Recovering Lexically and Semantically Reused Texts

Ansel MacLaughlin[*], **Shaobin Xu**[*], **David A. Smith**
Khoury College of Computer Science
Northeastern University
{ansel, shaobinx, dasmith}@ccs.neu.edu

Abstract

Writers often repurpose material from existing texts when composing new documents. Because most documents have more than one source, we cannot trace these connections using only models of document-level similarity. Instead, this paper considers methods for local text reuse detection (LTRD), detecting *localized* regions of lexically or semantically similar text embedded in otherwise unrelated material. In extensive experiments, we study the relative performance of four classes of neural and bag-of-words models on three LTRD tasks – detecting plagiarism, modeling journalists' use of press releases, and identifying scientists' citation of earlier papers. We conduct evaluations on three existing datasets and a new, publicly-available citation localization dataset. Our findings shed light on a number of previously-unexplored questions in the study of LTRD, including the importance of incorporating document-level context for predictions, the applicability of of-the-shelf neural models pretrained on "general" semantic textual similarity tasks such as paraphrase detection, and the trade-offs between more efficient bag-of-words and feature-based neural models and slower pairwise neural models.

1 Introduction

When composing documents in many genres—from news reports, to scientific papers, to political speeches—authors obtain ideas and inspiration from source documents and present them in the form of direct copies, quotations, summaries, or paraphrases. In the simplest case, e.g. in congressional bills, writers include text from earlier versions of the same document along with new material (Wilkerson et al., 2015). In news media, journalists often paraphrase or quote speeches, press releases, and interviews (Niculae et al., 2015; Tan et al., 2016). In academia, citations of papers usually appear along with summaries of their contributions (Qazvinian and Radev, 2010). These are instances of lexical and semantic **local text reuse**, where both source and target documents contain lexically or semantically similar passages, surrounded by text that is unrelated or dissimilar. Often, reused text is presented without explicit links or citations, making it hard to track information flow.

While many state-of-the-art (SoTA) NLP architectures have been trained on the closely-related tasks of document- and sentence-pair similarity detection (Reimers and Gurevych, 2019) and ad-hoc retrieval (Dai and Callan, 2019), prior methods for local text-reuse detection (LTRD) are mostly limited to lexical matching (Lee, 2007; Clough et al., 2002; Leskovec et al., 2009; Wilkerson et al., 2015; Smith et al., 2014) with some dictionary expansion (Moritz et al., 2016). To our knowledge, only Zhou et al. (2020) has applied neural models to this problem, proposing hierarchical neural models that use a cross-document attention mechanism to model local similarities between two candidate documents.

In this paper, we conduct a large-scale evaluation of several lexical overlap and SoTA neural models for LTRD. Among the neural models, we benchmark not only the hierarchical neural models proposed by Zhou et al. (2020), but also study the effectiveness of three classes of models not yet applied to LTRD: **1)** BERT-based (Devlin et al., 2019) passage encoders trained on generic paraphrase, semantic textual similarity, and IR data (Reimers and Gurevych, 2019); **2)** feature-based BERT models with direct sentence-level supervision; and **3)** fine-tuned BERT-based models for sequence-pair tasks.

We conduct evaluations on four datasets, including **1)** PAN and S2ORC (Zhou et al., 2020), benchmark LTRD datasets for plagiarism detection and

[*] Equal contribution.

Proceedings of the 10th Conference on Lexical and Computational Semantics, pages 52–66
August 5–6, 2021, Bangkok, Thailand (online)

citation localization; **2)** Pr2News (MacLaughlin et al., 2020), a dataset of text reuse in news articles labeled with a mix of expert, non-expert, and heuristic annotation; **3)** ARC-Sim, a new, publicly available[1] citation localization dataset created using citation links in the ACL ARC (Bird et al., 2008).

Our experiments address a number of previously-unexplored questions in the study of LTRD, including **1)** the impact of training on weakly-supervised data on model accuracy; **2)** the effectiveness of SoTA neural models trained on "general" semantic similarity data for LTRD tasks; **3)** the importance of incorporating document-level context; **4)** the effects of domain-adaptive pretraining (Gururangan et al., 2020) on the accuracy of fine-tuned BERT models; and **5)** the trade-offs between more efficient lexical overlap and feature-based neural models and slower pairwise neural models.

2 Related Work

LTRD methods have been applied in many domains, including tracking short "memes" in news and social media (Leskovec et al., 2009), tracing specific policy language embedded in proposed legislation (Wilkerson et al., 2015; Funk and Mullen, 2018), studying reuse of scripture in historical and theological writings (Lee, 2007; Moritz et al., 2016), tracing information propagation in news and social media (Tan et al., 2016; Clough et al., 2002; MacLaughlin et al., 2020), and detecting plagiarism on the web (Potthast et al., 2013; Sánchez-Pérez et al., 2014; Vani and Gupta, 2017). Most applications, however, use only *lexical* overlap and alignment methods to detect reuse, sometimes with lemmatization and dictionary curation.

Our work builds on the recent efforts of Zhou et al. (2020), who demonstrate the efficacy of hierarchical neural models in detecting instances of non-literal reuse where authors paraphrase, summarize, and heavily edit source content. However, as discussed in §1, we conduct a much larger set of experiments beyond those of Zhou et al. (2020). In addition to the hierarchical neural models with document-level supervision proposed by Zhou et al. (2020), we evaluate four sets of models: lexical overlap models, SoTA neural models trained for general paraphrase detection, hierarchical neural models with sentence-level supervision, and fine-tuned sequence-pair BERT models. Further, in addition to evaluating models on the benchmark LTRD datasets introduced by Zhou et al. (2020), we conduct experiments on two more challenging datasets: ARC-Sim, a new citation localization dataset with hard negative examples, and Pr2News (MacLaughlin et al., 2020), a dataset of text reuse in science news articles with heuristically-labeled training data.

Also related to our work is research studying sentence-pair problems, e.g. paraphrase detection (PD) (Dolan and Brockett, 2005), semantic textual similarity (STS) (Cer et al., 2017) and textual entailment, (Bowman et al., 2015), and document-ranking problems, e.g. ad-hoc retrieval (Croft et al., 2009). In fact, it is trivial to adapt existing approaches to sentence-pair and document ranking problems to LTRD. As discussed in §3, we cast LTRD as sentence classification and ranking, identifying which sentences in a target text are lexically or semantically reused from some portion of the source. Thus, in order to adapt sentence-pair models to this task, we simply compute scores for all pairs of (source sentence, target sentence), and use some function (e.g. *max*) to aggregate the scores for each target sentence. Similarly, one can adapt existing ad-hoc retrieval approaches by treating each target sentence as a query and computing a score with the corresponding source. These approaches, however, may suffer from a lack of contextualization and/or efficiency issues. Sentence-pair models that encode each source and target sentence separately, while efficient, might miss important contextualizing information in surrounding sentences. Similarly, neural IR models that process each target sentence as a separate query do not contextualize target sentences and also require a computationally-expensive forward pass for each query. We study the importance and impact of these limitations in our work, testing the effectiveness of multiple SoTA BERT-based architectures for sequence-pair similarity and ranking.

