CS378: Final Project - Paragraph Segmentation code

Victor Wang
vw3795

Abstract

Paragraph segmentation is the task of organizing sentences into paragraphs, and falls within the broader task of text segmentation. The task may be applied to improve readability of a document. I formulate the task as a sequence labeling task and build a sentence-level transformer as a baseline model. I then add coreference-based attention in an attempt to allow the model to capture dependencies that don't make it into the sentence embeddings. Although I find that my model does not improve on the baseline, I discuss likely explanations and propose other strategies.

1 Introduction (5pt)

Paragraph segmentation is the task of organizing a sequence of sentences into paragraphs. Direct applications of this task include suggesting paragraph breaks to a writer and improving readability of a machine-generated transcript. Solving this task may also be useful for downstream tasks such as summarization.

To train a model, I take training data to be a corpus of documents with labeled paragraph breaks, since paragraph structure naturally occurs in human writing. The decision of whether to insert a paragraph break occurs at each sentence, so my model operates at the sentence level, taking as input their embeddings and returning a prediction for each sentence. On the other hand, word-level information is also critical to understanding the relationship between sentences. Thus, my model incorporates word-level information by computing coreference scores between sentences to use as attention weights.

I evaluate my model against a baseline model that ignores word-level information in the sense described above. Unfortunately, my model does not outperform the baseline model. This result may suggest that coreference is not sufficient for capturing the word-level information relevant to segmenting paragraphs.

2 Task / Datasets (20pt)

The task can be stated as a sequence labeling task. Given a sequence of sentences $s_1, \ldots, s_N$ that form a document, we predict their hidden labels $y_1, \ldots, y_N$, where y_i denotes whether s_i begins a paragraph.

The dataset I use is the Wikipedia dataset on HuggingFace [2]. This dataset is suitable for my task because it consists of a large number of documents with labeled paragraph breaks. The dataset is split into several subsets, including English, with 6,458,670 articles, and Simple English, with 205,328 articles. Due to computational resource constraints and only being able to load whole subsets, I use articles from Simple English. I preprocess the data by only including paragraphs that are long enough (at least 2 sentences) and documents that are long enough (at least 9 sentences and 3 paragraphs). After preprocessing, I have a training set of 9,900 articles, a development set of 100 articles, and a test set of 100 articles.

3 Model (25pt)

Figure 1 sketches my model architecture. The sentences are embedded using a pretrained SimCSE model [3]. To capture word-level information, I run coreference analysis on the document to compute a score, explained in the following paragraphs, between every pair of sentences. The coreference scores are used as attention weights in an "engineered" transformer block (purple in Figure 1), which is identical to a transformer block with single-headed attention [8] except that the attention weights are not computed with parameters W^Q and W^K. The result is passed through a transformer encoder followed by a linear layer and

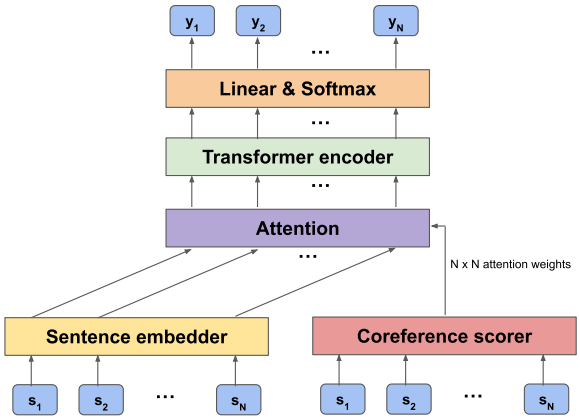

Figure 1: Model architecture.

softmax to make the predictions. The prediction classes are paragraph start, paragraph end, and neither. Since the task is evaluated as a binary task, at inference, I predict a paragraph start if and only if the model's prediction is paragraph start.

Analyzing coreference across sentences can highlight context features that would fail to be encoded in the sentence embeddings computed on independent sentences. For example, in the sentences

> Owls see better at night. This is because they have pentagon-shaped eyes.

it would be unnatural to insert a paragraph break, whereas the same cannot be said about the sentences

> Owls see better at night. Mammals have pentagon-shaped eyes.

