Modeling Organization in Student Essays

Isaac Persing, Alan Davis, and Vincent Ng
Human Language Technology Research Institute
University of Texas at Dallas

EMNLP 2010 – Boston, MA
October 9, 2010
Automated Essay Scoring

• Important educational application of NLP
• Recent academic research
  – Technical errors
  – Coherence
  – Relevance to prompt
    Little work done on modeling organization
What Is Organization?

• Structure of an essay’s argument
  – Writers must: introduce topic, state their position, give support, conclude argument
  – Transitions between *functions* of discourse structures

• Related work on organization
  – E-rater, v.2 (Attali and Burstein, 2004; 2006)
  – Counts number of discourse segments present:
    • 1 thesis, 3 main ideas, 3 supporting ideas, 1 conclusion
Contributions

• New computational model of organization
• New corpus annotated with organization scores
Overview

Corpus and Annotations

• Labeling Discourse Structures

• Organization Scoring Methods
  – Heuristic-Based Methods
  – Learning-Based Methods

• Experimental Results
Selecting a Corpus

• International Corpus of Learner English (ICLE)
  – 4.5 million words in more than 6000 essays
  – Written by university undergraduates who are learners of English as a foreign language
  – Mostly (91%) argumentative writing topics
    • Contain the discourse structures we want to model

• Essays selected for annotation
  – 1003 argumentative, untimed essays
Scoring Rubric

4 – essay is **very well structured** and is organized in a way that logically develops an argument

3 – essay is **fairly well structured** but could somewhat benefit from reorganization

2 – essay is **poorly structured** and would greatly benefit from reorganization

1 – essay is **completely unstructured** and requires major reorganization

• Half-point increments (i.e., 1.5, 2.5, 3.5) allowed
Annotator Training and Selection

• 30 applicants familiarized with scoring rubric and given sample essays to annotate
• Discussed essay scores with coordinator and other annotators until consensus reached on best scores
• Selected 6 applicants with highest consistency on 8 sample essays
Inter-Annotator Agreement

- Subset of 846 essays scored by 2 annotators
- Compare scores between pairs of annotators to calculate inter-annotator agreement
- Perfect agreement on only 29% of essays
- Scores within 0.5 point on 71% of essays
- Scores within 1.0 point on 93% of essays
Overview

Corpus and Annotations

Labeling Discourse Structures

• Organization Scoring Methods
  – Heuristic-Based Methods
  – Learning-Based Methods

• Experimental Results
Functions of Discourse Structures

• Organization refers to an argument’s structure

• Essential elements of an argument:
  – Introduce topic, state position, give support, conclude

• If these elements are missing or out of order, then organization is poor

Knowing the *functions* of discourse structures is helpful to score an essay’s organization
Paragraph Function Labels

• Identify discourse function of paragraphs
• 4 paragraph function labels:
  – Introduction
  – Body
  – Conclusion
  – Rebuttal
Paragraph Function Labeling

• Label paragraphs heuristically

• Features used to label a paragraph’s function:
  – Position of paragraph within essay
    • e.g., First paragraph is likely an Introduction
  – Types of sentences within paragraph
    • e.g., Support sentence Body paragraph
      Requires that we label sentences as well
Sentence Function Labels

• Identify discourse function of sentences
• 10 sentence function labels:
  — Prompt
  — Transition
  — Thesis
  — Main Idea
  — Elaboration
  — Support
  — Conclusion
  — Rebuttal
  — Solution
  — Suggestion
Sentence Function Labeling

- Label sentences heuristically
- Features used to label a sentence’s function:
  - Position of sentence within paragraph
    - e.g., Last sentence is likely a conclusion
  - Words (unigrams) and punctuation
    - e.g., “agree” | “think” | “opinion”  Thesis
Overview

Corpus and Annotations
Labeling Discourse Structures
Organization Scoring Methods
  – Heuristic-Based Methods
  – Learning-Based Methods

• Experimental Results
Heuristic-Based Organization Scoring

• Two heuristic methods to score organization
• Both methods use nearest neighbor approach:
  1) Find $k$ essays most similar to test essay $e$
  2) Predict $e$’s organization score by aggregating the scores of its $k$ nearest neighbors found in step 1
• These methods differ by:
  How do we find similar essays?
  How do we aggregate scores?
Method 1: Finding Similar Essays

