END-TO-END ARGUMENT MINING FOR DISCUSSION THREADS BASED ON PARALLEL CONSTRAINED POINTER ARCHITECTURE

Tokyo University of Agriculture and Technology, Japan.

Gaku Morio (Master course 2nd)
Katsuhide Fujita (Supervisor)

ArgMining 2018 @ EMNLP 2018
BACKGROUND AND MOTIVATION
Background

• Over the past dozen years or so, middle or large scale online discussions are available through online forums.
  • Recently, **online civic discussions** are also highlighted through the forum [Ito 2014, Park2018].


The problem is “massive posts.”

- While we can acquire a lot of posts in a short time by using the online forum, it is hard to understand all of the posts.
- For example, in the online civic discussion in our previous work [Morio 2018] included,
  - Several days for the discussion;
  - 800+ citizens who joined the discussion,
  - 1,300+ posts.

- So, how to understand the enormous opinions?
  - We estimate Argument Mining will do!

Motivation

• In the present study, we focus on argument mining to understand fine-grained opinions in the discussion forum,
  • because extracting premises behind citizens’ claim is important to understand their ideas.
CONTRIBUTIONS OF OUR WORK

Research Overview
Overview of the contributions

• We tackle “end-to-end” Argument Mining for discussion forums.
  • Because there’s no definitive studies about it.
• We provide following two contributions;
  1 A novel inner- and inter- post scheme, and annotations for discussion threads.
  2 End-to-end classification approaches for the scheme.
    • The biggest contribution in this study!
Contribution overview

- Annotation study for discussion threads.
- For this, we provide micro-level inner- and inter-post scheme.
- We first conducted the annotation for Japanese online civic discussion threads.

Our original annotation tool.
Contribution overview

- Parallel Constrained Pointer Architecture (PCPA)
  - PCPA is a novel end-to-end neural model using Pointer Networks [Potash 2017].
  - PCPA can discriminate:
    - A sentence type (i.e., claim, premise or none)
    - An inner-post relation;
    - An inter-post interaction; simultaneously.

Our neural model, PCPA.

CONTRIBUTION

Annotation Study
Argument Mining for discussion threads

• Related works:
  • There are a few studies which employ micro-level scheme for the discussion thread.
  • Also, most of existing work don’t consider multiple writers in the discussion thread.
    • Though [Hidey 2017] provided a micro-level annotation for the discussion thread, the work don’t distinguish inner- and inter- post scheme.

Our scheme for **inner-post argument**

- **We assume** each post as a **stand-alone** discourse.
- Therefore, for each post, an independent argument can be created.

i.e., **claim** and **premise** argument [Stab 2017]

---

Post:170  
**Depth = 0**

I think the municipal subway should introduce an around-the-clock operation.

---

Post:171  
**Depth = 1**

Yes, I think making the subway operating 24 hours is appealing.

---

I want to enjoy Nagoya until late at night.

---

Our scheme for **inter-post interaction**

- **To extract** the **inter-post interaction**, we introduce the interaction model similar to [Ghosh 2014].

---

Annotation

• We annotated our original online civic discussion.
  • The online civic engagement was held in Nagoya city, Japan, in cooperation with the local government.
  • In this study, we employ “sentence-level” annotation because a proposition appears per sentence in most cases.

• The data includes;
  • 399 threads;
  • 1327 posts;
  • 5559 sentences.
Annotation results