3 Problem Definition

Following Zhou et al. (2020), we define LTRD as two tasks: **document-to-document (D2D) alignment** and **sentence-to-document (S2D) alignment**. In D2D, for a given pair of documents (source document $\mathbf{S}$, target document $\mathbf{T}$), we aim to predict whether $\mathbf{T}$ reuses content from $\mathbf{S}$. Thus, each pair has a corresponding binary label of 1 if $\mathbf{T}$ reuses content, else 0. Note, this is different than

[1] https://github.com/maclaughlin/ARC-Sim

evaluating the similarity of the two documents as a whole, since, in this setting, only a small portion of $\mathbf{T}$ is adapted from $\mathbf{S}$, and most of it is possibly unrelated. In S2D, given an ($\mathbf{S}$, $\mathbf{T}$) pair, we aim to predict *which* specific sentences $\mathbf{t_i} \in \mathbf{T}$ contain reused $\mathbf{S}$ content. Thus, each pair has n corresponding labels, one label for each sentence $\mathbf{t_i} \in \mathbf{T}$.[2]

4 Models

We benchmark four classes of models on this task:

4.1 Lexical Overlap Models

We evaluate two unsupervised metrics:

- **TF-IDF Cosine Similarity**: Simple word overlap metrics are commonly-used baselines to measure the similarity between two passages for PD (Dolan and Brockett, 2005), STS (Reimers and Gurevych, 2019), document retrieval (Croft et al., 2009), and LTRD (Tan et al., 2016; Lee, 2007; Clough et al., 2002).
- **Rouge** (Lin, 2004): Since authors of derived documents often paraphrase and summarize source content, we evaluate Rouge, a popular summarization evaluation metric. We evaluate Rouge-{1, 2, L}, selecting the best configuration for each dataset using validation data.

We compute two versions of each metric: single-pair (sp) and all-pairs (ap). In sp, for a given document pair ($\mathbf{S}$, $\mathbf{T}$), we compute a score for each sentence $\mathbf{t_i} \in \mathbf{T}$ by computing its similarity to the *entire* $\mathbf{S}$. In ap, we compute a score for each sentence $\mathbf{t_i} \in \mathbf{T}$ by computing its similarity to *each sentence* $\mathbf{s_i} \in \mathbf{S}$, then selecting the maximum score over all $\mathbf{s_i}$. These scores are then thresholded to make binary predictions. For the D2D task, we predict $\mathbf{T}$ as positive if it contains at least one positively predicted sentence. For the S2D task, we evaluate the predicted score for each $\mathbf{t_i} \in \mathbf{T}$.

4.2 Pretrained Sentence-BERT Encoders

We evaluate Sentence-BERT (SBERT) (Reimers and Gurevych, 2019), a SoTA pretrained passage encoder for semantic-relatedness tasks. SBERT models are trained by 1) adding pooling (e.g. mean pooling) to the output of BERT; 2) training on pairs or triplets of passages to learn semantically meaningful passage representations; 3) at test time, computing the similarity between two passages as the cosine similarity between their pooled representations. We evaluate three SBERTs trained for different tasks:

- **Semantic Textual Similarity (STS)**: Roberta$_{LARGE}$ (Liu et al., 2019) trained on SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018) then fine-tuned on the STS-B (Cer et al., 2017) train set.
- **Paraphrase Detection (PD)**: distilled Roberta$_{BASE}$ (Sanh et al., 2019) fine-tuned on a large-scale paraphrase detection corpus.
- **Information Retrieval (IR)**: distilled Roberta$_{BASE}$ (Sanh et al., 2019) fine-tuned on MS MARCO (Campos et al., 2016).

Note, these pretrained SBERT models are *not* trained for LTRD. Instead, they are trained on large-scale datasets for other related tasks (PD, STS, IR). These experiments thus evaluate how well off-the-shelf tools generalize to a new task and domain.

Just as the lexical models, we evaluate in sp and ap settings. Following Reimers and Gurevych (2019), we embed each source document, source sentence, and target sentence separately, then compute cosine similarity for each pair.

4.3 Hierarchical Neural Models (HNM)

We also benchmark three HNM. Similar to SBERT (§4.2), HNM operate on frozen embeddings (Peters et al., 2019) which are computationally efficient since they only need to be calculated once (i.e. only one BERT forward pass for each source or target sentence). Unlike SBERT, however, HNM also have task-specific model architectures that learn to contextualize and align sentences.

BERT-HAN (shallow) (Zhou et al., 2020): this model mean pools frozen BERT embeddings to generate sentence representations, then uses a hierarchical attention network (HAN) (Yang et al., 2016) to add document-level context and a cross-document attention (CDA) mechanism to align passages across documents. See Zhou et al. (2020).

At training time, BERT-HAN only calculates loss at the document-pair level, i.e. D2D classification. There is no sentence-level supervision (S2D). At inference, two sets of predictions are output: 1) the D2D prediction, as during training; 2) the intermediate hidden representations of the sentences

[2] We could also study the sentence-to-sentence problem, learning to identify which source sentence(s) contain the content reused in a given target sentence, if any. However, as noted by Zhou et al. (2020), no datasets exist yet which contain such fine-grained annotation.

$\mathbf{t_i} \in \mathbf{T}$ are extracted, then ranked by their similarity to the final hidden representation of the entire source document $\mathbf{S}$.

GRU-HAN (deep) (Zhou et al., 2020): this model mirrors BERT-HAN, except with GloVe (Pennington et al., 2014) embeddings and a HAN with CDA at both the word and sentence level. It follows the same training and testing regime.

BCL-CDA: We adapt the BCL model from MacLaughlin et al. (2020) (originally designed for the task of intrinsic source attribution on Pr2News) for LTRD by adding a final CDA layer (Zhou et al., 2020). After generating contextualized representations of each source and target sentence with BCL, a CDA layer computes an attention-weighted representation of each target sentence, weighted by its similarity to the source sentences. The CDA-weighted and original target sentence representations are then concatenated and fed into a final layer for prediction.

At training time, BCL-CDA is supervised with target sentence labels. At testing time, it makes target sentence-level predictions (S2D) just as in training. We make a D2D prediction for each ($\mathbf{S}$, $\mathbf{T}$) pair by taking the max over its sentence-level predictions. See Appendix C for full model details.

4.4 Fine-tuned BERT-based Models

Finally, we evaluate fine-tuned BERT-based models for sequence pair classification. Unlike the other three classes of models described above, features for these fine-tuned models cannot be precomputed. Instead, at test time, a separate forward pass is required for each ($\mathbf{S}$, $\mathbf{T}$) or ($\mathbf{S}$, $\mathbf{t_i}$) pair. Thus, though these models might achieve better performance than feature-based alternatives (Peters et al., 2019), it may be unfeasible to test them on large collections where many pairwise computations would be required.

Sequence Pair Models: We fine-tune Roberta$_{Base}$ (Liu et al., 2019) using the standard setup for sequence-pair tasks such as PD, STS, and IR (Devlin et al., 2019; Akkalyoncu Yilmaz et al., 2019). We create an input example for each (source document $\mathbf{S}$, target sentence $\mathbf{t_i}$) pair:

$$[\texttt{CLS}] < \texttt{s}_1, ..., \texttt{s}_n > [\texttt{SEP}]\ \texttt{t}_i\ [\texttt{SEP}]$$

where $< \texttt{s}_1, ..., \texttt{s}_n >$ contains the source document, split into sentences, with each sentence separated by a special [SSS] token ("source sentence start") and $\texttt{t}_i$ is a single target sentence. We feed the [CLS] representation into a final layer to make a prediction for $\texttt{t}_i$. Thus, making a prediction for an entire ($\mathbf{S}$, $\mathbf{T}$) document pair requires n forward passes, one for each $\mathbf{t_i} \in \mathbf{T}$.

Table 1: Dataset statistics: the total number of (source $\mathbf{S}$, target $\mathbf{T}$) example pairs, the average # of sentences and words in each $\mathbf{S}$ and $\mathbf{T}$, and the average # of positively labeled $\mathbf{T}$ sentences in each *positive* ($\mathbf{S}$, $\mathbf{T}$) pair. For Pr2News, we report the average # of $\mathbf{T}$ sentences with label > 0 in the human-labeled val and test sets.

		Avg. Source		Avg. Target		
Dataset	# Examples	# Sents	# Words	# Sents	# Words	# Positive
PAN	18,903	23.7	527.6	22.4	538.5	14.0
S2ORC	188,311	7.2	190.7	11.5	335.8	1.2
Pr2News	64,779	35.3	934.7	30.1	761.3	8.7
ARC-Sim	105,381	5.0	126.6	16.7	400.8	1.2

Domain-adapted Sequence Pair Models: As shown by Gururangan et al. (2020), further pretraining BERT-based models on in-domain text improves performance on a variety of tasks. We explore the effects of DAPT for LTRD, testing Roberta models domain-adapted on either biomedical publications, computer science publications or news data. We fine-tune these models as above.

