PubSE: A Hierarchical Model for Publication Extraction from Academic Homepages — Supplementary Material

Yiqing Zhang **Jianzhong Qi** **Rui Zhang*** **Chuandong Yin**
School of Computing and Information Systems
The University of Melbourne
{yiqingz2,chuandongy}@student.unimelb.edu.au
{jianzhong.qi,rui.zhang}@unimelb.edu.au

A Preprocessing Decisions

HTML document: We do not consider the HTML DOM tree structure because HTML tags vary significantly across different homepages. Learning the DOM tree structure of a publication string itself is a challenging task. Our initial investigation shows that, given our limited data, the DOM tree structure offers little help in the extraction task.

Listing 1 shows an example of HTML source[1] for a publications string. The corresponding rendered webpage is shown in Figure 1. In the HTML source, the text of a single publication string is split into multiple HTML DOM nodes. It is a non-trivial task to identify the text from these nodes that form the same publication string, especially for multi-line publication strings. The HTML font modifiers further complicate the task. For example, Listing 1 uses tag for emphasizing.

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <p><strong>Cleavage furrow
       organization requires
       PIP2-mediated recruitment of
       anillin
6  </strong><br/>Liu, J., Fairn, G.D.,
       Ceccarelli, D.F., Sicheri, F.
       and Wilde A. <br/>
7  Curr. Biol. 2012 22:64-69 &
       nbsp;<a href="http://
       www.sciencedirect.com/science/
       article/pii/S0960982211013224">
       Read</a></p>
8
9  </body>
10 </html>
```

Listing 1: An example of HTML source for a publication string

Cleavage furrow organization requires PIP2-mediated recruitment of anillin

Liu, J., Fairn, G.D., Ceccarelli, D.F., Sicheri, F. and Wilde A.

Curr. Biol. 2012 22:64-69 Read

Figure 1: Rendered webpage for the HTML source in Listing 1

Non-English webpages and non-ASCII encodings: For simplicity, we do not consider webpages that mainly contain non-English characters. As for Unicode text, we use a Package named Unidecode[2] to obtain the ASCII transliterations of Unicode text on webpages.

Newline characters: Our PubSE model works directly on the visible text content of a webpage. In the preprocessing step, we use a special word embedding `N` for the newline character. The word embedding `N` is obtained by averaging the word embedding in GloVe of all the infrequent words (obtained by excluding the 10,000 most frequent words[3] from the GloVe vocabulary). This special treatment of the newline character allows us to use an `IO` tagging scheme, which reduces the learning space. The model is expected to tag the newline character as `I`, if it is within a multi-line publication, and `O` otherwise.

B Detailed Error Analysis

Figure 2 shows six prediction examples given by various models. We analyze them as follows.

Typical error of the line-level model: Examples 1 and 2 show typical errors occurred in line-level model prediction results. As shown in Example 1, the line-level models do not handle multi-line publication strings well. As shown in Example 2, the line-level model mistakenly predicts a presentation and a supervised thesis as publication

