Finding the WRITE Stuff: Automatic Identification of Discourse Structure in Student Essays

Jill Burstein, *ETS Technologies*

Daniel Marcu and Kevin Knight, *Information Sciences Institute, University of Southern California*

An essay-based discourse analysis system can help students improve their writing by identifying relevant essay-based discourse elements in their essays. The system presented here uses a voting algorithm based on decisions from three independent discourse analysis systems to automatically label elements in student essays.

Automated essay-scoring technologies can enhance both large-scale assessment and classroom instruction. Essay evaluation software not only numerically rates essays but also analyzes grammar, usage, mechanics, and discourse structure.[1,2] In the classroom, such applications can supplement traditional instruction by giving students automated feedback that helps them revise their work and ultimately improve their writing skills. These applications also address educational researchers' interest in individualized instruction. Specifically, feedback that refers explicitly to students' own writing is more effective than general feedback.[3]

Our discourse analysis software, which is embedded in Criterion (www.etstechnologies.com), an online essay evaluation application, uses machine learning to identify discourse elements in student essays. The system makes decisions that exemplify how teachers perform this task. For instance, when grading student essays, teachers comment on the discourse structure. Teachers might explicitly state that the essay lacks a thesis statement or that an essay's single main idea has insufficient support. Training the systems to model this behavior requires human judges to annotate a data sample of student essays. The annotation schema reflects the highly structured discourse of genres such as persuasive writing.

Our discourse analysis system uses a voting algorithm that takes into account the discourse labeling decisions of three independent systems. The three systems employ natural language processing methods to extract essay-based features that help predict the discourse labels. They also use machine learning to classify the sentences in an essay as particular discourse elements. Our tool automatically labels discourse elements in student essays written on any topic and across writing genres.

Essay-based discourse

Researchers have proposed a variety of discourse analysis schemes to capture the semantics of multisentence texts. Some schemes associate a hierarchical representation to a given text, while others a linear one. The representation used in our work is linear. It assumes that essays can be segmented into sequences of discourse spans and that each span is associated with an overall communicative goal. We focus on essay-specific communicative goals, which we encode using intuitive labels that are frequently used in teaching writing, such as thesis statements, main ideas, and conclusion statements.

Essay annotation protocol

To facilitate development of our discourse analysis systems, two human judges annotated several hundred essays. The judges labeled elements in the essay data according to a protocol that explained how to annotate several discourse categories:

- *Title* segments indicate essay titles.
- *Introductory material* segments provide the context or set the stage in which the thesis, a main idea, or the conclusion is to be interpreted.

1094-7167/03/$17.00 © 2003 IEEE
Published by the IEEE Computer Society

- *Thesis* segments state the writer's position statement and are related to the essay prompt.
- *Main idea* segments assert the author's main message in conjunction with the thesis.
- *Supporting idea* segments provide evidence and support the claims made in the main ideas, thesis statements, or conclusions.
- *Conclusion* segments summarize the essay's entire argument.
- *Irrelevant* segments do not fit into the other discourse categories and do not meaningfully contribute to the essay.

Figure 1 shows an example annotated essay.

Test questions on standardized tests and in classroom instruction often elicit *persuasive* or *informative* essays. Persuasive writing requires students to state their opinion on a topic and to validate that opinion with convincing arguments. An informative prompt also requires students to state their opinion on a topic but might suggest that students use personal experiences and observations to substantiate their opinion. Informative essays often involve more descriptive writing. Both genres adhere to strict discourse strategies that require at least a thesis statement, several main and supporting ideas, and a conclusion.

<Introductory material> "You can't always do what you want to do," my mother said. She scolded me for doing what I thought was best for me. It is very difficult to do something that I do not want to do. **</Introductory material> <Thesis>** But now that I am mature enough to take responsibility for my actions, I understand that many times in our lives we have to do what we should do. However, making important decisions, like determining your goal for the future, should be something that you want to do and enjoy doing**. </Thesis>**

<Introductory material> I've seen many successful people who are doctors, artists, teachers, designers, etc. **</Introductory material> <Main point>** In my opinion they were considered successful people because they were able to find what they enjoy doing and worked hard for it**. </Main point> <Irrelevant>** It is easy to determine that he/she is successful, not because it's what others think, but because he/she have succeed in what he/she wanted to do. **</Irrelevant>**

<Introductory material> In Korea, where I grew up, many parents seem to push their children into being doctors, lawyers, engineer etc. **</Introductory material> <Main point>**Parents believe that their kids should become what they believe is right for them, but most kids have their own choice and often doesn't choose the same career as their parent's**. </Main point> <Support>** I've seen a doctor who wasn't happy at all with her job because she thought that becoming doctor is what she should do. That person later had to switch her job to what she really wanted to do since she was a little girl, which was teaching. **</Support>**

<Conclusion> Parents might know what's best for their own children on a daily basis, but deciding a long term goal for them should be one's own decision of what he/she likes to do and wants to do. **</Conclusion>**

Figure 1. An annotated essay. Judges manually labeled all sentences as belonging to one of seven specified categories.

Annotation process and agreement results

In the beginning, or pretraining, phase, judges practiced annotation on an initial set of approximately 50 essays from three essay prompts, which called for both persuasive and informative writing styles.

