Automatic Identification of Organizational Structure in Writing using Machine Learning

Laurence Anthony and George V. Lashkia

Dept. of Computer Science, Faculty of Engineering
Okayama Univ. of Science, 1-1 Ridai-cho, Okayama

anthony@ice.ous.ac.jp  lashkia@ice.ous.ac.jp

http://antpc1.ice.ous.ac.jp
Presentation Outline

- Background
- Research Aim
- System Design (Overview)
- Application to Research Abstracts
- Results (Accuracy)
- Results (Effectiveness in the Classroom)
- Software Demonstration
- Conclusions
Background

- Importance of Text Structure
- Studies on Text Structure
  - INTRODUCTIONs - Swales (1990), Anthony (1999)
  - PATENTS - Bazerman (1994)
  - GRANT PROPOSALS - Connor & Mauranen (1999)
  - LEGAL WRITING - Bhatia (1993)
Background

- Problems with Analyzing Text Structure
  - We need a large corpus of text data
    (The text data must ‘ACURATELY’ represent what we hope to study)
  - We need a lot of research time
    (We must analyze a lot of texts)
  - We need good validation and reliability tests
    (Because evaluating structure can be very subjective)

- Most Text Structure Studies are ‘Small Scale’
Background

- Henry et al. (2001)
  - 40 Application Letters
- Tarone et al. (2000)
  - 2 Physics Research Articles
- Connor et al. (1999)
  - 34 Grant Proposals
- Williams (1999)
  - 5 Medical Research Articles
- Anthony (1999)
  - 12 Computer Science Research Article Introductions
Research Aim

- Develop a Computer System to Process Texts and Analyze Text Structure Automatically
  - A 'Machine Learning System’ for text structure
    - Easy to process a large corpus of text data
    - Fast
    - The analytic process would be clearly defined
    - Easy to test the reliability and validity
System Design (Overview)

- **Machine Learning: Unsupervised ? Supervised Learning’?**

- **In Supervised Learning,**
  - Give the system a structural model (set of classes)
  - Give the system examples of the model
  - Tell the system what ‘features’ in the examples are important
  - Define a relation between the classes and the features
  - Classify new text examples by comparing its features with those in each class
System Design (Overview)

- Problems
  - We need a ‘good’ model of structure
    - But there are many models of structure in the literature
  - We need a set of ‘labeled examples’
    - But many systems work well with only a few labeled examples
  - We need a ‘good’ set of features
    - But language contains a LOT of noise words!
      (e.g. a, the, of, in, at, but?, though?, ...)
    - Building a list of features by hand is infeasible
  - We need a ‘good’ relation between the classes and the features
Application of System to Research Abstracts

- Give the system a structure model:
  'Modified’ CARS Model (Swales, 1990: Anthony, 1999)

**Move 1** Establishing a Territory
  1.1 Claiming centrality
  1.2 Making topic generalizations
  1.3 Reviewing items of previous research

**Move 2** Establishing a niche
  2.1A Counter claming
  2.1B Indicating a gap
  2.1C Question raising
  2.1D Continuing a tradition

**Move 3** Occupying the niche
  3.1A Outlining purpose
  3.1B Announcing present research
  3.2 Announcing principal findings
  3.3 Evaluation of research
  3.4 Indicating RA structure
Application of System to Research Abstracts

- **Give the system examples of the model**
  - 100 Abstracts (IEEE Trans. on PDS) divided into 692 labeled ‘Steps Units’ (only examples from 6 classes)
  - 554 Step Units (80%) used for ‘training’ the system
  - 138 Step Units (20%) used for ‘testing’ the system

- **Tell the system what ‘features’ to look at**
  - All word clusters (chunks) up to 5 words long
  - Position of step unit in abstract (i.e. 1st line, 2nd line, ...)

- **(Reduce ‘Noise’ in Features)**
  - Automatically rank words by ‘importance’ using: raw frequency, Information Gain
  - Use only high ranked words
Application of System to Research Abstracts

- “In this paper, we propose a new system.”
  - 1 word chunks
    - in/ this/ paper/ we/ propose/ a/ new/ system
  - 2 word chunks
    - in this/ this paper/ we/ propose/ a/ new/ system
  - 3 words chunks
    - in this paper/ this paper we/ propose/ a/ new/ system
Application of System to Research Abstracts

- “In this paper, we propose a new system.”
  - 1 word chunks
    - in/ this/ paper/ we/ propose/ a/ new/ system
  - 2 word chunks
    - in this/ this paper/ paper we/ we propose/ propose a/ a new/ new system
  - 3 word chunks
    - in this paper/ this paper we/ paper we propose/ we propose a/ propose a new/ a new system
  - ...