- **We acquired state-of-the-art size** of discussion dataset.
  - Also, some properties like a large proportion of premises compared to claims are confirmed.
- However, **inter-annotator agreements are lower** than the essays.
  - We attribute this as following two factors;
    1. Most of citizen’s comments are not well written.
    2. Our sentence-level annotation, rather than token-level.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Type</th>
<th>Size</th>
<th>(\kappa)</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLLAGREEM</td>
<td>Claim</td>
<td>1449</td>
<td>.531</td>
</tr>
<tr>
<td></td>
<td>Premise</td>
<td>2762</td>
<td>.554</td>
</tr>
<tr>
<td></td>
<td>NonArg</td>
<td>1348</td>
<td>.529</td>
</tr>
<tr>
<td></td>
<td>IPR w/ A0</td>
<td>2762</td>
<td>.466</td>
</tr>
<tr>
<td></td>
<td>IPI</td>
<td>745</td>
<td>.430</td>
</tr>
<tr>
<td>Persuasive Essays</td>
<td>Claim</td>
<td>1506</td>
<td>.635</td>
</tr>
<tr>
<td>[Stab2017]</td>
<td>Premise</td>
<td>3832</td>
<td>.833</td>
</tr>
<tr>
<td></td>
<td>Inner-essay rel</td>
<td>3832</td>
<td>.708-.737</td>
</tr>
</tbody>
</table>
CONTRIBUTION 2

Parallel Constrained Pointer Architecture (PCPA)
Parallel Constrained Pointer Architecture (PCPA)

- **PCPA** is a novel neural model which can discriminate;
  - Claim;
  - Premise;
  - Inner-post relation (IPR);
  - inter-post interaction (IPI);

simultaneously (i.e., end-to-end model).
Parallel Constrained Pointer Architecture (PCPA)

• In related works,
  • [Eger 2017] pointed out that end-to-end neural models have advantages in terms of “low error propagation.”
  • Also, [Potash 2017] employed Pointer Networks to discriminate relation target in arguments.

• Thus, in this study we propose an end-to-end model based on Pointer Networks, PCPA.
  • Our PCPA has two Pointer Networks for inner- and inter- relation i.e., parallel architecture.
  • Our PCPA can effectively constrain computation space based on explicit constraints of discussion threads i.e., constrained pointer architecture.
    • So we call our model Parallel Constrained Pointer Architecture (PCPA).

PCPA is composed of:

1. Input module
2. Encoding module
3. Output modules
PCPA is composed of:

1. Input module
2. Encoding module
3. Output modules

For example, assume given following thread with two posts.

![Thread Diagram]

---

PCPA is composed of:

1. Input module
2. Encoding module
3. Output modules

For example, assume given following thread with two posts.

![Thread Diagram]
PCPA is composed of:

1. **Input module**
2. Encoding module
3. Output modules

In the input module, each sentence is converted into sentence representation.

![Diagram](image)
PCPA is composed of:

1. Input module
2. **Encoding module**
3. Output modules

Next, the encoding module with **BiLSTM** acquires context-aware sentence representations.
PCPA is composed of:

1. Input module
2. Encoding module
3. Output modules

The output modules are PCPA’s classification module which has **three output classification layers**.
PCPA is composed of:

1. Input module
2. Encoding module
3. Output modules

First, we explain the **Component Classifier**.
PCPA is composed of:

1. Input module
2. Encoding module
3. Output modules

This layer classifies a sentence type (premise, claim or non-argumentative.)

\[
p(y_k^{\text{type}} | P_j^{(i)}) = \text{softmax}(z_k^{(i,j)})
\]

\[
L_i^{\text{type}} = \sum_{j=1}^{N_i} \sum_{k=1}^{N_{i,j}} \log p(y_k^{\text{type}} | P_j^{(i)})
\]

Objective
PCPA is composed of:

1. Input module
2. Encoding module
3. Output modules

This layer classifies a sentence type (premise, claim or non-argumentative.)

Component Classifier

Input

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Post</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

Reply
PCPA is composed of:

1. Input module
2. Encoding module
3. Output modules

**Pointer Network** can estimate the relation target by a pointer distribution.

Next, the IPR Classifier discriminates *inner-post relations* using Pointer Networks.
PCPA is composed of:

1. Input module
2. Encoding module
3. Output modules

For example, let me explain how to search an inner-post relation (IPR) target of sentence “3.”