During the training phase, judges were allowed to discuss their decisions as they labeled identical sets of essays on three topics. Using a program that implements J.S. Uebersax's kappa computation,[4] we regularly ran kappa statistics to ensure that the judges maintained a kappa of at least 0.8 for all categories. The kappa statistic measures pairwise agreement among a set of judges who make categorical judgments, correcting for chance expected agreement. Research in content analysis suggests that kappa values higher than 0.8 reflect high agreement.[5]

In the final annotation phase, the judges could not discuss the essays. They annotated independent data sets of 120 essays for each of the three prompts used in the pretraining phase, with the exception of 40 overlapping essays. To ensure consistent labeling, we ran kappa statistics on their independent judgments to measure agreement on the overlapping cases. Again, if kappa for any particular category fell below 0.8, the judges reviewed the protocol (but did not discuss individual essays).

Agreement between judges is critical in machine-learning applications because the system learns from their decisions. Table 1 shows a high level of agreement on the overlapping sets: kappa (K) values are usually larger than 0.8. Table 1 also shows precision (P), recall (R), and F-measure figures that reflect the relative performance among human judges:

- *Precision*: the number of cases in which J1 and J2 agree divided by the number of cases labeled by J2, where J1 = human judge 1 and J2 = human judge 2
- *Recall*: the number of cases in which J1

Table 1. Agreement between two human judges for 40 essay responses to prompts A, B, and C (precision, recall, and *F*-measure).

Prompt	A				B				C			
Discourse element	***K***	***P***	***R***	***F***	***K***	***P***	***R***	***F***	***K***	***P***	***R***	***F***
Conclusion	1.00	1.00	1.00	1.00	0.90	0.91	0.92	0.91	0.96	0.97	0.98	0.97
Introductory Material	0.86	0.82	0.94	0.88	0.82	0.85	0.82	0.83	0.88	0.97	0.81	0.89
Main points	0.85	0.87	0.86	0.87	0.85	0.86	0.87	0.87	0.96	0.96	0.96	0.96
Other*	1.00	1.00	1.00	1.00	0.56	1.00	0.50	0.67	0.79	0.74	0.88	0.80
Support	0.96	0.99	0.98	0.98	0.90	0.95	0.97	0.96	0.96	0.99	0.98	0.98
Thesis	0.92	0.97	0.89	0.93	0.77	0.82	0.78	0.80	0.94	0.92	0.99	0.96
System-wide	0.95	0.97	0.97	0.97	0.86	0.92	0.92	0.92	0.94	0.97	0.97	0.97

* Because the original categories *title* and *irrelevant* occur infrequently, we collapsed them and any unlabeled text into the category *other* and used this for training and testing.

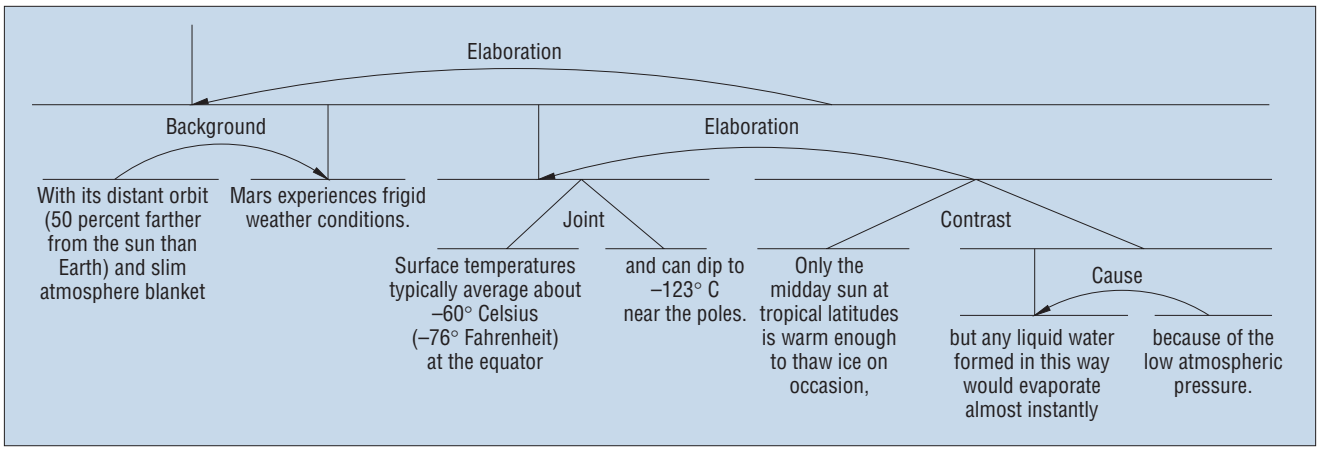


Figure 2. Example rhetorical structure tree. Nuclei are represented as straight lines and satellites as arcs. Names assigned to internal nodes reflect their rhetorical relation.

and J2 agree divided by the number of cases labeled by J1
- *F-measure*: 2 × precision × recall/(precision + recall)

Finally, the judges annotated approximately 250 essay responses to six prompts (A, B, C, G, H, and N). The data sets included persuasive and informative writing from 12th graders and first-year college students.