Sequential Sequence Pair Models: Since the fine-tuned models discussed above operate on a single $\mathbf{t_i}$ at a time, they cannot leverage information from the surrounding target context. Following the success of BERT-based models for sequential sentence classification (Cohan et al., 2019), we construct new input examples containing the full source and target documents, split into sentences:

$$[\texttt{CLS}] < \texttt{s}_1, ..., \texttt{s}_n > [\texttt{SEP}] < \texttt{t}_1, ..., \texttt{t}_n > [\texttt{SEP}]$$

Again, $< \texttt{s}_1, ..., \texttt{s}_n >$ contains the source sentences. Similarly, $< \texttt{t}_1, ..., \texttt{t}_n >$ contains the target sentences, with each separated by a special [TSS] token ("target sentence start"). We feed the final [TSS] representations into a multi-layer feed-forward network to make a prediction for each target sentence.

Each pair is labeled with all corresponding target sentence labels. Since many pairs exceed Roberta's 512 Wordpiece length limit, we use Longformer$_{Base}$ (Beltagy et al., 2020), a Roberta-based model with an adapted attention pattern to handle up to 4,096 tokens. We put global attention on the [SSS] and [TSS] tokens to allow the model to capture cross-document sentence similarity.

5 Datasets

We benchmark the proposed models on four different datasets (Table 1). See Appendix A for further dataset stastics and preprocessing details.

5.1 PAN (Zhou et al., 2020)

PAN contains pairs of ($\mathbf{S}$, $\mathbf{T}$) web documents where $\mathbf{T}$ has potentially plagiarized $\mathbf{S}$. Positive pairs contain synthetic plagiarism, generated by methods such as back-translation (Potthast et al., 2013). Negative examples are created by replacing $\mathbf{S}$ with another, unplagiarized source text, $\tilde{\mathbf{S}}$, sampled from the corpus. D2D labels are binary: plagiarized or not. The S2D labels for $\mathbf{t_i} \in \mathbf{T}$ are 1 if $\mathbf{t_i}$ plagiarizes $\mathbf{S}$, else 0 (labels in negative pairs are 0).

5.2 Pr2News (MacLaughlin et al., 2020)

Pr2News contains pairs of (press release $\mathbf{S}$, science news article $\mathbf{T}$), where each $\mathbf{T}$ has reused content from $\mathbf{S}$. There are three aspects of this dataset which are unlike the others we study: **1)** All ($\mathbf{S}$, $\mathbf{T}$) pairs are positive and contain reuse. Thus, we only evaluate the S2D task. **2)** While the val and test sets are human-annotated, the ($\mathbf{S}$, $\mathbf{T}$) pairs in the training set are labeled using a heuristic (TF-IDF cosine similarity). Though there has been some success training neural models on scores generated by word-overlap heuristics for the problems of document retrieval (Dehghani et al., 2017) and source attribution (MacLaughlin et al., 2020), applications of weakly-supervised models have not yet been studied on human-labeled LTRD test sets. **3)** Target sentences, $\mathbf{t_i} \in \mathbf{T}$, in the val and test sets are labeled on a 0-3 ordinal scale, ranging from no reuse (0) to near or exact duplication (3).

5.3 S2ORC (Zhou et al., 2020)

S2ORC is a citation localization dataset, containing (abstract $\mathbf{S}$, paper section $\mathbf{T}$) pairs. Citation localization consists of identifying which $\mathbf{t_i} \in \mathbf{T}$, if any, cite the source. All citation marks are removed from the texts, so models can only make predictions by comparing the language of $\mathbf{S}$ and $\mathbf{T}$, not just simply identify citation marks. Positive examples are created by sampling scientific papers from the broader S2ORC corpus (Lo et al., 2020), finding sections in those papers that contain citation(s) to another paper in the corpus, and pairing together the (cited source abstract $\mathbf{S}$, citing section $\mathbf{T}$). Negative pairs are created by pairing $\mathbf{T}$ with $\tilde{\mathbf{S}}$, the abstract of a paper it does *not* cite. The D2D labels are 0 for negative pairs, 1 for positive. The S2D labels for $\mathbf{t_i} \in \mathbf{T}$ are 1 if $\mathbf{t_i}$ contains a citation of $\mathbf{S}$, else 0. S2D labels for negative pairs are all 0.

The design of this dataset follows the assumption that the citing sentence(s) in $\mathbf{T}$ often paraphrase or summarize some portion of the cited paper, which is, in turn, summarized by its abstract $\mathbf{S}$. This assumption, however, may be incorrect if the citing sentence is a poor summary of the cited paper (Abu-Jbara and Radev, 2012) or it refers to content in the cited paper which is not included in the abstract. Nevertheless, this assumption allows for easy creation of large-scale, real-world LTRD datasets. This is in contrast to Pr2News, which is substantially smaller due its reliance on human-annotated val and test labels, and PAN, which uses automatic methods to generate synthetic examples. We discuss the trade-offs of using citation marks to generate LTRD datasets in §5.4.

5.4 ARC-Sim

Motivated by the design of S2ORC, we propose a new citation localization dataset[3] built on the ACL Anthology Conference Corpus (ARC) (Bird et al., 2008). Just as S2ORC, we construct our dataset using citations links between papers. Thus, we first break up each ARC paper by section, then use ParsCit (Councill et al., 2008) to find all sections that cite another paper in ARC. Positive examples are pairs (abstract $\mathbf{S}$, paper section $\mathbf{T}$) where $\mathbf{S}$ is cited by at least one $\mathbf{t_i} \in \mathbf{T}$. Using this method we generate 61,131 positive ($\mathbf{S}$, $\mathbf{T}$) pairs. Most (88%) $\mathbf{T}$ contain only one positive sentence.

To create negative examples, we pair each $\mathbf{S}$ from the positive samples with a new section, $\tilde{\mathbf{T}}$, that does not cite it. Importantly, $\tilde{\mathbf{T}}$ is sampled from the *same* target paper as the original $\mathbf{T}$. This generates 44,250 negative pairs.[4] We argue that these negative samples method will be both more *difficult* and *realistic* than those in S2ORC. In S2ORC, negatives are generated by sampling a new source $\tilde{\mathbf{S}}$ to pair with $\mathbf{T}$. However, due to the large scale of the corpus, $\tilde{\mathbf{S}}$ and $\mathbf{T}$ are often completely unrelated (e.g. Bio vs. CS). These examples, therefore, are trivial and can be easily classified using simple lexical overlap. In ARC-Sim, however, negatives are generated by sampling a new section $\tilde{\mathbf{T}}$ from the *same* paper as $\mathbf{T}$. We hypothesize that differentiating between these positive and negative examples will **1)** be more difficult since $\tilde{\mathbf{T}}$ is likely still topically related to $\mathbf{S}$ and may contain some spurious lexical or semantic overlap; **2)** be more indicative of real-world performance, since real users may

[3] Available for download here.

[4] We sample 1 negative pair per (source abstract, target paper), so target papers that cite the source in more than 1 section will have more positive examples than negative.

need to identify which specific sections in a full target paper reuse content from the source. Further preprocessing and dataset split information is detailed in Appendix A. We use the same labeling scheme as S2ORC.

With dataset creation complete, we sample a set of 50 positive pairs from the val set to analyze in depth. Three expert annotators (authors of this paper) perform the LTRD task, predicting which $\mathbf{t_i} \in \mathbf{T}$ reuse content from $\mathbf{S}$. Five pairs are marked by all annotators (Fleiss' Kappa: 0.83). The remaining 45 are split into 15 per annotator. Overall, we find that annotators mark more sentences as reused (avg. 1.6 sents / target) than the true citation labels (1.3 / target). This is reasonable since $\mathbf{T}$ often only cites $\mathbf{S}$ once, even if it discusses $\mathbf{S}$ in multiple sentences (Qazvinian and Radev, 2010). These false negatives are one disadvantage of using citation marks as supervision. Further, we find that annotators and ground truth often, but not always, agree – annotators identify at least one true citing sentence in 72% of pairs. This difference is mostly due to **1)** citing sentences that discuss source content not described in the source abstract; **2)** OCR errors that can make text hard to read. On the whole, we find that ACL-Sim is a useful LTRD dataset, but there are clear avenues for improvement, such as manually annotating reused sentences without citation marks and improving OCR.