E.g. Pointer distribution

![Diagram showing the Pointer Network with input modules, encoding module, and output modules connected through arrows, and specific posts marked with numbers.]
PCPA is composed of:

1. Input module
2. Encoding module
3. Output modules

In this case, the IPR target is “4.” with the max value of the pointer distribution.
PCPA is composed of:

1. Input module
2. Encoding module
3. Output modules

There is a problem; we noticed that the computation space of an ordinal Pointer Network is too wide for our scheme.
PCPA is composed of:

1. Input module
2. Encoding module
3. Output modules

Therefore, PCPA constrains computation space. More specifically, we don’t need to scan out of post distributions in IPR because IPR is an inner-post relation.

\[ L^{i_{pr}}_i = \sum_{j=1}^{N_i} \sum_{k=1}^{N_{ij}} \log p(y^{i_{pr}}_k \mid P^{(i)}_j) \]
PCPA is composed of:

1. Input module
2. Encoding module
3. Output modules

Finally, we explain the inter-post interaction (IPI) layer.
For the IPI classifier, we employ a pointer network similar to the IPR. For example, let’s search IPI target from sentence “5.”

PCPA is composed of:

1. Input module
2. Encoding module
3. Output modules
PCPA is composed of:

1. Input module
2. Encoding module
3. Output modules

In the IPI, PCPA can also constrain computation space, and we don’t need to scan no relevant sentences like “6,7” because IPI is a post-to-post relation.
PCPA is composed of:

1. Input module
2. Encoding module
3. Output modules

In the IPI, PCPA can also constrain computation space, and we don’t need to scan no relevant sentences like “6,7” because IPI is a post-to-post relation.
PCPA is composed of:

1. Input module
2. Encoding module
3. Output modules

In the IPI, PCPA can also constrain computation space, and **we don’t need to scan no relevant sentences** like “6,7” because IPI is a post-to-post relation.

\[ p(y^{ipi}_k \mid P^{(i)}_{j_1}, P^{(i)}_{j_2}) = \text{softmax}([q^{(i,j_1,k)}; q^{(i,j_2,k)}]) \]

\[ L^{ipi}_i = \sum_{(j_1,j_2) \in R^{(i)}} \sum_{k=1}^{N_{i,j_2}} \log p(y^{ipi}_k \mid P^{(i)}_{j_1}, P^{(i)}_{j_2}) \]
PCPA is composed of:

1. Input module
2. Encoding module
3. Output modules

Finally, we arrive at the final objective function.

\[ \text{Loss} = \frac{1}{N} \sum_i \left( -\alpha L_{i\text{pr}}^i - \beta L_{i\text{pi}}^i ight) - (1 - \alpha - \beta) L_{i\text{type}}^i \]
Time complexity

- **PCPA reduces** its time complexity compared to the standard Pointer Networks.
  - Given;
    - The average # of posts in a thread ($n_p$);
    - The average # of sentences in a post ($n_S$),
  - PCPA’s time complexity is $O(n_S^2 \times n_p)$ while the standard Pointer Networks take $O(n_S^2 \times n_p^2)$.
    - You may think $O(n_S^2 \times n_p)$ is large enough, though, the number of sentences per post is not so large in real world.
EXPERIMENTS
Experimental setting

• We employ following state-of-the-art baselines;
  • [Potash 2017] Pointer Networks (Seq2Seq)
    • An ordinal Pointer Networks (w/o constraints.)
  • [Potash 2017] Pointer Networks (no Seq2Seq)
    • Non-sequence-to-sequence model.
  • MTL-BiLSTM similar to [Eger 2017]
    • BiLSTM-based multi-task learning model which doesn’t employ Pointer Networks.

• Our dataset is split into, \textit{train: test} = 8:2.
Performance results

- We show **F1 scores** for each model.
  - We can find from the table that **PCPA significantly outperforms all baselines in terms of IPR and IPI classifications.**
  - This results indicate that **constraining computation space is effective.**