Automated discourse analysis

To train our systems, we used identical data sets categorized by our two human judges as introductory material, thesis, main ideas, supporting ideas, conclusion, title, and other. We experimented with both decision-based and probabilistic systems. Our discourse analysis systems use different feature sets and methodologies to label all sentences in an essay.

Decision-based discourse analyzer

We use C5.0, a decision-tree machine-learning algorithm with boosting, to get the best model. For model building, several feature extraction programs identify various discourse-relevant features for all sentences from a training sample of essays. We input these feature vectors into C5.0, which generates a model for subsequent labeling. To label new, unseen data, the system reads in an essay, and the feature extraction programs find the relevant features for each sentence. Another program creates feature vectors for the sentences. The C5.0 model reads in each vector and classifies each sentence (vector) on the basis of the C5.0 feature set, which is composed of the following elements.

RST rhetorical relations and status. According to rhetorical structure theory (RST), people can associate a rhetorical structure tree to any text.[6] The tree's leaves correspond to elementary discourse units, and the internal nodes correspond to contiguous text spans. A *status* (nucleus or satellite) and a *rhetorical relation*—a relation that holds between two nonoverlapping text spans—characterize each tree node. The nucleus represents elements that are more essential to the writer's intention than those expressed by the satellite. Moreover, a rhetorical relation's nucleus is comprehensible independent of the satellite, whereas without the nucleus, the satellite is incomprehensible. When spans are equally important, the relation is multinuclear.

Rhetorical relations reflect semantic, intentional, and textual relations that hold between text spans, as Figure 2 illustrates. For example, one text span might elaborate on another text span, the information in two text spans might differ, or the information in one text span might provide background for the information presented in another. Figure 2 displays a text fragment's rhetorical structure tree in the Mann and Thompson style.[6] The figure represents nuclei as straight lines and satellites as arcs. The parser labeled internal nodes with rhetorical relation names (elaboration, background, and so on).

We build rhetorical structure trees automatically for each essay using Marcu's cue-phrase-based discourse parser,[7] which assigns RST rhetorical relations and status to essay sentences. We associate a feature with each sentence in an essay that reflects the status of its parent node (nucleus or satellite), and another feature that reflects its rhetorical relation. For example, we associate the status *satellite* and the relation *elaboration* to the last sentence in Figure 2 because it is the satellite of an elaboration relation. It associates the status *nucleus* and the relation *elaboration* to sentence 1 because it is the nucleus of an elaboration relation.

Discourse marker words, terms, and structures. A discourse analysis submodule identifies cue words, terms, and syntactic structures that function as discourse markers. Earlier research indicates that these elements relate to the organization of ideas in an essay.[8] For example, the lexicon specifies classes of cue words containing information about whether or not the item is a discourse development term. So, in the sentence, "I think that people should travel to new places because it enhances their perspective," "because" marks the development of the idea that "people should travel to new places." A cue word class might indicate the beginning of a new argument, such as when "first" occurs as an adverbial conjunct, as in the sentence, "First, I think that people should travel to new places." Syntactic structures, such as infinitive clauses, also indicate new arguments. For example, infinitive clauses that begin sentences and occur toward the beginning of a paragraph tend to mark the beginning of a new argument. These cue words, terms, and structures correspond to particular essay-based discourse elements.

Lexical items for general essay and category-specific language. Empirical analyses[1,9] show that particular words and terms characterize

two sublanguages: a general essay sublanguage and another related to certain discourse categories. "Should," "might," "agree," "disagree," and "I" relate to the general essay sublanguage; the lexical items "opinion" and "feel" link specifically to thesis statements; and the term "in conclusion" clearly relates to conclusions. The system uses these kinds of words and terms to predict discourse labels.

Syntactic structure and sentence mechanics. In addition to rhetorical and lexical elements, some syntactic structures and grammatical features are relevant to essay-based discourse elements. We identify the following syntactic units in sentences:

- Subordinating clauses
- Complement clauses
- Infinitive clauses
- Relative clauses
- Auxiliary verbs

Four features relate to sentence and paragraph position:

- The sentence number within the essay
- The sentence number within a paragraph
- The paragraph number in which the sentence occurs
- The relative position of the paragraph in which the sentence occurs (for example, first paragraph, body paragraph, or final paragraph)

Sentence-final punctuation also relates to essay-based discourse. Our system considers four types of final punctuation:

- Full stop
- Question mark
- Exclamation point
- No sentence-final punctuation

Probabilistic-based discourse analyzers

Our probabilistic-based discourse analyzers are couched in the noisy-channel framework, as Figure 3 shows. In this framework, we assume that a stochastic process that assigns a probability $P(\mathbf{L})$ to every label sequence $\mathbf{L} = l_1\, l_2 \ldots l_n$ generates a vector of discourse labels **L**. Intuitively, we want this stochastic process to assign high probabilities to likely sequences, which resemble the sequences found in the training data, and low probabilities to unlikely sequences. For example, given the nature of essay writing,