6 Evaluation Settings & Metrics

D2D Metrics: We evaluate the D2D task as $(\mathbf{S}, \mathbf{T})$ pair classification using F1 score. A positive label indicates that $\mathbf{T}$ reuses content from $\mathbf{S}$. A negative label indicates no text reuse. There is no D2D task for Pr2News since all examples are positive.

S2D Metrics: We evaluate S2D in two settings: *corpus level* (i.e. evaluating *all* target sentences from all pairs at once), and *document level* (i.e. evaluating the sentences in each target document w.r.t each other, then averaging scores across documents). The metrics for each setting depend on the dataset. At the corpus level, we evaluate binary-label datasets (PAN, S2ORC, ARC-Sim) with sentence-level F1 and ordinal-label datasets (Pr2News: 0-3 scale), with spearman's correlation (ρ) and NDCG@N (where N is the number of target sentences in the test set). At the document level, we evaluate binary-label datasets with mean average precision (MAP) and top-k accuracy (Acc@k), defined as the proportion of test examples where a positively-labeled sentence in $\mathbf{T}$ is ranked in the top k by the model. We evaluate ordinal-label datasets with NDCG@{1,3,5}. Note, in order for these document-level metrics to be meaningful, $\mathbf{T}$ must contain at least one positive sentence. Thus, our document-level evaluations are *only* calculated on the positive $(\mathbf{S}, \mathbf{T})$ pairs in each dataset.[5] Since Pr2News only contains positive examples, we use the full test set for all evaluations.

BERT-HAN & GRU-HAN: Since both HAN models are trained on document-level, not sentence-level, labels, we cannot train them on Pr2News, where all document-level labels are positive. Thus, we skip evaluating the HAN models on this dataset.

Domain-adapted RoBERTa Models: We evaluate three DAPT models: 1) Biomed-DAPT for S2ORC and Pr2News since they contain biomedical texts, 2) News-DAPT for Pr2News since the target documents are news articles, 3) CS-DAPT model for S2ORC and ARC-Sim since they contain CS papers.[6] We do not apply DAPT to PAN since no models are adapted to a similar domain.

7 Results & Discussion

As seen in Tables 2 & 3, BERT-based models fine-tuned on LTRD data perform the best in general, outperforming lexical overlap, SBERT, and HNM. Overall, models achieve their best performances on PAN. We suspect that this is because many positive ($\mathbf{S}$, $\mathbf{T}$) pairs are easy, containing many plagiarized passages with high lexical overlap, and since many negative ($\tilde{\mathbf{S}}$, $\mathbf{T}$) pairs are topically unrelated and share little lexical or semantic overlap. On the other end of the spectrum is ARC-Sim, where models score relatively poorly. We hypothesize that this is because most $\mathbf{T}$ only contain one citing sentence and since, as discussed in §5.4, we focus on selecting hard negative target texts, $\tilde{\mathbf{T}}$, sampled from the same document as the original $\mathbf{T}$.

[5] We confirmed that Zhou et al. (2020) calculate their document-level metrics, MRR, P@5 and P@10, across *all* $(\mathbf{S}, \mathbf{T})$ pairs. For the negative pairs, they give models full credit on the S2D task if their corresponding D2D prediction is correct. We argue that this is not indicative of model performance, and thus conduct our document-level evaluations on *only positive* pairs.

[6] CS- and Biomed-DAPT models are adapted on an internal version of the S2ORC corpus (Lo et al., 2020). Since the S2ORC LTRD dataset is randomly sampled from that same corpus, it is possible that the DAPT models are pretrained on some portion of the S2ORC LTRD test set. We do not believe this overlap exists for any other (DAPT, LTRD dataset) pairs.

Table 2: D2D and S2D results on PAN and S2ORC.

Setting	Model	PAN						S2ORC					
		D2D-F1	S2D-F1	MAP	Acc@1	Acc@3	Acc@5	D2D-F1	S2D-F1	MAP	Acc@1	Acc@3	Acc@5
Single-pair baselines	TF-IDF	84.5	69.6	92.4	97.3	99.8	99.8	90.4	27.1	52.9	36.0	65.9	79.6
	Rouge	82.8	59.5	92.2	98.1	99.7	99.9	71.3	21.3	47.1	30.2	58.0	73.0
	SBERT-STS	78.8	32.8	76.6	91.7	99.0	**100.0**	83.3	23.6	48.0	30.4	59.6	75.4
	SBERT-PD	80.8	36.8	80.8	95.1	99.2	99.8	87.6	22.2	44.4	26.7	54.5	71.2
	SBERT-IR	76.0	41.6	77.5	89.5	98.3	99.5	89.9	25.4	50.6	33.5	62.5	77.6
All-pairs baselines	TF-IDF	86.7	79.4	94.5	97.7	99.7	**100.0**	91.1	26.5	51.1	33.6	63.7	78.8
	Rouge	91.8	80.8	94.9	98.2	99.7	99.8	74.9	21.3	45.4	27.4	56.6	73.8
	SBERT-STS	89.3	74.4	93.1	98.2	99.8	99.9	85.9	24.6	48.3	30.5	60.3	76.3
	SBERT-PD	90.8	76.1	94.4	98.5	99.8	99.9	88.6	21.7	42.9	24.8	53.2	70.3
	SBERT-IR	87.3	71.8	91.8	97.9	99.4	99.8	90.8	25.5	50.4	32.8	62.7	78.0
Hierarchical Neural Models	Bert-Han (Shallow)	74.6	44.3	54.6	46.2	74.5	86.4	90.8	10.0	37.7	19.4	46.2	63.1
	Gru-Han (Deep)	77.2	44.3	54.0	42.8	76.0	89.4	91.8	10.2	40.4	21.5	50.0	68.1
	BCL-CDA	74.1	68.8	81.1	78.1	90.5	94.6	88.6	37.2	58.9	42.5	72.3	84.9
Fine-tuned BERT	Roberta	**95.0**	**82.2**	**96.8**	**99.2**	**99.9**	**100.0**	88.8	54.2	76.7	65.7	88.3	94.4
	Biomed-Roberta	–	–	–	–	–	–	90.3	54.0	77.6	66.8	89.2	94.7
	CS-Roberta	–	–	–	–	–	–	89.9	54.1	**77.7**	**67.2**	**89.3**	**94.9**
	Longformer$_{sequential}$	76.6	68.3	89.6	90.7	96.3	98.5	**96.6**	**58.5**	75.5	63.6	88.0	94.4

7.1 Impact of Weak Supervision

In general, the supervised BERT-based models outperform the unsupervised lexical overlap baselines. The exception to this finding is Pr2News, where the lexical overlap baselines Rouge$_{ap}$ and Rouge$_{sp}$ have the best corpus-level and document-level S2D scores, respectively. This result is perhaps not unexpected, since, unlike other datasets, the labeling methods of Pr2News differ substantially between training (heuristic generated by TFIDF$_{ap}$ scores), validation (non-expert-labeled) and test (expert-labeled). However, our results still contrast Dehghani et al. (2017), who, working on a document ranking task, find that weakly-supervised neural models consistently outperform the unsupervised methods used to label their training data. We hypothesize that our negative finding might be due, in part, to the small scale of Pr2News and our reliance on only a single heuristic as the supervision signal source. To address this, future work could explore applications on larger weakly-supervised LTRD datasets, e.g. closer in scale to the 50M document collection of Dehghani et al. (2017), and improving the weak-supervision signal to better reflect human judgements, e.g. through combination of multiple heuristics (Boecking et al., 2021).