• Essays have labeled paragraphs (e.g., *IBBBC*)
• Organization depends on *transitions* between paragraph functions
  – *Sequence* of labels is what’s important
• Find similar paragraph label sequences
  – e.g., *IBBBC* similar to *IBBRC*

Use *sequence alignment* algorithm to calculate similarity score for any pair of label sequences
Aligning Label Sequences

• Needleman-Wunsch algorithm finds an optimal alignment of a pair of sequences

• Scoring function $S(a, b)$ is set heuristically:
  
  $S(a, b) = +1$  when $a = b$  (reward for match)
  
  $S(a, b) = −1$  when $a ≠ b$  (penalty for mismatch)
  
  $S(a, −) = S(−, a) = −1$  (penalty for indel)

• Aligning *IBBBC* with *IBBRC* scores +3 (similar)

• Aligning *IBBBC* with *CRRRI* scores −5 (dissimilar)
Method 1: Scoring Organization

1) Find \( k \) essays most similar to test essay \( e \)
   • Calculate similarity score between essay \( e \) and each essay in the training set by aligning their sequences of paragraph labels

2) Predict test essay \( e \)’s organization score by aggregating its \( k \) nearest neighbors’ scores
   • 3 ways to aggregate scores (mean, median, mode)
   \( H_\rho \) has 3 variations
Method 2: Finding Similar Paragraphs

- Paragraphs have labeled sentences
- Organization also depends on transitions between sentence functions
- Find similar paragraphs by aligning sentence label sequences
- Associate each similar paragraph with its essay’s organization score
Method 2: Scoring Organization

1) For each paragraph $p_i$ of test essay $e$:
   a) Find $k$ paragraphs most similar to $p_i$
      - Calculate similarity score between paragraph $p_i$ and each paragraph in the training set by aligning their sequences of *sentence* labels
   b) Score $p_i$ by aggregating $k$ nearest neighbors’ scores
      - 3 ways to aggregate scores (mean, median, mode)

2) Predict $e$’s organization score by aggregating its paragraphs’ scores obtained in step 1b
   - 3 ways to aggregate scores (mean, median, mode)
Heuristic-Based Scoring Methods

• Total of 12 heuristic-based scoring methods:
  – 3 variants of $H_p$ (using paragraph label sequences)
  – 9 variants of $H_s$ (using sentence label sequences)

Which of these 12 variations is the best?
How should we combine these methods?
Learning-Based Organization Scoring

• Use learning system to decide which methods to combine to predict organization score
  – $\text{SVM}^{\text{light}}$ implementation of regression SVMs

• Three different approaches:
  – $R_l$ uses linear kernel
  – $R_s$ uses string kernel
  – $R_a$ uses alignment kernel
Regression with Linear Kernel

• $R_l$ incorporates three types of features:
  – Nearest neighbor score predictions from $H_p$ and $H_s$
  – Paragraph-label subsequences of length 1 to 5
    • Give learner more direct access to paragraph labels
  – Sentence-label subsequences of length 1 to 5
    • Organization depends on order of sentence functions
Regression with String Kernel

• SVMs enable the use of \textit{structured} features (e.g., sequences) rather than only \textit{flat} features (i.e., discrete- or real-valued)

• $R_s$ uses \textit{string kernel} to efficiently compute similarity between paragraph label sequences based on common subsequences of length 3
Regression with Alignment Kernel

• Kernels compute similarity between examples
  Sequence alignment algorithm does this too!
  – Use alignment scores as kernel values
  – $R_a$ uses *alignment kernel* to compute similarity

• Kernel must always return non-negative value
  – Increase each score by the lower bound to ensure all are non-negative
Regression with Composite Kernel

• We want a learner to use *multiple* kernels
• Use *composite kernel*:

\[
K_c(F_1, F_2) = \frac{1}{n} \sum_{i=1}^{n} K_i(F_1, F_2)
\]

where \(F_1\) and \(F_2\) are two essays’ features
Overview

Corpus and Annotations
Labeling Discourse Structures
Organization Scoring Methods
  Heuristic-Based Methods
  Learning-Based Methods
Experimental Results
Evaluation Metrics

