communication Excellence of the Graduate College at Iowa State University. Her research interests include computer-assisted language learning and assessment, corpus-based genre analysis, English for specific/academic purposes, and data-driven learning. She is the author of *Genre-based Automated Writing Evaluation for L2 Research Writing* published by Palgrave Macmillan. Her work has also appeared in a number of edited volumes, and in journals such as *CALICO Journal*, *ReCALL*, *Language Testing*, *International Journal of Computer-Assisted Language Learning and Teaching*, *Journal of English for Academic Purposes*, *Writing & Pedagogy* and *Language Learning and Technology*. E-mail address ecotos@iastate.edu.

Nick Pendar is Senior NLP/ML Scientist at Skytree – The Machine Learning Company®, San Francisco Bay Area. He holds a doctoral degree in Linguistics and Computer Science from the University of Toronto. His expertise lies in statistical natural language processing, symbolic computational linguistics, text categorization, sentiment analysis, information retrieval, text data mining, text analytics, and machine learning. E-mail address npendar@gmail.com.

References

Anthony, L., & Lashkia, G. (2003). Mover: A machine learning tool to assist in the reading and writing of technical papers. *IEEE Transactions on Professional Communication,* 46 (3), 185–193. http://dx.doi.org/10.1109/TPC.2003.816789

Attali, Y. (2004). Exploring the feedback and revision features of Criterion. *Journal of Second Language Writing*, 14, 191–205.

Bazerman, C., Bonini, A., & Figueiredo, D. (Eds). (2009). *Genre in a Changing World. Perspectives on Writing.* Fort Collins, Colorado: The WAC Clearinghouse and Parlor Press. Available at http://wac.colostate.edu/books/genre/

Biber, D. (1995). *Dimensions of register variation: A cross-linguistic comparison*. Cambridge: Cambridge University Press. http://dx.doi.org/10.1017/CBO9780511519871

Biber, D., Connor, U., & Upton, T. (2007). *Discourse on the Move: Using Corpus Analysis to Describe Discourse Structure*. Amsterdam: John Benjamins. http://dx.doi.org/10.1075/scl.28

Burstein, J. (2003). The e-rater text registered scoring engine: Automated essay scoring with natural language processing. In M. D. Shermis & J. Burstein (Eds), *Automated essay scoring: A cross-disciplinary perspective,* 113–121. Mahwah, NJ: Lawrence Erlbaum.

Burstein, J. (2009). Opportunities for natural language processing research in education. *Computational Linguistics and Intelligent Text Processing*, 6–27. http://dx.doi.org/10.1007/978-3-642-00382-0_2

Burstein, J., Marcu, D., & Knight, K. (2003). Finding the WRITE stuff: Automatic identification of discourse structure in student essays. *IEEE Intelligent Systems: Special Issue on Natural Language Processing,* 18 (1), 32–39. http://dx.doi.org/10.1109/MIS.2003.1179191

Burstein, J., Tetreault, J., & Madnani, N. (2013). The e-rater automated essay scoring system. In M. D. Shermis, & J. Burstein (Eds), *Handbook of automated essay scoring: Current applications and future directions,* 55–67. New York: Routledge.

Chen, C. F. E., & Cheng, W. Y. E. (2008). Beyond the design of automated writing evaluation: Pedagogical practices and perceived learning effectiveness in EFL writing classes. *Language Learning & Technology*, 12 (2), 94–112.

Chodorow, M., Gamon, M., & Tetreault, J. (2010). The utility of article and preposition error correction systems for English language learners: Feedback and assessment. *Language Testing*, 27 (3), 419–436. http://dx.doi.org/10.1177/0265532210364391

Connor, U., Upton, U., & Kanoksilapatham, B. (2007). Introduction to Move Analysis. In D. Biber & T. Ulla Upton (Eds) *Discourse on the Move: Using Corpus Analysis to Describe Discourse Structure,* 23–42. Amsterdam: John Benjamins.

Cotos, E. (2011). Potential of automated writing evaluation feedback. *CALICO Journal*, 28 (2), 420–459. http://dx.doi.org/10.11139/cj.28.2.420-459

Cotos, E. (2014). *Genre-based automated writing evaluation for L2 research writing: From design to evaluation and enhancement*. New York: Palgrave Macmillan. http://dx.doi.org/10.1057/9781137333377

Cotos, E. (2015). AWE for writing pedagogy: From healthy tension to tangible prospects. *Writing and Pedagogy*, 7 (2-3), 197–231.