7.2 Effectiveness of Off-the-shelf Tools

Next, we take a closer look at the performances of SBERT (Reimers and Gurevych, 2019). Note, these off-the-shelf models are trained on the related tasks of either PD, STS, or IR, *not* on our LTRD datasets. Though PD, STS, and IR receive substantially more attention in the NLP and IR literature, prior research has not yet explored the generalizability of models trained on these tasks to LTRD. We focus in particular on SBERT-PD, since Reimers and Gurevych (2019) recommend it for various applications and claim that it achieves strong results on various similarity and retrieval tasks. Examining our results, however, we find the opposite – SBERT performs worse in general than the lexical overlap baselines, and SBERT-PD performs no better than SBERT-IR (though both better than SBERT-STS). We suspect that the SBERT models would perform better if they were fine-tuned on in-domain LTRD data. However, since we aimed to evaluate the effectiveness of an off-the-shelf tool, we did not test this hypothesis.

7.3 Importance of Document-level Context

To examine the importance of incorporating document-level context for LTRD, we compare the results of Roberta and Longformer.[7] As noted in §4, input to both models follows the standard BERT sequence-pair setup (Devlin et al., 2019). However, Roberta operates on pairs of source documents and single target sentences ($\mathbf{S}$, $\mathbf{t_i}$), while Longformer operates on full document pairs ($\mathbf{S}$, $\mathbf{T}$), making predictions for all target sentences simultaneously.

From Tables 2 & 3, we see that modeling target document context does not consistently improve performance. While Longformer outperforms Roberta on the D2D and corpus-level S2D tasks on most datasets, Roberta consistently scores higher on document-level S2D. To investigate the discrepancy between Longformer's strong corpus-level S2D performance and its relatively weaker document-level S2D scores, we examine S2ORC

[7] Longformer is initialized from Roberta$_{Base}$, but has additional parameters and is further pretrained on a long-document corpus. Thus, though we cannot disentangle these effects from the benefits of incorporating document-level context, we believe our experiments provide a relatively fair comparison between two SoTA models for short vs. long input sequences.

Table 3: D2D and S2D results on ARC-Sim and S2D results on Pr2News.

Setting	Model	ARC-Sim						Pr2News				
		D2D-F1	S2D-F1	MAP	Acc@1	Acc@3	Acc@5	ρ	NDCG@N	NDCG@1	NDCG@3	NDCG@5
Single-pair baselines	TF-IDF	77.4	18.4	44.8	30.1	52.5	65.5	60.0	90.8	76.0	70.0	71.0
	Rouge	75.3	14.6	36.6	21.8	42.4	55.5	64.5	94.2	**77.3**	**77.0**	**77.7**
	SBERT-STS	76.3	15.0	39.6	23.6	47.0	61.4	35.5	81.6	40.3	42.8	49.2
	SBERT-PD	77.2	16.6	41.4	25.8	48.1	62.5	36.4	82.6	48.7	51.2	55.4
	SBERT-IR	76.9	16.9	41.3	25.7	49.2	62.8	35.3	79.7	54.0	54.7	54.5
All-pairs baselines	TF-IDF	77.3	18.3	43.1	27.6	51.4	64.3	66.7	97.1	66.7	69.8	72.8
	Rouge	75.1	12.9	35.9	20.2	42.0	56.4	**67.2**	**97.3**	69.7	74.6	77.4
	SBERT-STS	76.7	15.8	39.2	23.1	46.7	60.8	58.0	94.8	59.2	61.1	64.4
	SBERT-PD	77.2	16.6	40.4	24.4	47.8	61.3	63.0	96.4	70.3	70.8	71.4
	SBERT-IR	77.4	16.9	40.2	24.3	47.9	62.5	56.1	94.2	56.3	62.4	63.3
Hierarchical Neural Models	Bert-Han (Shallow)	78.7	7.5	35.5	19.8	41.1	56.5	–	–	–	–	–
	Gru-Han (Deep)	79.7	22.1	43.5	27.2	53.2	67.3	–	–	–	–	–
	BCL-CDA	79.9	30.8	55.3	38.5	67.7	80.4	26.3	80.2	24.3	32.5	37.1
Fine-tuned BERT	Roberta	82.0	41.6	**69.7**	**57.0**	**81.8**	**90.4**	60.5	93.7	71.3	68.6	71.4
	Biomed-Roberta	–	–	–	–	–	–	60.1	93.3	68.6	66.2	69.9
	News-Roberta	–	–	–	–	–	–	60.6	93.4	68.3	67.8	70.6
	CS-Roberta	81.6	44.0	69.5	56.7	81.7	89.8	–	–	–	–	–
	Longformer$_{sequential}$	**84.5**	**46.5**	68.3	55.0	80.9	88.8	62.7	95.9	69.5	70.4	70.8

and ARC-Sim. At the corpus-level, Roberta mostly makes false positives (FP) errors, while Longformer makes roughly equal FP and FN errors (and fewer errors overall). For both models, most of these FPs occur in positive ($\mathbf{S}$, $\mathbf{T}$) pairs, i.e. pairs where at least one $\mathbf{t_i}$ cites $\mathbf{S}$. As discussed in §5, these errors are reasonable, since $\mathbf{T}$ often only cites $\mathbf{S}$ once, even if it discusses $\mathbf{S}$ in multiple sentences (Qazvinian and Radev, 2010). Roberta's more-frequent FP errors, however, do not affect its document-level scores as much. Since, at the document-level, we evaluate how well models *rank* the $\mathbf{t_i}$ in each $\mathbf{T}$ w.r.t each other, models perform well if they score positive sentences higher than negatives (no reuse). Indeed, though Roberta predicts high scores for many negatives, it does better than Longformer at scoring positives higher, leading to better ranking performance.

Next, we first perform error analysis on PAN, the only dataset where Roberta outperforms Longformer across all metrics. We find that Roberta makes few D2D errors, of which most (80%) are FPs. Longformer, on the other hand, not only makes substantially more errors overall, but splits them roughly equally between FPs and FNs. These FNs are especially surprising since many positive examples in PAN have high lexical overlap. On the other hand, for the corpus-level S2D task, we find that both models have similar numbers of TPs and FNs, but that Longformer generates an order of magnitude more FPs, i.e. predicting that negative target sentences contain reuse.

7.4 Effects of Domain-adaptive Pretraining

We next examine the benefits of DAPT. Gururangan et al. (2020) find that further pretraining Roberta on text from a new domain improves downstream performance, provided that this new domain is similar to the downstream task. To examine whether this finding holds for LTRD, we conduct DAPT evaluations on three datasets – S2ORC, ARC-Sim and Pr2News. Unlike Gururangan et al. (2020), however, we find mixed results. On ARC-Sim and Pr2News, standard Roberta models outperform the corresponding DAPT models on most metrics. The ARC-Sim findings are especially surprising, since its domain (NLP papers) is substantially different from Roberta's standard pretraining data (books, news, web documents) and since Gururangan et al. (2020) show strong performance gains from DAPT on a classification dataset also based on ACL-ARC. Moving on to S2ORC, our findings are reversed, with both DAPT models outperforming Roberta. However, as noted in §6, since the extra pretraining data for these DAPT models is sampled from the same corpus as S2ORC, we cannot be sure how much of this boost is due to DAPT models pretraining on S2ORC's test data.

7.5 Trade-offs between Models

Finally, we discuss the trade-offs between models, focusing on differences in performance and relative computational efficiency. On one end of the efficiency spectrum are the lexical overlap metrics (TFIDF, Rouge-{1,2}) which are easily scaled to large document collections by simply keeping track of the ngrams in each source or target passage, then computing word-overlap scores for each ($\mathbf{S}$, $\mathbf{T}$) pair.[8] As discussed in §4, we evaluate these metrics in two settings, sp and ap, depending on whether we compute similarity scores between target sen-

[8]Rouge-L cannot be scaled as easily as the other lexical overlap baselines. However, it performs worse than Rouge-1 and -2 on all validation sets and is not applied to any test data.

tences and entire source documents or with each source sentence separately (then compute an aggregate score). Though no single metric or evaluation setting consistently achieves the best performance, these models provides a very strong baseline, especially on the D2D task.