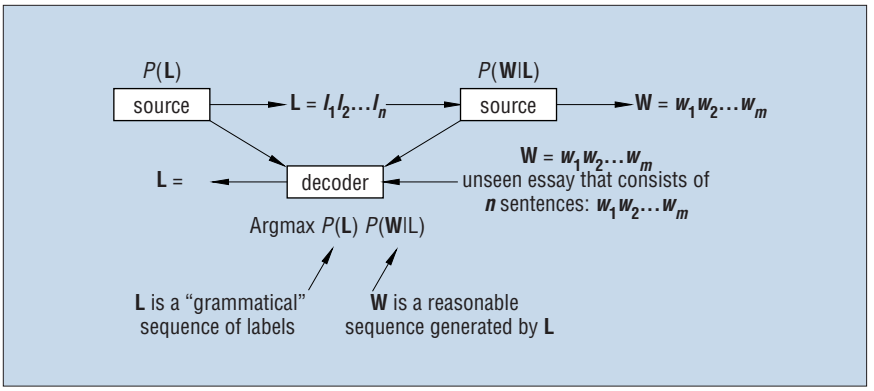


Figure 3. The noisy-channel framework used by the probabilistic discourse labelers. L represents a sequence of discourse labels, and W represents its corresponding sequence of words. *P*(L) represents the probability of the sequence of labels L, while *P*(W|L) represents the probability of the sequence L to generate the sequence W.

the sequence Thesis Main_idea Supporting_idea Supporting_idea Conclusion Conclusion is more likely and should have higher probability than the sequence Conclusion Supporting_idea Main_idea Thesis Main_idea Conclusion. (In the training data, we found no sequence of labels starting with a conclusion sentence or with a thesis sentence surrounded by two main idea sentences.)

We also assume that each label $l_i \in \mathbf{L}$ passes through a noisy channel and maps into a sentence in a student-written essay. For example, when passed through the noisy channel, the first Thesis label maps into the first thesis sentence in Figure 1, "But now that I am mature enough to take responsibility for my actions, I understand that many times in our lives we have to do what we should do." The next Thesis label maps into the second thesis sentence, "However, making important decisions, like determining your goal for the future, should be something that you want to do and enjoy doing." If we model the channel properly, the probability that a Thesis label generates such a sentence should be greater than the probability of it generating a sentence such as "In Korea, where I grew up, many parents seem to push their children into being doctors, lawyers, engineer, etc," which the human annotators labeled as introductory material.

Channel modeling. Our application uses a simple noisy-channel model. We assume that a probabilistic finite-state transducer automatically generates each pair <sequence of discourse labels, sequence of sentences (essay)>. The transducer has eight states, as Figure 4 shows:

- One start state, which is also the final state
- Six intermediary states, one for each discourse label recognized by the system
- One end_of_sentence state, to facilitate transitions between the start and intermediary states

From the start state, the system moves on an epsilon/null transition (*e*) with equal probability to any intermediary state. In each intermediary state, the system generates either

- A word on an epsilon transition according to a given probability and remains in the same intermediary state
- A discourse label when it receives as input an end-of-sentence special character (When this occurs, the system ends in the end_of_sentence state from which it returns to the start state with probability one)

The diagram in Figure 4 partially represents the finite-state transducer. The finite-state machine assumes that the words in each sentence are generated independently according to probabilities estimated from the training data using simple maximum-likelihood techniques.[10] The word "conclusion," for example, is more likely to appear in the conclusion state than in the introductory material state.

Language modeling. Next, we need a model that assigns probability to each conceivable sequence of discourse labels. These probabilities should describe future label sequences: If a sequence has high probability, it should appear frequently in the test data. We use the language model to select among competing hypotheses suggested by the channel model. Its reasoning is completely independent of actual words; rather, it concentrates on determining which label sequence is most likely.

Our training data consists of 1,179 sequences correctly labeled by human annotators. In the example sequences in Figure 5, we use BR to denote paragraph breaks. Even from this small set, we can already see patterns, such as main ideas often being followed by supporting ideas, and sequences containing a single contiguous block of thesis labels. We