• Define 3 evaluation metrics:

\[ S_1 = \frac{1}{N} \sum_{A_i \neq E_i} 1 \] (frequency of error)

\[ S_2 = \frac{1}{N} \sum_{i=1}^{N} |A_i - E_i| \] (mean error distance)

\[ S_3 = \frac{1}{N} \sum_{i=1}^{N} (A_i - E_i)^2 \] (mean squared error)

\( A_i \) and \( E_i \) are annotated and estimated scores
Baseline Scoring System

• No standard baseline for scoring organization
• \( \text{Avg} \) – assigns the average organization score of essays in training set
  – Any score prediction system using information in the essay should be able to beat this
• Simple, but not easy to beat
  – 41% of essays have score of 3
  – 96% of essays have score within 1 point of 3
Heuristics-Based Scoring Systems

<table>
<thead>
<tr>
<th>System</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg</td>
<td>.585</td>
<td>.412</td>
<td>.348</td>
</tr>
<tr>
<td>$H_p$</td>
<td>.548</td>
<td>.339</td>
<td>.198</td>
</tr>
<tr>
<td>$H_s$</td>
<td>.575</td>
<td>.397</td>
<td>.329</td>
</tr>
</tbody>
</table>

- Both $H_p$ and $H_s$ outperform Avg baseline.
- $H_p$ performs significantly ($p < 0.01$) better than both Avg and $H_s$ systems under $S_2$ and $S_3$.

Examining the transition of paragraph functions is more important than with sentence functions.
Learning-Based Scoring Systems

<table>
<thead>
<tr>
<th>System</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg</td>
<td>.585</td>
<td>.412</td>
<td>.348</td>
</tr>
<tr>
<td>$H_p$</td>
<td>.548</td>
<td>.339</td>
<td>.198</td>
</tr>
<tr>
<td>$H_s$</td>
<td>.575</td>
<td>.397</td>
<td>.329</td>
</tr>
<tr>
<td>$R_l$</td>
<td>.520</td>
<td>.331</td>
<td>.186</td>
</tr>
</tbody>
</table>

- $R_l$ performs better than Avg, $H_p$ and $H_s$
- Results are not significant, even at $p < 0.1$
  - Only major benefit of $R_l$ is that it combines all 12 heuristic methods, so we don’t have to choose one
  - $H_p$ is a fairly effective heuristic scoring method
Learning-Based Scoring Systems

<table>
<thead>
<tr>
<th>System</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Avg$</td>
<td>.585</td>
<td>.412</td>
<td>.348</td>
</tr>
<tr>
<td>$H_p$</td>
<td>.548</td>
<td>.339</td>
<td>.198</td>
</tr>
<tr>
<td>$H_s$</td>
<td>.575</td>
<td>.397</td>
<td>.329</td>
</tr>
<tr>
<td>$R_l$</td>
<td>.520</td>
<td>.331</td>
<td>.186</td>
</tr>
<tr>
<td>$R_s$</td>
<td>.577</td>
<td>.369</td>
<td>.222</td>
</tr>
</tbody>
</table>

- $R_s$ performs better than $Avg$ and $H_s$ ($S_2$ and $S_3$) — Extracts useful information from paragraph labels
- $R_s$ performs significantly worse than $H_p$ and $R_l$ — Nearest neighbor features are very valuable
Learning-Based Scoring Systems

<table>
<thead>
<tr>
<th>System</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg</td>
<td>.585</td>
<td>.412</td>
<td>.348</td>
</tr>
<tr>
<td>$H_p$</td>
<td>.548</td>
<td>.339</td>
<td>.198</td>
</tr>
<tr>
<td>$H_s$</td>
<td>.575</td>
<td>.397</td>
<td>.329</td>
</tr>
<tr>
<td>$R_f$</td>
<td>.520</td>
<td>.331</td>
<td>.186</td>
</tr>
<tr>
<td>$R_s$</td>
<td>.577</td>
<td>.369</td>
<td>.222</td>
</tr>
<tr>
<td>$R_a$</td>
<td>.686</td>
<td>.519</td>
<td>.429</td>
</tr>
</tbody>
</table>