In the middle of the efficiency spectrum are SBERT and HNM. Though these models require an expensive forward pass to generate an embedding for each source or target passage, these embeddings can then be saved and reused. Scores for each ($\mathbf{S}$, $\mathbf{T}$) pair can be computed relatively quickly by either computing cosine similarity scores (SBERT) or running the pair through a lighter-weight task-specific architecture (HNM). However, we find mixed and negative results regarding their effectiveness. Specifically, as discussed in §7.2, off-the-shelf SBERT models generally lag behind the computationally-cheaper lexical overlap baselines. Results are slightly more positive, though, for the HNMs. BCL-CDA, the best HNM, achieves the second best performance on two datasets (S2ORC, ARC-Sim). However, it still lags behind the best model, fine-tuned BERT, by a significant margin. Further, it performs worse than lexical overlap baselines on the other datasets, PAN and Pr2News. Turning next to the HAN models, we find that though they achieve competitive D2D performance on two of the three datasets, they have very weak S2D scores. We suspect that this is because they are only trained on the D2D task – at test time, they make sentence-level predictions by computing similarity scores between hidden source and target representations extracted from a pretrained D2D model. Due to this training formulation, the HAN models fail to learn sentence-level representations that are useful for prediction. See Appendix B for a discussion of our efforts to replicate the results from the HAN models on our datasets.

Lastly, the least efficient models are fine-tuned BERTs, which require a separate forward pass to compute a score for each ($\mathbf{S}$, $\mathbf{T}$) or ($\mathbf{S}$, $\mathbf{t_i}$) pair. As is the trend with other NLP tasks, though, these computationally-intense and parameter-rich models achieve the best average performance. This finding is clearest on S2ORC and ARC-Sim, where few $\mathbf{t_i}$ contain reuse and that reuse is non-literal (e.g. paraphrase). On these datasets, the best fine-tuned BERT outperforms the next-best model (BCL-CDA) by an average of 6.3% (D2D) and 15.5% (S2D). However, on datasets where target documents directly copy large spans of source content with minimal changes (PAN) or where large-scale supervised training data is unavailable (Pr2News), fine-tuned BERT provides much less or no improvement over the lexical overlap metrics.

8 Conclusion

We study methods for local text reuse detection, identifying passages in a target document that lexically or semantically reuse content from a source. Through evaluations on four datasets, including a new citation localization dataset, we confirm the strong performance of BERT models fine-tuned on our task. However, we also find that lexical-overlap methods, e.g. TFIDF, provide strong baselines, frequently outperforming off-the-shelf neural passage encoders and hierarchical neural models.

Based on these findings, we suggest practitioners take one of two approaches: **1)** in instances with little labeled training data or where most reuse is exact (i.e. copying), use traditional lexical overlap models; **2)** in instances with large-scale labeled training data and where much of the reuse is non-literal (e.g. summarization, paraphrasing), use a lexical overlap method to filter possible ($\mathbf{S}$, $\mathbf{T}$) pairs, then run a more expensive fine-tuned BERT on that subset. We suggest users opt for fine-tuned BERT models over pretrained passage encoders (SBERT) or HNMs for this second step since they achieve substantially higher performance. Suggestion #2 follows current approaches to neural IR, where neural models only rerank smaller lists of documents retrieved by a cheaper lexical overlap method, e.g. TF-IDF. Performance *may* be further boosted by fine-tuning BERT-based models that incorporate document-level context (i.e. Longformer) or ones that are adapted to the target domain of interest (i.e. DAPT), but often the standard Roberta$_{Base}$ achieves highly competitive results.

Acknowledgements

Ansel MacLaughlin was supported by a National Endowment for the Humanities Digital Humanities Advancement Grant (HAA-263837-19) and a Northeastern University Dissertation Completion Fellowship.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado,

Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Amjad Abu-Jbara and Dragomir Radev. 2012. Reference scope identification in citing sentences. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 80–90, Montréal, Canada. Association for Computational Linguistics.

Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Applying BERT to document retrieval with birch. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 19–24, Hong Kong, China. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *ArXiv*, abs/2004.05150.

Steven Bird, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev, and Yee Fan Tan. 2008. The ACL Anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Benedikt Boecking, W. Neiswanger, E. Xing, and A. Dubrawski. 2021. Interactive weak supervision: Learning useful heuristics for data labeling. In *ICLR*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Daniel Fernando Campos, T. Nguyen, M. Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, L. Deng, and Bhaskar Mitra. 2016. Ms marco: A human generated machine reading comprehension dataset. *ArXiv*, abs/1611.09268.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Paul Clough, Robert Gaizauskas, Scott S.L. Piao, and Yorick Wilks. 2002. Measuring text reuse. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 152–159, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Arman Cohan, Iz Beltagy, Daniel King, Bhavana Dalvi, and Dan Weld. 2019. Pretrained language models for sequential sentence classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3693–3699, Hong Kong, China. Association for Computational Linguistics.

Isaac Councill, C. Lee Giles, and Min-Yen Kan. 2008. ParsCit: an open-source CRF reference string parsing package. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

W. Croft, Donald Metzler, and Trevor Strohman. 2009. *Search Engines - Information Retrieval in Practice*. Pearson.

Zhuyun Dai and J. Callan. 2019. Deeper text understanding for ir with contextual neural language modeling. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

M. Dehghani, Hamed Zamani, A. Severyn, J. Kamps, and W. Croft. 2017. Neural ranking models with weak supervision. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Kellen Funk and Lincoln Mullen. 2018. The spine of American law: Digital text analysis and U.S. legal practice. *The American Historical Review*, 123(1):1–39.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

John Lee. 2007. A computational model of text reuse in ancient literary texts. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 472–479, Prague, Czech Republic. Association for Computational Linguistics.

Jure Leskovec, Lars Backstrom, and Jon Kleinberg. 2009. Meme-tracking and the dynamics of the news cycle. In *KDD*, pages 497–506.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Ansel MacLaughlin, J. Wihbey, Aleszu Bajak, and D. A. Smith. 2020. Source attribution: Recovering the press releases behind health science news. In *ICWSM*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL*, pages 55–60.

Maria Moritz, Andreas Wiederhold, Barbara Pavlek, Yuri Bizzoni, and Marco Büchler. 2016. Non-literal text reuse in historical texts: An approach to identify reuse transformations and its application to Bible reuse. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1849–1859, Austin, Texas. Association for Computational Linguistics.

Vlad Niculae, Caroline Suen, Justine Zhang, Cristian Danescu-Niculescu-Mizil, and J. Leskovec. 2015. Quotus: The structure of political media coverage as revealed by quoting patterns. *Proceedings of the 24th International Conference on World Wide Web*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Martin Potthast, Tim Gollub, Matthias Hagen, Martin Tippmann, Johannes Kiesel, Paolo Rosso, Efstathios Stamatatos, and Benno Stein. 2013. Overview of the 5th international competition on plagiarism detection. In *Working Notes Papers of the CLEF 2013 Evaluation Labs*.

Vahed Qazvinian and Dragomir R. Radev. 2010. Identifying non-explicit citing sentences for citation-based summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 555–564, Uppsala, Sweden. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance

study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

M. Sánchez-Pérez, G. Sidorov, and Alexander Gelbukh. 2014. A winning approach to text alignment for text reuse detection at pan 2014. In *CLEF*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

David A. Smith, Ryan Cordell, Elizabeth Maddock Dillon, Nick Stramp, and John Wilkerson. 2014. Detecting and modeling local text reuse. In *JCDL*.

Chenhao Tan, Adrien Friggeri, and Lada A. Adamic. 2016. Lost in propagation? unfolding news cycles from the source. In *ICWSM*.