-...
Information Gain (IG)

Information Gain (IG) is a commonly used measure in decision tree algorithms to evaluate the quality of a split. It quantifies the reduction in entropy achieved by partitioning the data into subsets based on a particular attribute.

Mathematically, the Information Gain (IG) for a feature $w$ is defined as:

$$Gain(D, w) = Entropy(D) - \sum_{v \in Values(w)} \left( \frac{|D_v|}{|D|} \cdot Entropy(D_v) \right)$$

where $Values(w)$ is the set of all possible values for word $w$, $D_v$ is the subset of $D$ for which the word $w$ has a value $v$. $D$ is the dataset, and $D_v$ is the subset of $D$ for which the word $w$ has a value $v$.

The entropy of a dataset $D$, $Entropy(D)$, is calculated as:

$$Entropy(D) = \sum_{j=1}^{c} -p_j \log_2 p_j$$

where $p_j$ is the proportion of data (D) in a class $j$ from the set of classes $C$. This measures the impurity or randomness of the data distribution across classes.
## Information Gain (IG)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Raw Frequency</th>
<th>Information Gain (IG)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>the</td>
<td>however</td>
</tr>
<tr>
<td>2</td>
<td>a</td>
<td>2_however</td>
</tr>
<tr>
<td>3</td>
<td>to</td>
<td>difficult_to</td>
</tr>
<tr>
<td>4</td>
<td>in</td>
<td>is_often</td>
</tr>
<tr>
<td>5</td>
<td>of</td>
<td>transmitting</td>
</tr>
<tr>
<td>6</td>
<td>is</td>
<td>often</td>
</tr>
<tr>
<td>7</td>
<td>and</td>
<td>not</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>difficult</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>task_migration</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>Process</td>
</tr>
</tbody>
</table>
Define a relation between features and classes
- Use probability of each class and the probability of features (clusters) being in each class

(A NAÏVE BAYES Classifier)

- Class 1 (Claiming Centrality)
- Class 2 (Making topic generalizations)
- Class 3 (Indicating a gap)
- Class 4 (Outlining purpose)
- Class 5 (Announcing principal findings)
- Class 6 (Evaluation of research)

| Class 1 | Class 1 Prob. | Feat. 1 prob. | Feat. 2 prob. | Feat. 3 prob. | ... |
| Class 2 | Class 2 Prob. | Feat. 1 prob. | Feat. 2 prob. | Feat. 3 prob. | ... |
| Class 3 | Class 3 Prob. | Feat. 1 prob. | Feat. 2 prob. | Feat. 3 prob. | ... |
| Class 4 | Class 4 Prob. | Feat. 1 prob. | Feat. 2 prob. | Feat. 3 prob. | ... |
| Class 5 | Class 5 Prob. | Feat. 1 prob. | Feat. 2 prob. | Feat. 3 prob. | ... |
| Class 6 | Class 6 Prob. | Feat. 1 prob. | Feat. 2 prob. | Feat. 3 prob. | ... |
Application of System to Research Abstracts

- Classify the structure of new text examples
  - Choose the most probable class containing the features in each step unit.
    - "This paper is an effort in the same direction"
      (Step 3.1B - Announcing Present Research”)

- Features Contained in Training Data
  - paper (c3), this_paper (c4), is (c14) this (c18) the (c39)
  - 2 (c103) is_an (c364) in (c571)

Step 1.1 Prob. = -2.9498 + -7.0449 + -7.0449 + -4.3368 + ... + -4.4058 = -48.7690
Step 1.2 Prob. = -1.8398 + -7.4899 + -7.4899 + -3.8523 + ... + -3.8790 = -45.5972
Step 2.1B Prob. = -3.1391 + -6.9157 + -6.9157 + -4.3507 + ... + -4.2076 = -47.0826
Step 3.1B Prob. = -1.3335 + -4.1566 + -4.2436 + -4.8497 + ... + -3.9169 = -39.0836
Step 3.2 Prob. = -1.8398 + -6.3677 + -6.3677 + -3.6936 + ... + -3.7837 = -40.8448
Step 3.3 Prob. = -1.5809 + -6.6178 + -6.6178 + -3.7846 + ... + -4.0528 = -43.2638

- Most Probable Step ...
Application of System to Research Abstracts

- Classify the structure of new text examples
  - Choose the most probable class containing the features in each step unit.
    - “2 this paper is an effort in the same direction”
      (Step 3.1B - Announcing Present Research”)

- Features Contained in Training Data
  - paper (c3), this_paper (c4), is (c14) this (c18) the (c39)
    2 (c103) is_an (c364) in (c571)