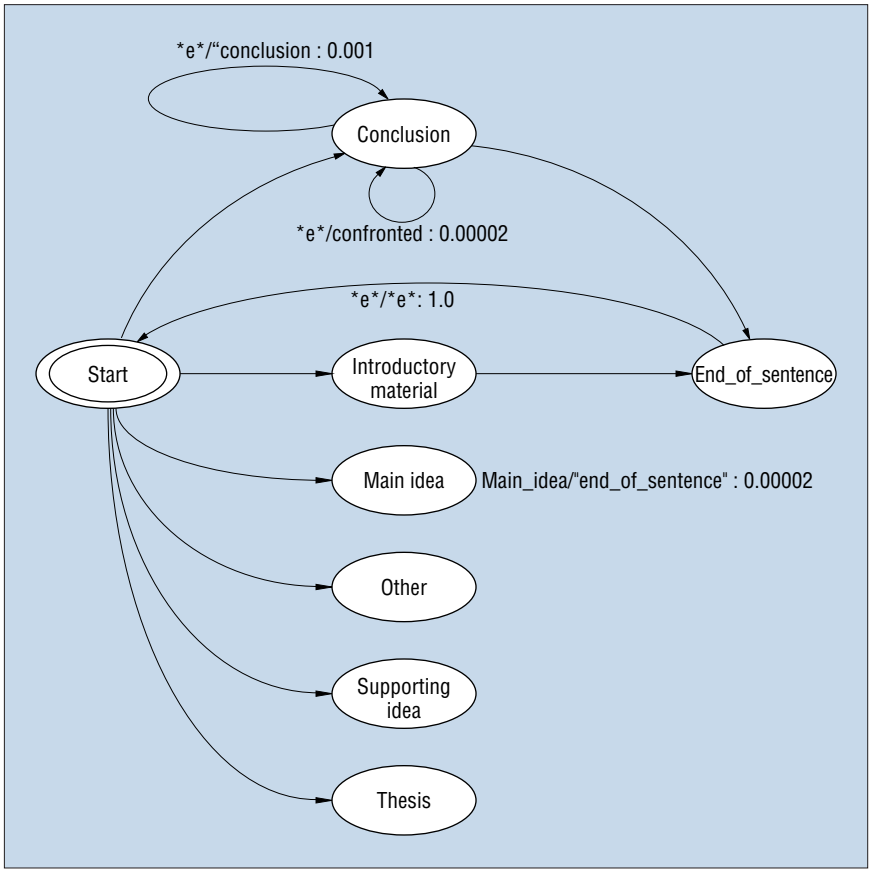


Figure 4. Partial representation of the finite-state transducer that implements the channel model. The transducer has eight states, represented as ovals, corresponding to the start and end of a sentence as well as the discourse labels recognized by the system.

Other BR Introductory_material Introductory_material Introductory_material BR Thesis Thesis Thesis BR Main_idea Supporting_idea Supporting_idea

Thesis Thesis BR Main_point Supporting_idea Supporting_idea Supporting_idea Supporting_idea BR Main_point Supporting_idea Supporting_idea Supporting_idea Supporting_idea BR Main_point Supporting_idea Supporting_idea Supporting_idea BR Conclusion Conclusion

Introductory_material Introductory_material Introductory_material BR Main_point Supporting_idea Supporting_idea Supporting_idea Supporting_idea Supporting_idea Supporting_idea Supporting_idea Supporting_idea Supporting_idea Supporting_idea BR Thesis Supporting_idea Supporting_idea Supporting_idea Supporting_idea Supporting_idea Supporting_idea BR Main_point Supporting_idea Supporting_idea Supporting_idea Supporting_idea BR Conclusion

Figure 5. Sample sequences from the training data. Human annotators labeled 1,179 sequences taken from student essays.

model the probabilities associated with label sequences using two models: the *local language model* and the *global language model*.

The local language model exploits local dependencies among labels. We approximate the sequence's probability by pretending that each new label depends only on the previous two labels:

$$P(l_1\, l_2\, l_3 \ldots l_n) \approx P(l_1 \mid \text{START}) * P(l_2 \mid l_1\ \text{START}) * P(l_3 \mid l_1\, l_2) * \ldots * P(\text{END} \mid l_n\, l_{n-1})$$

This trigram model lets us estimate individual probabilities such as $P(l_3 \mid l_1\, l_2)$ directly from the 1,179-sequence training set—every time the subsequence $l_1\, l_2$ appears, we tabulate what comes next. Sparse data creates a technical problem in which many tabulated trigrams have zero probability, although they do in fact occur in unseen test data. To smooth these probabilities we use an interpolation formula:

$$P(l_3 \mid l_1\, l_2) \approx \lambda_3 * \text{count}(l_1\, l_2\, l_3) / \text{count}(l_1\, l_2) + \lambda_2 * P(l_3 \mid l_2) + \lambda_1 * P(l_3) + \lambda_0$$

We estimate the lambdas using iterative expectation maximization (EM) training. This model discriminates channel-generated hypotheses by virtually ruling out any sequence containing strange subsequences.

Local language modeling does not capture global effects. If we ask the model to stochastically generate sequences of discourse labels, for example, we observe a lack of overall coherence that does not match the kinds of sequences we observe. Many such global affects exist. For example, 96 percent of sequences contain a single block of thesis statements—a pattern local language modeling might not catch. As another example, a conclusion tends to come directly after a thesis statement followed by n blocks of Main_idea/Supporting_idea/BR. If $n > 2$, this tendency is extremely strong (88 percent); if $n < 3$ (47 percent), the tendency is weaker.

We modeled such affects by manually creating a finite-state network of grammatical label sequences containing 43 nodes and 70 transitions. Each transition represents a contiguous block of a certain label type. We expand the transitions to allow the finite-state machine to generate 1, 2, 3, and so on labels within the block. Our finite-state network did not account for 13 percent of the training sequences. Many of the sequences were rare and had no obvious explanation (such as discontiguous thesis blocks), and straightforwardly accounting for all of them would amount to removing useful grammatical constraints. Finally, we trained transition probabilities from our corpus of 1,179 sequences using the EM algorithm.

We represent both language models straightforwardly as probabilistic finite-state acceptors.

Probabilistic systems. We train the channel and language models using simple maximum-likelihood techniques and the EM algorithm.[10] When given as input a sequence **W** of unlabeled sentences (essay), we associate

Table 2. Performance of positional baseline, decision-based, and probabilistic systems (precision, recall, and *F*-measure).