Cortes, V. (2013). 'The purpose of this study is to': Connecting lexical bundles and moves in research article introductions. *Journal of English for Academic Purposes,* 12 (1), 33–43. http://dx.doi.org/10.1016/j.jeap.2012.11.002

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning,* 20 (3), 273–297. http://dx.doi.org/10.1007/BF00994018

Dikli, S. (2006). An overview of automated scoring of essays. *Journal of Technology, Learning, and Assessment,* 5 (1). Retrieved 18 August 2007 from http://www.jtla.org.

Dörnyei, Z., & Skehan, P. (2003). Individual differences in second language learning. In C. J. Doughty & M. H. Long (Eds), *The handbook of second language acquisition,* 589–630. Malden, MA and Oxford: Blackwell. http://dx.doi.org/10.1002/9780470756492.ch18

Gamper, J., & Knapp, J. (2002). A review of intelligent CALL systems. *Computer Assisted Language Learning*, 15 (4), 329–342. http://dx.doi.org/10.1076/call.15.4.329.8270

Garrett, N. (1987). A Psycholinguistic Perspective on Grammar and CALL. In Wm. Flint Smith (Ed.) *Modern media in foreign language education: Theory and implementation,* 169–196. Lincolnwood, IL: National Textbook.

Gliner, J. A., & Morgan, G. A. (2000). *Research methods in applied settings: An integrated approach to design and analysis*. Mahwah, NJ Lawrence Erlbaum.

Hyland, K. (2000). *Disciplinary discourses*. London: Longman.

Kessler, B., Numberg, G., & Schütze, H. (1997). Automatic detection of text genre. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics,* 32–38. Association for Computational Linguistics.

Kivinen, J., Warmuth, M., & Auer P. (1997). The Perceptron algorithm vs. Winnow: Linear vs. logarithmic mistake bound when few input variables are relevant. *Artificial Intelligence*, 1–2, 325–343. http://dx.doi.org/10.1016/S0004-3702(97)00039-8

Kurohashi, S., & Nagao, M. (1994). Automatic detection of discourse structure by checking surface information in sentences. In *Proceedings of the 15th International Conference on Computational Linguistics*, *Vol. 2,* 1123–1127. http://dx.doi.org/10.3115/991250.991334

Litman, D. J. (1994). Classifying cue phrases in text and speech using machine learning. *arXiv preprint cmp-lg/9405014*.

Mann, W. C., & Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8 (3), 243–281. http://dx.doi.org/10.1515/text.1.1988.8.3.243

Madnani, N., Heilman, M., Tetreault, J., & Chodorow, M. (2012). Identifying high-level organizational elements in argumentative discourse. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 20–28.

Marcu, D. (2000). *The theory and practice of discourse parsing and summarization*. Cambridge, MA: MIT Press.

Maron, M. E. (1961). *Automatic indexing: An experimental inquiry*. Santa Monica, CA: Rand Corporation.

McLachlan, G., Do, K., & Ambroise, C. (2004). *Analyzing microarray gene expression data*. Hoboken, NJ: John Wiley & Sons. http://dx.doi.org/10.1002/047172842X

Mitchell, T. M. (1997). *Machine learning*. Boston, MA: McGraw-Hill.

Mladenic, D. (1998). Turning yahoo into an automatic web-page classifier. In *Proceedings of the 13th European Conference on Artificial Intelligence,* 473–474.

Nagata, R, & Nakatani, K. (2010). Evaluating performance of grammatical error detection to maximize learning effect. In *Proceedings of COLING,* 894–900.

Pendar, N., & Cotos E. (2008). Automatic identification of discourse moves in scientific article introductions. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications,* 62–70, Association for Computational Linguistics. Columbus, Ohio. http://dx.doi.org/10.3115/1631836.1631844

Porter, M.F. (1980). An algorithm for suffix stripping. *Program*, 14 (3), 130–137. http://dx.doi.org/10.1108/eb046814

Saricaoglu, A., & Cotos, E. (2013). A Study of the inter-annotator reliability for an AWE Tool: Research Writing Tutor (RWT). Paper presented at the *Technology for Second Language Learning Conference*, Ames, IA.