However, it is unlikely that an embedding of the isolated second sentence in each case would allow the transformer to behave discriminately, especially because SimCSE models are trained on natural language inference data, which generally feature self-contained sentences. This motivates a mechanism for applying attention between sentences based on the extent to which they corefer. To conceptualize the role of such an attention mechanism, consider a sentence to have an incomplete embedding if it corefers to an entity in another sentence. To fill in the gap, the sentence adds some of the other sentence's value to its own embedding.

I run coreference analysis with the open-source library neuralcoref[1]. The result is a set of clusters C, each representing an entity through the set of mentions m to the entity. The matrix cs of coreference scores between N sentences is computed as follows.

$cs \leftarrow N \times N$ matrix of 0s
for $c_i \in C$ **do**
  for $(m_{ij}, m_{ik}) \in c_i \times c_i$ **do**
    $cs[m_{ij}.\text{sent\_id}, m_{ik}.\text{sent\_id}] \mathrel{+}= 1/|c_i|$
for $i \leftarrow 1, \ldots N$ **do**
  $cs[i,i] \mathrel{+}= sw[i] \cdot \sum_j cs[i,j]$
  $cs[i] \mathrel{/}= \sum_j cs[i,j]$

where sw represents the self-weight scores of sentences, defined

$$sw[i] = \log\left(1 + \sum_{w_{ij} \in s_i} \frac{1}{\text{tf}(w_{ij})}\right)$$

where $\text{tf}(w_{ij})$ is the number of times word w_{ij} occurs in the document (stop words ignored).

Let's unpack this computation. The score increases for sentences that corefer, and the amount is spread across all coreference instances to a given entity. A sentence should also retain the bulk of its representation for itself, so we allocate that portion based on the total importance of its words, which is modeled with inverse term frequency.

I learn the model with negative likelihood loss and Adam optimization. My model's hyperparameters are the transformer hyperparameters, which I choose to be d_model=768 (to match the sentence embedding size from SimCSE's `sup-simcse-roberta-base` model that I use), n_heads=4, num_layers=3, feedforward layer size=2048. I chose these values by starting with the BERT-base model hyperparameters [1] and then reducing values to speed up training based on limited decrease in performance on a development set.

4 Experiments

4.1 Experimental Setups (5pts)

I choose the baseline model to be my model but without the coreference-based attention (in Figure 1, red and purple blocks removed). I also consider another model in which I run coreference analysis and replace coreference mentions with the entity's "main" mention (provided by neuralcoref) before passing the sentences to the baseline model.

My dataset comes from the Simple English Wikipedia dataset: training set of 9,900 articles,

[1] https://github.com/huggingface/neuralcoref

development set of 100 articles, test set of 100 articles, containing a total of 260,860 sentences. I evaluate the models with the F1 score, where a paragraph break is treated as a positive label.

4.2 Results (15pts)

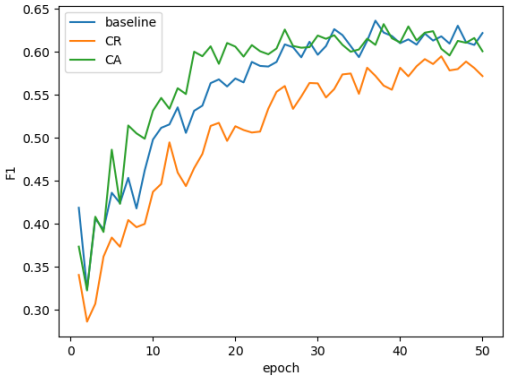

Figure 2: Evaluation on development set.

	Baseline	CR	CA
F1	0.603	0.549	0.613

Table 1: Evaluation on test set.

I refer to my model (which contains coreference-based attention) as CA, and refer to the model with coreference replacement as CR. Figure 2 compares the models' performance during training. Although CA does better than the baseline on during part of training, they end up with very similar performance. Surprisingly, CR does the worst of all. The test set results, shown in Table 1 match the development set results.