	Positional baseline			Decision-based			Probabilistic systems								
							Base			Local			Global		
Agreement	***P***	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***
Overall system	71	70	70	81	81	81	69	70	69	78	80	79	80	79	79

discourse labels with each sentence by searching for the sequence of discourse labels **L** that maximizes the product of the channel and language model probabilities:

$$\mathbf{L} = \text{argmax}_{\mathbf{L}} P(\mathbf{L})\, P(\mathbf{W} \mid \mathbf{L})$$

Because we represent both the channel and language models as probabilistic finite-state machines, finding the most probable sequence of discourse labels **L** means finding the path of maximum probability in the final state machine that results from composing the channel and language model machines.

Using the noisy-channel framework, we can easily experiment with three systems:

- A *base* system assigns discourse labels using only the lexical information in each sentence. It uses the channel model to assign labels to sentences in an essay.
- A *local* system assigns discourse labels based on both the lexical information in each sentence and the local, trigram-based language model.
- A *global* system assigns discourse labels based on the lexical information in each sentence and the global essay-based language model.

Positional baseline algorithm

Our baseline system uses a position-based selection algorithm that assigns discourse labels to sentences in an essay. The algorithm implements rules that operationalize regularities specific to the discourse annotations in the training corpus. The rules are

- *Introductory material*: Select the first sentence of the essay.
- *Thesis statement*: Select all text in the first paragraph except the first sentence.
- *Main ideas*: Select the first sentence of all body paragraphs.
- *Supporting ideas*: Select all text in the body paragraphs, except the first sentence.
- *Conclusion*: Select all text in the final paragraph.
- *Other*: Select all header and footer text, and sentences with no final punctuation. (A preprocessing program identifies all header and footer text.)

Evaluation

We're interested in addressing three system-related questions:

- Can our automatic discourse analyzers assign discourse labels at performance levels higher than the baseline?
- Given that our systems use different techniques and features to infer the discourse label associated with each sentence, can we combine the outputs of the individual systems to produce a system that outperforms the systems individually?
- Can our systems analyze essays on topics other than those used for training? To assess the systems' generalizability—that is, their applicability to a variety of essay topics—we estimate the degree to which their performance is affected when we test them on data belonging to an essay topic different from those used for training.

Individual system performance

We evaluated the positional baseline system, the decision-based system, and the probabilistic-based systems using sample essays from all prompts in the training and cross-validation sets. Even though these sets are independent, both contain sample essays from all prompt topics. Table 2 compares the overall performance of the decision- and probabilistic-based systems to the positional baseline. Three of the four systems (decision-based, probabilistic-local, and probabilistic-global) significantly outperform the baseline.

In a 10-fold cross-validation, we use nine-tenths of the data for model building and the remaining one-tenth for testing, randomly resampling the data 10 times. Each time we used a different nine-tenths of the data for training and a different one-tenth for testing. The total data set includes 1,462 human-annotated essays.

We use independent samples to create training and test sets from the 1,462 annotated essays. The training set contained 1,179 annotated essays from six prompts. The test set contained the remaining 283 annotated essays from the six prompts.

Voting in topic-dependent systems

We tried combining the output of multiple systems to improve overall performance. We built a voting system using the label assignments for each sentence in an essay from three systems: decision-based, probabilistic-based, and probabilistic-local systems. (We didn't include the probabilistic-global system because it was too time consuming for a commercial application.) We found that a decision-based voting method performed better than a majority vote. Because a decision-based method can use the information in the decision-tree model to evaluate instances in which all three systems assign a different label, we used C5.0 for our voting models.

Table 3 compares the positional baseline system, the best single system (that is, the decision-based system), and a voting system. For the single system, the results in Table 3 represent the same runs used in Table 2 for the decision-based system

Using the 10-fold cross-validation, the voting algorithm outperforms the baseline algorithm and the single system at both the category and overall system levels.

Topic independence

In a classroom, teachers can give students writing assignments on any topic. Because annotating data on all possible essay topics is impractical, we must evaluate a system's ability to generate reliable labeling decisions independent of essay topic.

We train the best single system and the voting system with five prompts and hold the sixth prompt for testing. This lets us evaluate system performance on new data, independent of the essay topic. In the topic-independent (TI) sets TIA, TIB, TIC, TIG, TIH, and TIN, the final letter indicates the essay response set used for testing. There are approximately 1,200 essays in each training set and 250 essays in the test sample.

Both the single system and the voting system clearly exceed baseline system performance. Voting shows a slight increase in mean system performance as compared to a single system. More notably, the voting algorithm shows better performance at the category level than the single system for introductory material, thesis, and conclusion categories.