K. Vani and D. Gupta. 2017. Detection of idea plagiarism using syntax-semantic concept extractions with genetic algorithm. *Expert Syst. Appl.*, 73:11–26.

John Wilkerson, David A. Smith, and Nick Stramp. 2015. Tracing the flow of policy ideas on legislatures: A text reuse approach. *American Journal of Political Science*, 59(4):943–956.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

Xuhui Zhou, Nikolaos Pappas, and Noah A. Smith. 2020. Multilevel text alignment with cross-document attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5012–5025, Online. Association for Computational Linguistics.

A Data Preprocessing

Table 4 lists the training, validation and test set sizes for each dataset. Each split is separated into the number of positive examples that contain reuse and the number of negative examples that do not. Below we discuss the data preprocessing steps we follow for each dataset:

ARC-Sim We create this dataset using papers from the ACL Anthology Conference Corpus (Bird et al., 2008). Since we use citation marks to identify instances of text reuse, we use ParsCit (Councill et al., 2008) to first identify all in-line citation marks. We then create examples by matching together a section in a paper that contains a citation with the abstract of the cited paper (assuming the cited paper is also in the ACL ARC). Since citation marks have a distinctive lexical pattern, we remove them all after matching the pairs. We then split sections and abstracts into sentences using Stanford CoreNLP (Manning et al., 2014), keeping track of where the original citation was in order to generate S2D labels. We create negative examples by matching a cited abstract together with another section from the same paper as the original citing section (the new section is selected so that it does not cite the paper). Finally, for computational feasibility, we limit source documents to 20 sentences and target sections to 50, the 90th percentiles in the data. We remove pairs where the citation occurs after the 50th sentence in the target section. We split the dataset into train/val/test by cited abstract **S**, yielding the splits detailed in Table 4.

PAN: We download the public dataset. We filter out 1) malformed positive pairs that do not contain any positively-labeled sentences or contain positively-labeled sentences with no words; 2) extremely long pairs which cause GPU memory issues for our models, removing (source, target) pairs that contain more than 4,000 tokens total (80th percentile). Following Zhou et al. (2020), we split documents into sentences and tokenize them using NLTK (Bird and Loper, 2004).

For the hierarchical neural models (BERT-HAN, GRU-HAN, BCL-CDA), we follow Zhou et al. (2020) and cap documents at a predefined number of sentences so that the models fit in GPU memory. We cap source documents at 50 sentences (90th percentile). We split examples with target documents containing more than 45 sentences (90th percentile) into multiple examples, i.e. (source document, first 45 sentences of target document), (source docu-

Table 4: Number of examples in the training, validation and test sets of each dataset, split into numbers of positive and negative examples. Pr2News contains no negative examples.

	Train		Val		Test	
Dataset	# Pos	# Neg	# Pos	# Neg	# Pos	# Neg
PAN	6,152	7,567	1,243	1,336	1,253	1,352
S2ORC	74,807	75,861	9,262	9,562	9,258	9,561
Pr2News	64,684	–	45	–	50	–
ARC-Sim	50,197	36,227	5,269	3,852	5,665	4,171

ment, next 45 sentences of target document), and so on. Predictions for split examples are merged back together at test time.

S2ORC: We download the public dataset. As for PAN, we filter out malformed positive pairs that do not contain any positively-labeled sentences or contain positively-labeled sentences with no words. S2ORC examples, are, in general, short and do not require length-based filtering. Following Zhou et al. (2020), we split documents into sentences and tokenize them using NLTK (Bird and Loper, 2004).

For the hierarchical neural models (BERT-HAN, GRU-HAN, BCL-CDA), we cap source documents at 20 sentences (99th percentile). We split examples with target documents containing more than 29 sentences (99th percentile) into multiple examples and merge back predictions at test time.

Pr2News: We obtain the preprocessed and filtered Pr2News dataset from MacLaughlin et al. (2020, §4-5), who created it with data from Altmetric. We evaluate models on the provided test set of 50 expert-labeled (press release, news article) pairs. We use the set of 45 non-expert-labeled (press release, news article) pairs as our validation set (we filter out the 5 spurious validation set pairs noted by MacLaughlin et al. (2020)). Finally, we use the remaining 64,684 pairs labeled with their TF-IDF cosine similarity heuristic as training data. For pairs with more than one matched press release, we select the press release with the highest TFIDF cosine similarity to the news article.

For the hierarchical neural models (BERT-HAN, GRU-HAN, BCL-CDA), we cap source documents at 54 sentences (90th percentile). We split examples with target documents containing more than 57 sentences (90th percentile) into multiple examples and merge back predictions at test time.

B Implementation of BERT-HAN and GRU-HAN

Although we use the official source code from Zhou et al. (2020) to run the HAN models, our results differ on PAN and S2ORC from their originally reported results (mostly slightly, but, in one instance, substantially). With the exception of using BERT_{BASE} as the passage encoder for BERT-HAN instead of BERT_{LARGE}, we follow their recommended hyperparameters. But, as compared with the results from Zhou et al. (2020), on the D2D task (measured by F1), BERT-HAN's scores are substantially lower on PAN and slightly lower on S2ORC. GRU-HAN's scores, on the other hand, are very slightly higher on both PAN and S2ORC. We hypothesize that the minor differences in performance are due to **1)** differences in model random initialization (Reimers and Gurevych, 2017); **2)** differences in the datasets – as noted in Appendix A, we filtered out some examples from PAN and S2ORC since they contained some malformed positive examples with either no positively-labeled sentences or positively-labeled sentences that were empty strings; **3)** for BERT-HAN, we use BERT_{BASE} as the encoder rather than BERT_{LARGE}. Despite these factors, BERT-HAN's large performance drop on PAN is still surprising. However, we emphasize that even when using Zhou et al. (2020)'s original numbers, BERT-HAN still lags behind both our lexical overlap baselines and fine-tuned BERT models, so our overall takeaways from §7 still stand.

For the S2D task, our results are not directly comparable to the original numbers of Zhou et al. (2020) for two reasons:

1. We use different metrics – we use MAP and Acc@k, while they use MRR and P@k. MAP is more appropriate than MRR since there are often multiple positively-labeled target sentences. Acc@k is more appropriate than P@k when k is greater than the number of

positively-labeled target sentences. When there are fewer than k positively-labeled target sentences in an example, a perfect system will still have a P@k < 1. Systems receive a perfect Acc@k score, on the other hand, if at least one positively-labeled target sentence appears anywhere in the top k.

2. We evaluate on different sets of the data – as noted in §6 , Zhou et al. (2020) calculate their S2D ranking metrics (MRR, P@k) on all test examples, both positive and negative. However, these metrics cannot be computed on negative examples where no target sentences contain reuse. We confirmed with Zhou et al. (2020) that, in these instances, they give their models full credit if the corresponding D2D prediction is correct, i.e. the model predicts that the target document contains no reuse. Since many negative examples in PAN and S2ORC are easy to classify, this manner of calculation substantially inflates the results. To address this, we calculate our S2D ranking metrics (MAP, Acc@k) on only the subset of positive examples. Calculating in this way shows substantially decreased S2D performance for the HAN models.

C Model Hyperparameters & Best Configurations

Below, we discuss all searched hyperparameters (HP) for each model. For all models, we search for a threshold, $t \in [0, 1]$, to differentiate between positive and negative sentence and document predictions (not used for the Pr2News dataset). Table 5 lists the optimal HP values for each dataset (as selected by average performance on the val set).

All neural models, with the exception of BCL-CDA (Tensorflow: Abadi et al. (2015)) were implemented in Pytorch (Paszke et al., 2019) and run on 16GB or 32GB Nvdia P100s or V100s.

TF-IDF: We search over n-gram size (unigrams or unigrams & bigrams).

Rouge: We search over three different Rouge measures, Rouge-{1, 2, L}.

Sentence-BERT: None except threshold. We test the following pretrained Sentence-BERT models: Semantic Textual Similarity: stsb-roberta-large, Paraphrase Detection: paraphrase-distilroberta-base-v1, Information Retrieval: msmarco-distilroberta-base-v2.