The experiments were run on a GTX node on the TACC Maverick2 machine[2]. I ran 50 epochs, each consisting of training on 9,900 examples and evaluating on 100 examples, containing a total of 258,443 sentences. Figure 3 compares the runtime of embedding sentences, coreference analysis, and the epochs. Coreference analysis is by far the most expensive component. This is largely due to not being able to get neuralcoref to use the GPU, unlike the other components. Even if it were run on the GPU, though, it would still be expensive because it is the only component that considers interaction among words in different sentences.

[2]https://docs.tacc.utexas.edu/hpc/maverick2/

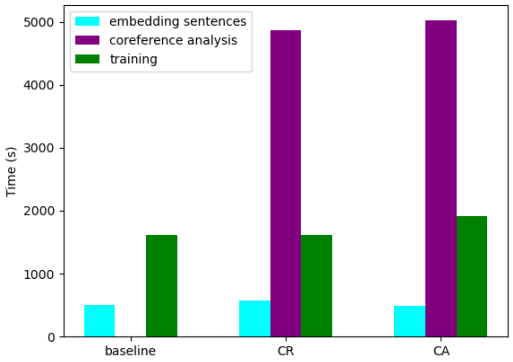

Figure 3: Runtime comparison.

5 Analysis (20pts)

I analyze two examples. Tables 2 and 3 show a development set example and a test set example along with the true labels and model predictions. Figures 4 and 5 show the respective coreference-based attention. In the development set example, the only noted coreference is in sentences 1 and 2. On the other hand, in sentences 3 and 4, there is no coreference relationship between "in some countries" and "in other countries", even though the meaning of "other" is relative to "some". In the test set example, the only noted coreference is between sentences 7, 8, and 9. On the other hand, "a money manager" in sentence 3 is not a coreference to but rather an instantiation of "money managers" in sentence 2. Noting this relationship may have avoided incorrectly predicting a paragraph break. These examples provide evidence that while coreference can capture some useful relationships, it is not nearly broad enough as a phenomenon to capture the majority of the word-level information that determines how separable a pair of sentences are. In fact, this could be seen back in the example "Owls see better at night. This is because they have pentagon-shaped eyes." where "This" does not refer to a coreferenceable entity but rather the entire previous sentence.

One approach to take advantage of word-level information is to simply pass words rather than sentences to the transformer. But since prediction occurs at a sentence-level, some method is needed to aggregate the word-level predictions. Another, more important concern is runtime, because attention scales quadratically with sequence length. Although coreference analysis also looks at words in different sentences, it only considers a small

Sentence	Truth	Baseline	CR	CA
A school uniform is a standard set of clothing students wear when they go to some schools.	1	1	1	1
It might have a particular color of trousers or skirt, plus a matching shirt and perhaps a jacket or necktie, with matching shoes.				
In some countries, like Germany, students can wear anything they like when they go to school.	1			
In other countries, like England, there is usually a standard dress code in school, usually a set of dressing for girls and one for boys.		1		
Boys and girls need to wear school uniforms when they go to school.				
In many countries, such as the United States, some schools require wearing a uniform, and some do not.				
Originally, school uniforms were introduced to hide the social differences between students, but uniforms can also help with safety.	1	1	1	1
Using standard uniforms can also save the money needed to buy extra clothes as fashion to impress other people at school.				
Uniforms are also thought to improve discipline and school spirit.		1		1
However, school uniforms can also help with health and safety by having clothes which have been tested to be safer when worn.	1			
Some fabrics can cause skin rashes in some people, while a uniform can be made of comfortable fabrics.				1
Also, loose-fitting clothes can get caught in machinery or playground equipment, which limits what activities children can do safely.				
There are several types of economic bullying which can be lessened by use of school uniforms.	1	1	1	1
When many students are from families with less money, sometimes students with more money have stood out because they wore newer shoes, where neither shoe was in poor shape.				
In schools where more students are rich, poorer students have been insulted for the old-style or tattered clothes they wore.				

Table 2: Development set example. 1 means the sentence begins a paragraph.