Table 4 shows that our discourse systems are generalizable—that is, we can use them to label essay responses to prompts for which no training material is available. Figure 6

Table 3. Category agreement with a human judge: best single system and voting system (precision, recall, and *F*-measure)

	Positional baseline			Best single system			Voting system		
Agreement	***P***	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***
Category									
Introductory material	35	23	28	44	23	30	68	50	57
Conclusion	56	67	61	79	83	81	84	84	84
Main point	71	74	73	76	83	81	76	78	77
Other	28	56	37	93	63	75	90	66	76
Support	92	78	84	88	91	89	89	93	91
Thesis	42	63	51	60	67	63	74	73	73
Overall system	71	70	70	81	81	81	85	85	85

Table 4. Topic-independent system agreement with a human judge: best single system and voting system (precision, recall, and *F*-measure)

	Positional baseline			Best single system			Voting system		
Agreement	***P***	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***
Prompt									
TIA	78	78	78	82	82	82	82	82	82
TIB	74	74	74	81	81	81	81	81	81
TIC	79	79	79	80	80	80	81	81	81
TIG	63	63	63	79	79	79	80	80	80
TIH	56	56	56	73	73	73	75	75	75
TIN	54	53	54	74	74	74	74	74	74
System mean	67	67	67	78	78	78	79	79	79

shows the category-specific performance differences between the baseline, single system, and voting system.

Most students rely on off-the-shelf spell checkers and grammar-checking software to enhance the quality of their written work. These types of applications can help students improve some aspects of their writing, namely grammar and mechanics. Certainly, students will continue to use such tools because improvement in these areas will remain critical to a students' ability to produce high-quality essays.

As students become more sophisticated writers, they start thinking about discourse organization and development in their writing. For this, they must consider discourse structure. Our discourse analysis tool offers students feedback about their essays' discourse structure. It gives students a comprehensive analysis of the discourse elements in their essays. For instance, if the system feedback indicates that a student's essay has no conclusion, the student can work on this organizational aspect of the essay. This kind of automated feedback, which resembles traditional teacher feedback, is an initial step in helping students improve their essays' organization and development.

Our discourse analysis tool is part of a larger commercial application—the Criterion online essay evaluation—that also offers feedback on grammar, mechanics, word usage, style, and general essay quality. Criterion is the first application to offer this automated discourse analysis capability. Teachers are excited about its potential as a supplement to writing instruction and as a step toward individualized instruction in the classroom. ■

The Authors

Jill Burstein is a codirector of research at ETS Technologies. Her research interests include NLP-based applications for automatic analysis of writing, discourse analysis, intelligent tutoring, and educational technology. She received her bachelor's degree in linguistics and Spanish from New York University, and her master's and PhD in linguistics from the City University of New York, Graduate Center. She is a member of the Association for Computational Linguistics. Contact her at ETS Technologies, Rosedale Rd., MS 18E, Princeton, N.J. 08541; jburstein@etstech.com.

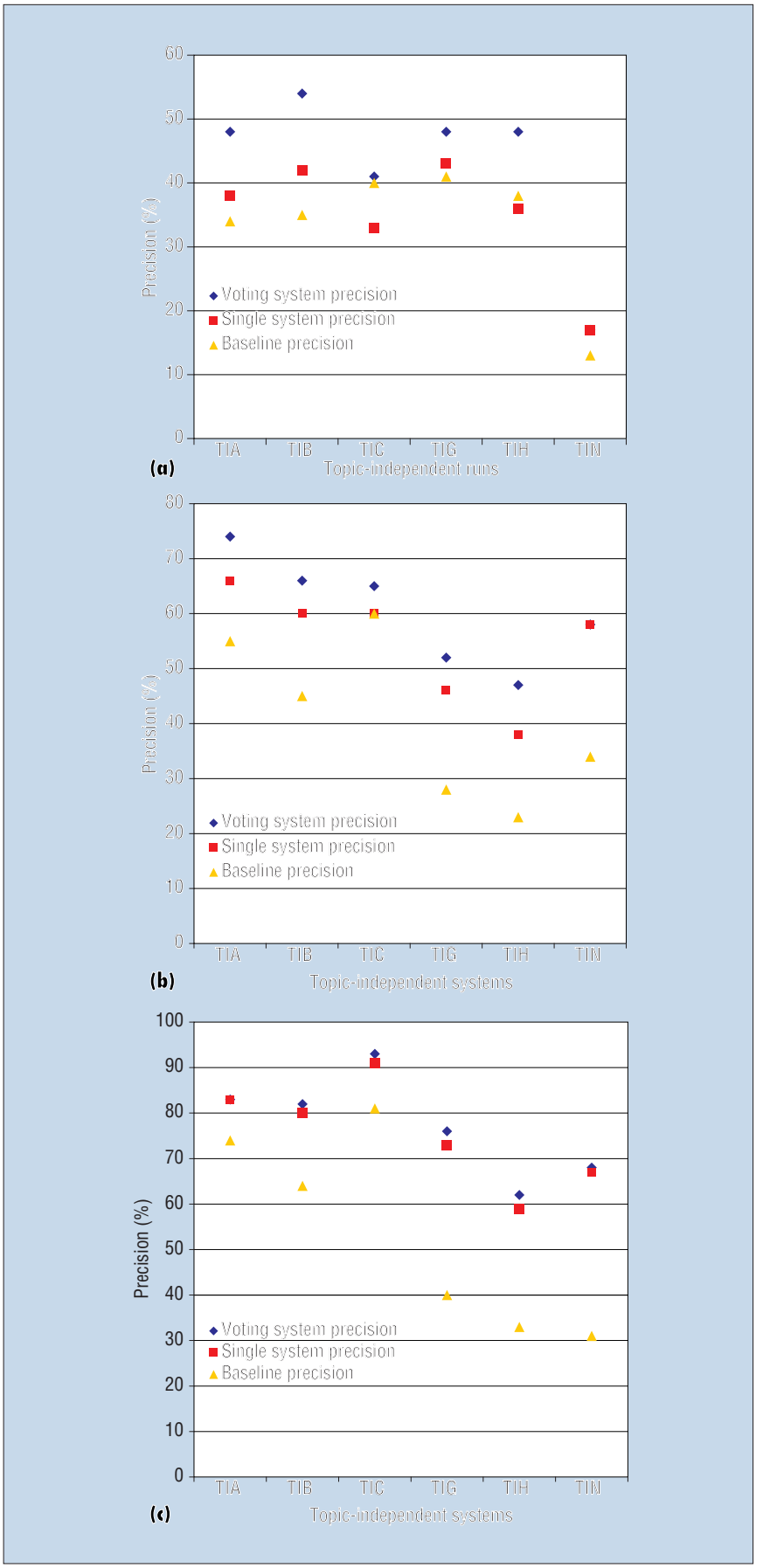
Daniel Marcu is a senior research scientist and project leader at the Information Sciences Institute, University of Southern California, and a research assistant professor in the university's Computer Science Department. His research interests include text and discourse theories, machine translation, summarization, and question answering. He received his bachelor's degree in automation and computers from the Technical University of Cluj, Romania, and his master's and PhD in computer science from University of Toronto. He is a member of the Association for Computational Linguistics and the AAAI. Contact him at Information Sciences Inst., Univ. of Southern California, 4676 Admiralty Way, Ste. 1001, Marina del Rey, CA 90292; marcu@isi.edu.