Schilder, F. (2002). Robust discourse parsing via discourse markers, topicality and position. *Natural Language Engineering,* 8 (3), 235–255. http://dx.doi.org/10.1017/s1351324902002905

Sebastiani, F. (2002). Machine Learning in automated text categorization. *ACM Computing Surveys,* 34, 1–47. http://dx.doi.org/10.1145/505282.505283

Shrout, P., & Fleiss, J. L. (1979). Intraclass correlation: Uses in assessing rater reliability. *Psychological Bulletin*, 86 (2), 420–428. http://dx.doi.org/10.1037/0033-2909.86.2.420

Stamatatos, E., Fakotakis, N., & Kokkinakis, G. (2000). Automatic text categorization in terms of genre and author. *Computational Linguistics,* 26 (4), 471–495. http://dx.doi.org/10.1162/089120100750105920

Swales, J. M. (1990). *Genre analysis.* Cambridge: Cambridge University Press.

Taboada, M., & Mann, W. C. (2006). Rhetorical Structure Theory: Looking back and moving ahead. *Discourse Studies*, 8 (3), 423–459. http://dx.doi.org/10.1177/1461445606061881

Toms, E. G., & Campbell, D. G. (1999). Genre as interface metaphor: Exploiting form and function in digital environments. In *System Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference On*. IEEE. http://dx.doi.org/10.1109/hicss.1999.772652

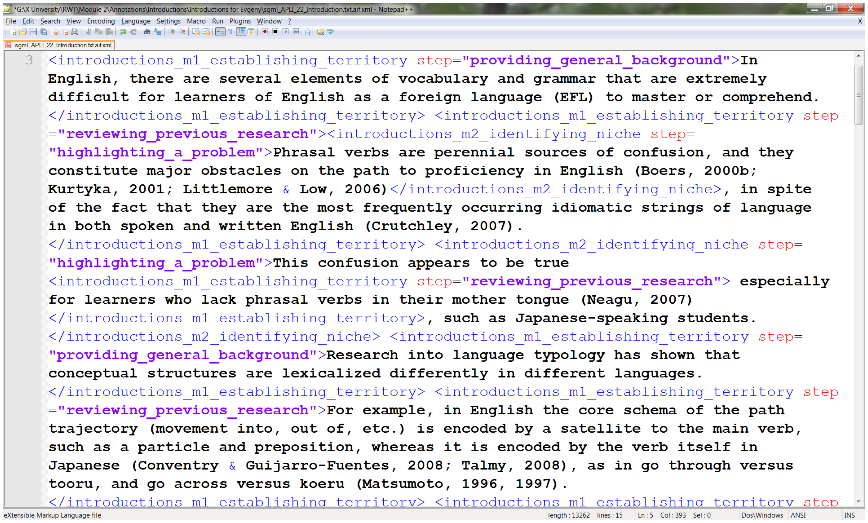
Upton, T., & Connor, U. (2001). Using computerized corpus analysis to investigate the text-linguistic discourse moves of a genre. *English for Specific Purposes,* 20 (4), 313–329. http://dx.doi.org/10.1016/S0889-4906(00)00022-3

Van Rijsbergen, C. J. (1979). *Information Retrieval* (2nd ed.). London: Butterworth.

Yang, J. C., & Akahori, K. (1998). Error analysis in Japanese writing and its implementation in a computer assisted language learning system on the World Wide Web. *CALICO Journal*, 15 (1–3), 47–66.

Appendix A

Excerpt from an annotated text in Applied Linguistics.