- $R_a$ performs significantly ($p < 0.01$) worse than $R_s$
  - Alignment kernel *appears* to not be extracting any useful information from paragraph label sequences.
Learning-Based Scoring Systems

<table>
<thead>
<tr>
<th>System</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg</td>
<td>.585</td>
<td>.412</td>
<td>.348</td>
</tr>
<tr>
<td>$H_p$</td>
<td>.548</td>
<td>.339</td>
<td>.198</td>
</tr>
<tr>
<td>$H_s$</td>
<td>.575</td>
<td>.397</td>
<td>.329</td>
</tr>
<tr>
<td>$R_l$</td>
<td>.520</td>
<td>.331</td>
<td>.186</td>
</tr>
<tr>
<td>$R_s$</td>
<td>.577</td>
<td>.369</td>
<td>.222</td>
</tr>
<tr>
<td>$R_a$</td>
<td>.686</td>
<td>.519</td>
<td>.429</td>
</tr>
</tbody>
</table>

- $R_l$ performs best among learning-based methods
- $R_l$ and $H_p$ are statistically indistinguishable
- $R_a$ performs significantly worse than $R_s$ and $R_l$
## Composite Kernel Scoring Systems

<table>
<thead>
<tr>
<th>System</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg</td>
<td>.585</td>
<td>.412</td>
<td>.348</td>
</tr>
<tr>
<td>$H_p$</td>
<td>.548</td>
<td>.339</td>
<td>.198</td>
</tr>
<tr>
<td>$H_s$</td>
<td>.575</td>
<td>.397</td>
<td>.329</td>
</tr>
<tr>
<td>$R_l$</td>
<td>.520</td>
<td>.331</td>
<td>.186</td>
</tr>
<tr>
<td>$R_s$</td>
<td>.577</td>
<td>.369</td>
<td>.222</td>
</tr>
<tr>
<td>$R_a$</td>
<td>.686</td>
<td>.519</td>
<td>.429</td>
</tr>
<tr>
<td>$R_{ls}$</td>
<td>.534</td>
<td>.332</td>
<td>.187</td>
</tr>
<tr>
<td>$R_{la}$</td>
<td>.541</td>
<td>.332</td>
<td>.178</td>
</tr>
<tr>
<td>$R_{sa}$</td>
<td>.517</td>
<td>.325</td>
<td>.177</td>
</tr>
</tbody>
</table>

- $R_{sa}$ performs best among 2-kernel systems
## Composite Kernel Scoring Systems

<table>
<thead>
<tr>
<th>System</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg</td>
<td>.585</td>
<td>.412</td>
<td>.348</td>
</tr>
<tr>
<td>$H_p$</td>
<td>.548</td>
<td>.339</td>
<td>.198</td>
</tr>
<tr>
<td>$H_s$</td>
<td>.575</td>
<td>.397</td>
<td>.329</td>
</tr>
<tr>
<td>$R_l$</td>
<td>.520</td>
<td>.331</td>
<td>.186</td>
</tr>
<tr>
<td>$R_s$</td>
<td>.577</td>
<td>.369</td>
<td>.222</td>
</tr>
<tr>
<td>$R_a$</td>
<td>.686</td>
<td>.519</td>
<td>.429</td>
</tr>
<tr>
<td>$R_{ls}$</td>
<td>.534</td>
<td>.332</td>
<td>.187</td>
</tr>
<tr>
<td>$R_{la}$</td>
<td>.541</td>
<td>.332</td>
<td>.178</td>
</tr>
<tr>
<td>$R_{sa}$</td>
<td>.517</td>
<td>.325</td>
<td>.177</td>
</tr>
<tr>
<td>$R_{lsa}$</td>
<td>.517</td>
<td>.323</td>
<td>.175</td>
</tr>
</tbody>
</table>
Feature Analysis

• $R_f$ uses three types of flat features:
  – Nearest neighbor score predictions from $H_p$ and $H_s$
  – Paragraph-label subsequences of length 1 to 5
  – Sentence-label subsequences of length 1 to 5

• Feature ablation – remove each feature group independently and find drop in performance
  – Nearest neighbor features are most important
  – Paragraph label sequences are least important
Conclusion

• New computational model of organization
  – Heuristic-based and learning-based methods
• New corpus annotated with organization scores
  – Release corpus to research community