* Corresponding author

[1] http://biochemistry.utoronto.ca/person/andrew-wilde/

[2] https://pypi.org/project/Unidecode/

[3] https://github.com/first20hours/google-10000-english

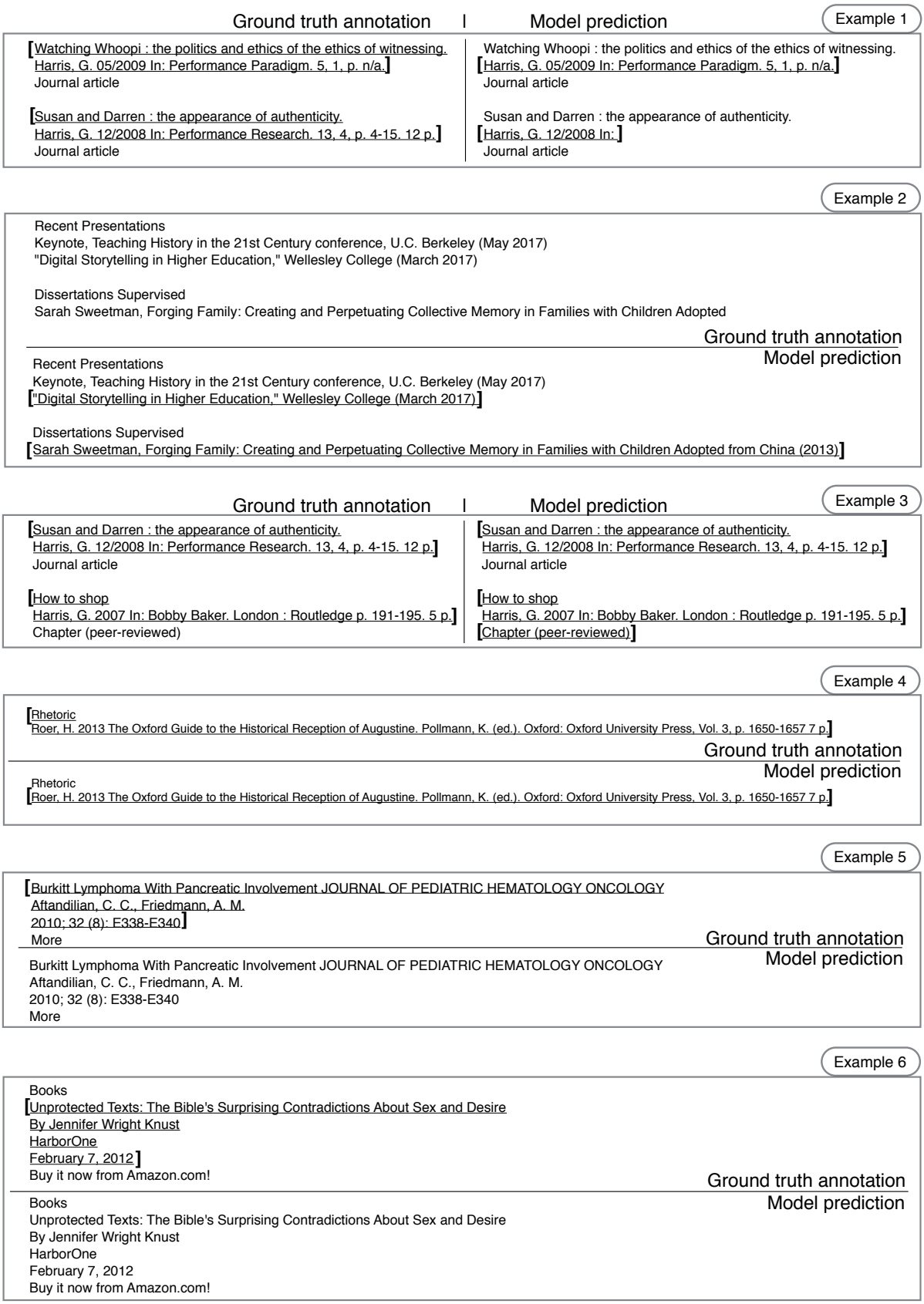

Figure 2: Predicted examples and corresponding ground truth annotations. Underlined tokens are labeled as `I` (publication). "[" and "]" are not part of the text. They are used to highlight the boundary of publication strings.

string, since it fails to capture dependency relationships in different lines.

Typical error of the webpage-level model: Example 3 is given by the webpage-level model. We see that the webpage-level model can make a more accurate prediction for multi-line publications. However, it may make false positive predictions for short lines (e.g., "Chapter (peer-reviewed)"), while the line-level model seldom makes such mistakes. This is the motivation for us to integrate both the line-level and the webpage-level models.

Typical error of the PubSE model: PubSE can avoid most of the errors shown in Examples 1 to 3. Nevertheless, PubSE still makes mistakes in some challenging cases. Examples 4 to 6 show such cases.

In Example 4, PubSE does not recognize that "Rhetoric" is a publication title. A possible explanation is that such a short publication title is less common.

In Examples 5 and 6, PubSE did not recognize both multi-line publication strings. For Example 5, the reason may be the venue name in upper-case letters and the page number with a letter "E". For Example 6, the reason may be the unconventional style to present the book, e.g., an exact date of publication and the word "by", which does not often appear in publication strings. We plan to investigate these problems in the future.

C Choosing Parameter Values: Batch Size of the Webpage-level Model

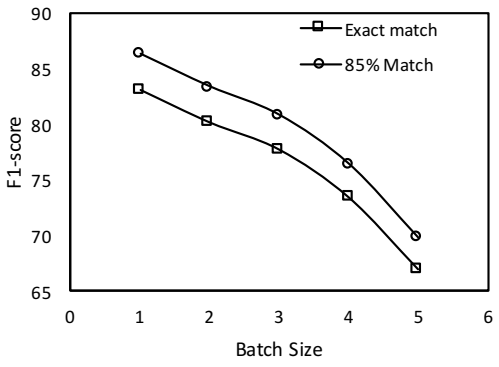

Figure 3: F1-scores of the webpage-level model on the HomePub dataset with different batch sizes

We conduct experiments on a computer with an NVIDIA Tesla K40 GPU which has 12GB GPU memory. Due to the limited GPU memory and the large size of each webpage (732.1 tokens per webpage on average, with a standard deviation of 1583.3), it is only feasible to set the batch size as one. If the batch size is larger than one, there will be an out-of-memory error. To test the effect of using different batch sizes, we conduct experiments using CPU with 64GB RAM memory, on which we are able to set the batch size to be up to five. The experimental results are shown in Figure 3. The prediction accuracy decreases when the batch size increases. This is mainly because the lengths of different webpages vary greatly, leading to high variance within the same mini-batch. Recent work (Masters and Luschi, 2018) has also shown that small mini-batch size helps to increase the prediction accuracy. As a result, we set the batch size of the webpage-level model to one in the PubSE model.

References

Dominic Masters and Carlo Luschi. 2018. Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*.