BERT-HAN (shallow): We use the suggested batch size (256), HAN hidden dimension size (50), and early stopping criterion (no improvement on val set for 5 epochs). We perform a search over Adam (Kingma and Ba, 2015) learning rates $\in$ {1e-5, 2e-5, 5e-5, 1e-4}. We use BERT$_{BASE}$ as the sentence encoder instead of BERT$_{LARGE}$ for efficiency reasons. For the S2ORC and ARC-Sim datasets, we find that BERT-HAN's S2D performance is substantially higher when we rank sentences by the complement of their scores, i.e. $score_{new} = 1 - score_{old}$. This, in effect, inverts the predicted target sentence ranking for each document (we select this transformation based on val set results). We are only able to achieve results roughly on par with those reported in Zhou et al. (2020) using this trick. This trick is not necessary for the PAN dataset nor for the GRU-HAN model.

GRU-HAN (deep): We use batch size 128 and 50 dimensional GloVe embeddings. Otherwise, the HPs are the same as for BERT-HAN.

BCL-CDA: We adapt the BCL model from MacLaughlin et al. (2020) for LTRD as follows (see MacLaughlin et al. (2020) for details of the BCL model): Each source and target sentence is fed into frozen BERT$_{BASE}$ separately. We then use a CNN with 1-max pooling over time to aggregate the token representations from BERT's second to last layer into a single representation for each sentence. We search over CNN filter size $\in$ {3,5,7} and number of filters $\in$ {50, 100, 200}. The sentence representations in each source or target document are then contextualized with document-level BiLSTMs (two separate BiLSTMs for source or target documents). We search over hidden dimension size $\in$ {64, 128} (same dimensionality for both BiLSTMs). After the BiLSTM layer, we are left with $s_i \in \mathbf{S}$ and $t_i \in \mathbf{T}$, contextualized sentence representations for the sentences in the source and target documents. Next, we use a sentence-level CDA layer to compute $\tilde{t}_i$, an attention-weighted (Luong et al., 2015, §3.1: general attention) representation of t_i, weighted by its similarity to the sentences $s_i \in \mathbf{S}$. Finally, we concatenate $[\tilde{t}_i;t_i]$ and feed this to a final layer to make a prediction for each target sentence.

We set dropout at 0.2, batch size at 32, and search over the max number of epochs (10, with early stopping). We optimize with Adam with learning rate $\in$ {1e-4, 5e-4}. For the PAN, S2ORC and ARC-Sim datasets, we use weighted cross-

Table 5: Best HP configurations for all models across all datasets. t is the classification threshold (only for PAN, S2ORC and ARC-Sim). BERT-HAN and GRU-HAN have two thresholds, one for document classification, the other for sentence classification. All other models have a single, sentence-level threshold. n-gram is the n-gram range for TF-IDF (unigrams or unigrams and bigrams). For the neural models: e is epochs, lr is learning rate, and w is the weight placed on positive examples in weighted cross-entropy loss (weight on negative examples is 1). For BCL-CDA, fs is the CNN filter size, nf is number of CNN filters, and lhd is the BiLSTM hidden dimension. '–' indicates that there are no HPs to be optimized. '×' indicates that the model is not trained on that dataset.

		PAN	S2ORC	ARC-Sim	Pr2News
Single-pair baselines	TF-IDF	$t = 0.10$, n-gram = 1 & 2	$t = 0.04$, n-gram = 1 & 2	$t = 0.04$, n-gram = 1 & 2	n-gram = n-gram = 1 & 2
	Rouge	$t = 0.03$, rouge = R-2	$t = 0.02$, rouge = R-2	$t = 0.02$, rouge = R-2	rouge = R-2
	Roberta$_{mean}$ (STS)	$t = 0.46$	$t = 0.48$	$t = 0.48$	–
	DistilRoberta$_{mean}$ (PD)	$t = 0.41$	$t = 0.39$	$t = 0.42$	–
	DistilRoberta$_{mean}$ (IR)	$t = 0.38$	$t = 0.38$	$t = 0.44$	–
All-pairs baselines	TF-IDF	$t = 0.17$, n-gram = 1 & 2	$t = 0.14$, n-gram = 1	$t = 0.14$, n-gram = 1	n-gram = 1 & 2
	Rouge	$t = 0.40$, rouge = R-1	$t = 0.27$, rouge = R-1	$t = 0.24$, rouge = R-1	rouge = R-2
	Roberta$_{mean}$ (STS)	$t = 0.63$	$t = 0.53$	$t = 0.52$	–
	DistilRoberta$_{mean}$ (PD)	$t = 0.58$	$t = 0.42$	$t = 0.45$	–
	DistilRoberta$_{mean}$ (IR)	$t = 0.61$	$t = 0.43$	$t = 0.48$	–
Frozen BERT	Bert-Han (Shallow)	doc $t = 0.41$, sent $t = 0$, lr =5e-5	doc $t = 0.34$, sent $t = 0.0$, lr =2e-5	doc $t = 0.33$, sent $t = 0.08$, lr = 1e-4	×
	Gru-Han (Deep)	doc $t = 0.34$, sent $t = 0$, lr = 1e-5	doc $t = 0.33$, sent $t = 0.01$, lr = 5e-5	doc $t = 0.42$, sent $t = 0.11$, lr = 1e-4	×
	BCL-CDA	$t = 0.57$, lr =5e-4, $e = 7$, $w = 5$, $fs = 5$, $nf = 50$, $lhd = 128$	$t = 0.44$, lr =5e-4, $e = 5$, $w = 5$, $fs = 3$, $nf = 50$, $lhd = 128$	$t = 0.74$, lr =1e-4, $e = 3$, $w = 15$, $fs = 5$, $nf = 200$, $lhd = 64$	lr =5e-4, e =7, $fs = 7$, $nf = 200$, $lhd = 128$
Fine-tuned BERT	Biomed-Roberta$_{single}$	×	$t = 0.34$, lr = 2e-5, $e = 2$, $w = 3$	×	lr = 3e-5, $e = 1$
	CS-Roberta$_{single}$	×	$t = 0.47$, lr = 2e-5, $e = 3$, $w = 3$	$t = 0.51$, lr = 2e-5, $e = 5$, $w = 10$	×
	News-Roberta$_{single}$	×	×	×	lr = 3e-5, $e = 1$
	Roberta$_{single}$	$t = 0.91$, lr =3e-5, $e = 4$, $w = 1$	$t = 0.4$, lr = 2e-5, $e = 4$, $w = 3$	$t = 0.39$, lr = 2e-5, $e = 3$, $w = 5$	lr = 2e-5, $e = 2$
	Longformer$_{seq}$	$t = 0.52$, $lr = 5e-5$, $e = 9$, $w = 5$	$t = 0.41$, lr =3e-5, e =19, w =5	$t = 0.53$, lr =3e-5, e =12, $w = 5$	$lr = 5e-5$, $e = 7$

entropy loss since the datasets are unbalanced (many more negative sentences than positive). We search over the weight w to put on examples from the positive class. Weights vary by dataset since datasets are not equally imbalanced: PAN $\in \{1, 3, 5\}$, S2ORC $\in \{3, 5, 10\}$, ARC-Sim $\in \{10, 15, 20\}$. Following MacLaughlin et al. (2020), we use MAE loss for Pr2News.

Fine-tuned RoBERTa$_{BASE}$, DAPT, and Longformer: We search over Adam learning rate $\in \{$2e-5, 3e-5, 5e-5$\}$. We use batch size 32 (with gradient accumulation to ensure that batches fit in GPU memory) and train models for 10 epochs at most (20 for Longformer), with early stopping. For PAN, S2ORC and ARC-Sim, following BCL-CDA, we search over weight w for weighted cross-entropy loss. We search over the same w ranges for each dataset as for BCL-CDA, except for ARC-Sim, where we search over $w \in \{5, 10, 20\}$. We use MAE loss for Pr2News.