*G:\X University\RWT\Module 2\Annotations\Introductions\Introductions for Evgeny\sgml_APLI_22_Introduction.txt.aif.xml - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
sgml_APLI_22_Introduction.txt.aif.xml
eXtensible Markup Language file
length : 13262 lines : 15 Ln : 5 Col : 393 Sel : 0 Dos\Windows ANSI INS

```
<introductions_m1_establishing_territory step="providing_general_background">In
English, there are several elements of vocabulary and grammar that are extremely
difficult for learners of English as a foreign language (EFL) to master or comprehend.
</introductions_m1_establishing_territory> <introductions_m1_establishing_territory step
="reviewing_previous_research"><introductions_m2_identifying_niche step=
"highlighting_a_problem">Phrasal verbs are perennial sources of confusion, and they
constitute major obstacles on the path to proficiency in English (Boers, 2000b;
Kurtyka, 2001; Littlemore & Low, 2006)</introductions_m2_identifying_niche>, in spite
of the fact that they are the most frequently occurring idiomatic strings of language
in both spoken and written English (Crutchley, 2007).
</introductions_m1_establishing_territory> <introductions_m2_identifying_niche step=
"highlighting_a_problem">This confusion appears to be true
<introductions_m1_establishing_territory step="reviewing_previous_research"> especially
for learners who lack phrasal verbs in their mother tongue (Neagu, 2007)
</introductions_m1_establishing_territory>, such as Japanese-speaking students.
</introductions_m2_identifying_niche> <introductions_m1_establishing_territory step=
"providing_general_background">Research into language typology has shown that
conceptual structures are lexicalized differently in different languages.
</introductions_m1_establishing_territory> <introductions_m1_establishing_territory step
="reviewing_previous_research">For example, in English the core schema of the path
trajectory (movement into, out of, etc.) is encoded by a satellite to the main verb,
such as a particle and preposition, whereas it is encoded by the verb itself in
Japanese (Conventry & Guijarro-Fuentes, 2008; Talmy, 2008), as in go through versus
tooru, and go across versus koeru (Matsumoto, 1996, 1997).
</introductions_m1_establishing_territory> <introductions_m1_establishing_territory step
```

Appendix B

RA Introduction sub-corpus used for training

Discipline	*# Texts*	*# Sentences*	*# Words*
1. Accounting	13	542	12,971
2. Aerospace Engineering	13	339	7,830
3. Agricultural & Bio-Systems Engineering	13	277	6,247
4. Agronomy	13	267	7,338
5. Analytical & Physical Chemistry	13	319	7,049
6. Animal Science	13	141	3,327
7. Applied Linguistics	13	358	9,418
8. Architecture	13	203	4,973
9. Art & Design	13	242	5,513
10. Bioinformatics & Computational Biology	13	380	8,576
11. Biochemistry & Biophysics	13	246	6,873
12. Biomedical Sciences	13	342	7,578
13. Business	13	205	4,960
14. Chemical Engineering	13	335	7,628
15. Computer Engineering	13	296	7,532
16. Computer Science	13	377	7,790
17. Curriculum & Instruction	13	300	6,390
18. Database Management	13	342	9,031
19. Economics	13	381	8,131
20. Electrical Engineering & Power Systems	13	384	8,654
21. Electrophysiology	13	440	8,582
22. Environmental Engineering	13	296	6,565
23. Hospitality Management	13	238	5,478
24. Food Science	2	46	1,039
25. Forestry	11	225	5,768
26. Geological & Atmospheric Sciences	13	269	6,756
27. Genetics	13	338	7,201
28. Health & Human Performance	13	347	9,115
29. Immunobiology	13	217	5,739
30. Inorganic Chemistry	13	285	7,051
31. Industrial Engineering	7	167	3,726
32. Journalism	13	318	7,509
33. Mathematics	13	444	10,842
34. Molecular Biology	13	342	7,340
35. Mechanical Engineering	13	201	4,341
36. Meteorology	13	269	6,268
37. Microbiology	13	343	8,996
38. Material Science & Engineering	13	247	5,750
39. Nano-Scale & Heat Transfer	13	215	4,982
40. Organic Chemistry	13	239	5,271
41. Public Administration	6	120	2,683
42. Physics & Astronomy	13	246	5,744
43. Plant Breeding	13	395	9,158
44. Plant Physiology	13	266	6,941
45. Psychology	13	299	7,758
46. Sociology	13	390	8,760
47. Special Education	13	558	15,337
48. Statistics	13	194	5,288
49. Toxicology	13	347	7,043
50. Community & Regional Planning	13	196	4,446
51. Veterinary Medicine	13	365	8,894
Total	650	15,460	366,089