Kevin Knight is a senior research scientist and project leader at the Information Sciences Institute, University of Southern California, and a research associate professor in the university's Computer Science Department. His research interests include machine translation, natural language generation, and text summarization. He received his bachelor's degree in computer science from Harvard University and his PhD in computer science from Carnegie Mellon University. He is a member of the Association for Computational Linguistics and the AAAI. Contact him at Information Sciences Inst., Univ. of Southern California, 4676 Admiralty Way, Ste. 1001, Marina del Rey, CA 90292; knight@isi.edu.

Acknowledgments

We owe considerable thanks to Slava Andreyev for discussions during the development of the decision-based discourse analysis system and the voting system, and for ongoing data preparation and system implementation. We thank Marisa Farnum and Hilary Persky for their significant contributions to the annotation protocol, and Jennifer Geoghan and Jessica Miller for their annotation work. The kappa calculation program was generously provided by Giovanni Flammia. We are grateful to Richard Swartz for continuous support of this research.

Figure 6. Agreement between human judges and the voting system for three categories: (a) introductory material, (b) thesis statements, and (c) conclusions.



References

1. J. Burstein and D. Marcu, "Developing Technology for Automated Evaluation of Discourse Structure in Student Essays," *Automated Essay Scoring: A Cross-Disciplinary Perspective*, M.D. Shermis and J. Burstein, eds., Lawrence Erlbaum Assoc., Mahwah, N. J., 2003, pp. 209–229.

2. C. Leacock and M. Chodorow, "Automated Grammatical Error Detection," *Automated Essay Scoring: A Cross-Disciplinary Perspective,* M.D. Shermis and J. Burstein, eds., Lawrence Erlbaum Assoc., Mahwah, N.J., 2003, pp. 195–207.

3. M. Scardamalia and C. Bereiter, "Development of Dialectical Processes in Composition," *Literacy, Language, and Learning: The Nature of Consequences of Reading and Writing,* D.R. Olson, N. Torrance, and A. Hildyard, eds., Cambridge Univ. Press, Cambridge, UK, 1985.

4. J.S. Uebersax, "A Generalized Kappa Coefficient," *Educational and Psychological Measurement*, vol. 42, Spring 1982, pp. 181–183.

5. K. Krippendorff, *Content Analysis: An Introduction to Its Methodology,* Sage, Thousand Oaks, Calif., 1980.

6. W.C. Mann and S.A. Thompson, "Rhetorical Structure Theory: Toward a Functional Theory of Text Organization," *Text,* vol. 8, no. 3, 1988, pp. 243–281.

7. D. Marcu, *The Theory and Practice of Discourse Parsing and Summarization,* MIT Press, Cambridge, Mass., 2000.

8. J. Burstein et al., "Enriching Automated Essay Scoring Using Discourse Marking," *Proc. Workshop Discourse Relations and Discourse Marking*, Assoc. for Computational Linguistics, East Stroudsburg, Pa., 1998, pp. 15–21.

9. J. Burstein et al., "Toward Automatic Classification of Discourse Elements in Essays," *Proc. 39th Ann. Meeting Assoc. for Computational Linguistics*, Assoc. for Computational Linguistics, East Stroudsburg, Pa., 2001, pp. 90–97.

10. C. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, Mass., 1999.

For more on this or any other computing topic, see our Digital Library at http://computer.org/publications/dlib.