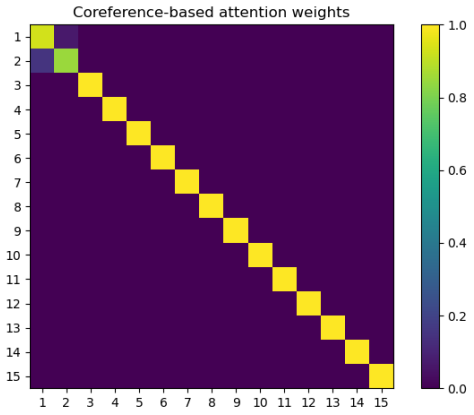

Figure 4: Coreference-based attention for the example in Table 2

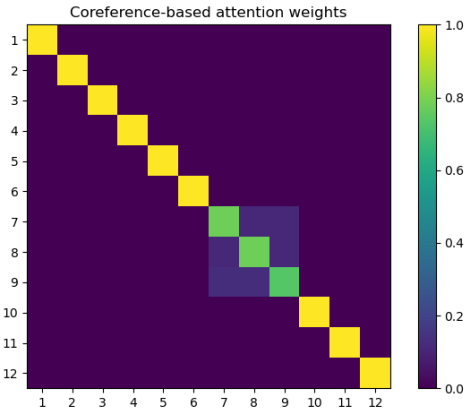

Figure 5: Coreference-based attention for the example in Table 3.

Sentence	Truth	Baseline	CR	CA
A Separately Managed Account or SMA is a type of financial investment account popular with some individual investors.	1	1	1	1
SMAs are offered by financial consultants and brokerage firms, and managed by independent "money managers."				
A money manager is simply a business that manages stock and other securities portfolios or baskets of investments for an investor.				1
SMAs have varying fee structures.				
Common characteristics of SMA programs are: they provide open investment choices; have multiple managers; and, offer a customized investment portfolio created for a client's specific needs.			1	
By customizing the portfolio, investors can limit the investment risk with various strategies, such as using stock options, according to the Wall Street Journal.				
Morningstar Inc., a fund research firm, released research that showed that SMAs beat mutual funds in 2008.	1	1	1	1
Morningstar said that SMAs outperformed mutual funds in 25 of 36 stock and bond market categories that year.				
In addition, Morningstar said that from 2006 to 2008, SMAs surpassed mutual funds in 22 of 26 categories.				
Financial experts point out that past performance of any investment is not a predictor of future performance. So, investors should talk with a licensed financial adviser and carefully review all information available before making any investment decision.	1	1	1	1
In addition, before making an investment, investors should check with the appropriate licensing and regulatory authorities to ensure that the firm(s) offering SMAs is (are) properly licensed and in good standing.				

Table 3: Test set example. 1 means the sentence begins a paragraph.

window, under the assumption that coreference occurs locally. By extending this assumption, it may be promising to use word-level attention but only between pairs of adjacent sentences. In this approach, it would be useful to learn a segment A/B embedding like in BERT [1].

Evaluating a model for paragraph segmentation on pre-labeled data is noisy because there are generally many acceptable ways to segment a document into paragraphs. Thus, to place in scope how good the model performances in this report are, we must compare them to inter-annotator agreement on the same task. Another, more expensive evaluation method would be to have annotators grade how good the paragraph breaks predicted by a model are.

6 Future Work

One direction of future work is designing the model output format. In the models benchmarked in this report, for each sentence the model outputs a probability distribution over three classes: paragraph start, paragraph end, and neither. I choose this over the binary classification inherent to the task because I believe that there are properties distinct to a paragraph end, similar to how there are properties distinct to a paragraph start, that the model can learn to distinguish if we provide it in the training signal, whereas the training signal will be noisier if we conflate the labels for paragraph end and paragraph middle. Another advantage is that when deciding whether to add a paragraph break, we can consider the probability that the sentence is a paragraph start as well as the probability that the previous sentence is a paragraph end. (In the current models, I make the predictions independently.)