<table>
<thead>
<tr>
<th>Model</th>
<th>Claim F1</th>
<th>Premise F1</th>
<th>NA F1</th>
<th>IPR F1</th>
<th>IPI F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCPA (ours)</td>
<td>58.1</td>
<td>71.5</td>
<td>58.8</td>
<td>*44.3</td>
<td>*26.9</td>
</tr>
<tr>
<td>Pointer Network (Seq2Seq)</td>
<td>58.3</td>
<td>70.8</td>
<td>48.6</td>
<td>27.2</td>
<td>19.4</td>
</tr>
<tr>
<td>Pointer Network (no Seq2Seq)</td>
<td><strong>60.1</strong></td>
<td>71.3</td>
<td>53.1</td>
<td>35.0</td>
<td>20.8</td>
</tr>
<tr>
<td>MTL-BiLSTM</td>
<td>54.2</td>
<td>65.6</td>
<td>56.9</td>
<td>14.9</td>
<td>12.6</td>
</tr>
</tbody>
</table>

For each model, we show the best score, and * indicates significant. at $p < 0.01$, two-sided Wilcoxon signed rank test.
IPR performance according to the thread depth

- We in turn observe performances of inner-post relation (IPR), according to the thread depth.
- In deeper threads, ordinal Pointer Networks (PNs) can’t keep their performances.
  - In contrast, our PCPA (red) can keep the performance even for deeper threads.

F1 for IPR.

→ Thread depth.
IPI performance according to the thread depth

- For inter-post interaction (IPI), our PCPA (red) improves the F1 score for deeper threads.
CONCLUSION
Conclusion

1. We applied **Argument Mining for discussion threads**.
   - Our scheme is based on [Stab 2017] and [Ghosh 2014].

2. We conducted **annotations for discussion threads**.
   - Real online civic discussions are annotated.
   - Inter-annotator agreements are evaluated.

3. We propose **Parallel Constrained Pointer Architecture**
   - The PCPA effectively constrains its computation space, and reduces time complexity.

4. **Experimental results demonstrate**;
   - PCPA outperformed baselines significantly.
   - Constraining computation space is effective for classifying the inner-post relation (IPR) and inter-post interaction (IPI).
ABOUT OUR DATA
Statistics of COLLAGREE data

• About COLLAGREE data
  • Date: from 12.2016 to 1.2017
  • 204 citizens joined
  • 399 threads
  • 1327 threads
  • 5559 sentences

• Average statistics:
  • # of posts per thread: 3.33 (SD 3.29)
  • The depth of a thread: 1.09 (SD 1.19)
  • # of sentences per post: 4.19 (SD 3.33)
  • # of words per sentence: 21.63 (SD 19.92)
Statistics of COLLAGREE data

• **Annotation design**
  
  • Independent three annotators annotate each sentence.
  
  • *Annotation phase* 1 includes classifying each sentence into component types i.e., claim, premise and non-argumentative, and extracting support/attack relationships between them.
  
  • *Annotation phase* 2 includes extracting target/callout relationships between post-to-post interaction.
  
  • We evaluate kappa agreement using Fleiss’ kappa.
Annotation Tool

Sequence Tagging Task (文章タグ付けタスク)[nagoya2016_0]

和便器についてはメーカーが生産中止を発表していたと思います。

今後は無くなっていきまね。

ところで、和便器は、便座に腰を下ろしたくないという人が案外多くて、使用者のリクエストなんだそうです。

人もそれぞれなんですね。

[nan](46)

私も、ちょい潔癖なんで、便座に座りたくないので和式使います。
• We examined position of argument components.

It's not realistic as long as we keep the municipal operations.

We should entrust not only to the subway but such business parts to private sectors.

Privatized parks are getting better and better.
Positions of claim and premise in a post

- This figure below shows a histogram of position of premises and claims in posts with more than two sentences.
  - Claims are tend to appear in the last of the post because citizens are likely to conclude their idea in the last.
Premises’ distance from a claim

• We examined the distance of premises from a claim.

It's not realistic as long as we keep the municipal operations.

We should entrust not only to the subway but such business parts to private sectors.

Privatized parks are getting better and better.
Premises’ distance from a claim

- This figure below shows a histogram of premise Dist.
  - It shows that premises are likely to appear immediately prior to a claim.
  - In fact, the result exhibits the same property on the essay corpus [Eger 2017].
Distinct feature: IDF

• We investigate a histogram of the average Inversed document frequency (IDF) value per argument component (claim and premise) with more than 5 words.
  • The significance of averages shows at $p < 0.0001$. 