  Step 1.1 Prob. = -2.9498 + -7.0449 + -7.0449 + -4.3368 + ... + -4.4058 = -48.7690
  Step 1.2 Prob. = -1.8398 + -7.4899 + -7.4899 + -3.8523 + ... + -3.8790 = -45.5972
  Step 2.1B Prob. = -3.1391 + -6.9157 + -6.9157 + -4.3507 + ... + -4.2076 = -47.0826
  **Step 3.1B Prob. =** -1.3335 + -4.1566 + -4.2436 + -4.8497 + ... + -3.9169 = -39.0836
  Step 3.2 Prob. = -1.8398 + -6.3677 + -6.3677 + -3.6936 + ... + -3.7837 = -40.8448
  Step 3.3 Prob. = -1.5809 + -6.6178 + -6.6178 + -3.7846 + ... + -4.0528 = -43.2638

- Most Probable Step = h step 3.1B = -39.0836
  (Decision is Step 3.1B “Announcing Present Research”)
## Results (Classification Accuracy)

- **Classification Accuracy (Overall)**
  - 554 Step Units used for ‘training’ the system (80% of entire data)
  - 138 Step Units used for ‘testing’ the system (20% of entire data)

<table>
<thead>
<tr>
<th>No. of Features</th>
<th>Accuracy (Raw Frequency)</th>
<th>Accuracy (Information Gain)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2208 (all)</td>
<td>56 %</td>
<td>-</td>
</tr>
<tr>
<td>1000</td>
<td>51 %</td>
<td>70 %</td>
</tr>
<tr>
<td>700</td>
<td>56 %</td>
<td>70 %</td>
</tr>
<tr>
<td>500</td>
<td>59 %</td>
<td>69 %</td>
</tr>
<tr>
<td>300</td>
<td>59 %</td>
<td>69 %</td>
</tr>
<tr>
<td>100</td>
<td>54 %</td>
<td>-</td>
</tr>
</tbody>
</table>

**Note:** Random guessing has an accuracy of 16.66% (NOT 50%!) Choosing the most common class = 26%
Results (Classification Accuracy)

- Classification Accuracy (Each Step Unit)
  - Number of features = 700
  - Ranked by Information Gain measure
  - Accuracy (overall) = 70%

<table>
<thead>
<tr>
<th>Class</th>
<th>Step 1.1</th>
<th>Step 1.2</th>
<th>Step 2.1b</th>
<th>Step 3.1b</th>
<th>Step 3.2</th>
<th>Step 3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1.1</td>
<td>2 (43 %)</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Step 1.2</td>
<td>0</td>
<td>17 (77 %)</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Step 2.1b</td>
<td>0</td>
<td>2</td>
<td>1 (17 %)</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Step 3.1b</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>34 (92 %)</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Step 3.2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>25 (66 %)</td>
<td>9</td>
</tr>
<tr>
<td>Step 3.3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td>17 (61 %)</td>
</tr>
</tbody>
</table>

Note: Classifications correspond with CARS Model ‘moves’ (Accuracy=88% when using ‘second opinion’)
Results (In the classroom)

- A ‘Windows’ Interface
  - To enable researchers, teachers and students to use the system it needs to be easily accessible via a ‘windows’ interface
  - A ‘windows’ system has been built using the programming language PERL 5.6 and PERL/Tk
## Results (In the classroom)

### Materials Selection by Non-Native Teacher

<table>
<thead>
<tr>
<th>Selection of 7 texts from 10 text corpus</th>
<th>By hand</th>
<th>Using System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to complete tasks</td>
<td>100 min.</td>
<td>28 min. (1 min. for analysis plus time to check results)</td>
</tr>
<tr>
<td>Errors</td>
<td>2/7</td>
<td>1/7</td>
</tr>
</tbody>
</table>
| Comments                               | "The decisions are fast."
|                                        | "It is simple and easy to complete the task."
|                                        | "I rely too much on the software and stop feeling like doing the analysis myself." |
Results (In the classroom)

- Text Analysis by Non-Native Student

<table>
<thead>
<tr>
<th>Selection of 4 texts from 10 text corpus</th>
<th>By hand</th>
<th>Using System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to complete tasks</td>
<td>38 min.</td>
<td>15 min.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1 min. for analysis plus time to check results)</td>
</tr>
<tr>
<td>Errors</td>
<td>2/4</td>
<td>0/4</td>
</tr>
</tbody>
</table>

Comments

- “It’s very fast.”
- “The structure is now very clear.”
- “The system has clearly analyzed the structure, what you should do is correct only the part that is strange. So the work is little.”
Conclusions

- A computer system was developed to analyze text structure
  - Learning method: ‘Supervised Learning’
  - Accuracy 70% (88% when using second opinion)
- System errors corresponded with CARS Model ‘moves’
- Effective in the classroom for use by teachers and students
- Runs in Windows environment