There is more room for design of the model output. We could even move away from classification and represent a sentence's position in a paragraph with a real number in $[0, 1]$ to enrich the training signal. If we continue with the hypothesis that the paragraph start and end are the most distinguishable positions, we can choose a function that transforms position to flatten out the middle positions, such as x^{2k+1} for $k \in \mathbb{N}$ as shown in Figure 6. (Note that for $k \to \infty$, we are back to the ternary classification.) To inference using the model output, we need an algorithm or heuristic to choose a set of paragraph breaks that minimizes the average difference between the po-

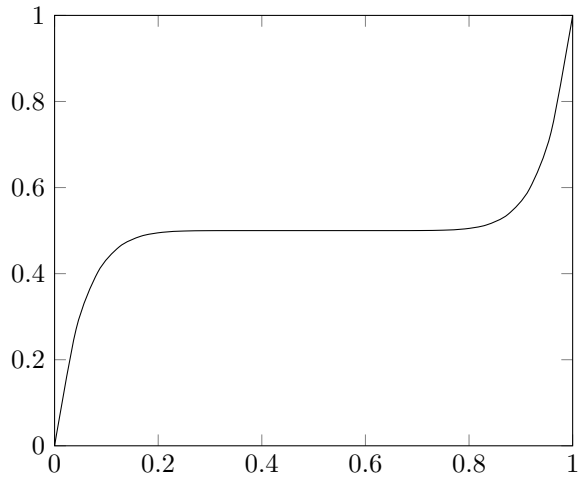

Figure 6: The function $f(x) = \frac{1}{2}((2x-1)^9+1)$ (a transformed version of x^9) to scale in-paragraph sentence position.

sition predicted by the model for a sentence and the position we assign it. It is worth noting that this model output design requires the assumption that a monotonic relationship can be learned between position labels in the classification setting and this real number position.

7 Related Work (5pt)

The paragraph segmentation task addressed in this report is a specific case of text segmentation, which seeks to segment text based on topic or some semantic property. A large dataset that has been created for this task is built from Wikipedia, where the table of contents is used to identify topic separators [5]. Two notable methods have been (1) TopicTiling, in which a topic distribution is computed for documents and words and then a coherence score is computed between each pair of adjacent sentences to create divides where there is low coherence [7], and (2) GraphSeg, a graph algorithm that finds maximal cliques in a graph where nodes are sentences and edges join similar sentences [4]. I believe that the paragraph segmentation task is more difficult than the broader text segmentation task because the former is more subjective and sensitive to sub-sentence phrasing.

The method I use as the baseline model largely draws from the extractive summarization model BertSum [6]. Their model passes sentence embeddings through a transformer to make the binary predictions of whether to include each sentence in the summary.

8 Conclusion (5pt)

In this report I tackled the paragraph segmentation task, which can be useful for things like writing assistance. I attempted to add word-level information through coreference analysis to an otherwise sentence-level model. I found that my model does not outperform the baseline model. Examining coreference scores for some examples showed that a lot of word-level information fails to be captured by coreference.

References

[1] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[2] Wikimedia Foundation. *Wikimedia Downloads*. URL: https://dumps.wikimedia.org.

[3] Tianyu Gao, Xingcheng Yao, and Danqi Chen. "SimCSE: Simple Contrastive Learning of Sentence Embeddings". In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2021.

[4] Goran Glavas, Federico Nanni, and Simone Paolo Ponzetto. "Unsupervised Text Segmentation Using Semantic Relatedness Graphs". In: *International Workshop on Semantic Evaluation*. 2016.

[5] Omri Koshorek et al. *Text Segmentation as a Supervised Learning Task*. 2018. arXiv: 1803.09337 [cs.CL].

[6] Yang Liu. *Fine-tune BERT for Extractive Summarization*. 2019. arXiv: 1903.10318 [cs.CL].

[7] Martin Riedl and Chris Biemann. "TopicTiling: A Text Segmentation Algorithm based on LDA". In: *Proceedings of the Student Research Workshop of the 50th Meeting of the Association for Computational Linguistics*. Jeju, Republic of Korea, 2012, pp. 37–42. URL: http://www.aclweb.org/anthology/W12-3307.

[8] Ashish Vaswani et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].