Compressive Summarization with Plausibility and Salience Modeling

Shrey Desai **Jiacheng Xu** **Greg Durrett**
Department of Computer Science
The University of Texas at Austin
shreydesai@utexas.edu {jcxu, gdurrett}@cs.utexas.edu

Abstract

Compressive summarization systems typically rely on a crafted set of syntactic rules to determine what spans of possible summary sentences can be deleted, then learn a model of what to actually delete by optimizing for content selection (ROUGE). In this work, we propose to relax the rigid syntactic constraints on candidate spans and instead leave compression decisions to two data-driven criteria: plausibility and salience. Deleting a span is *plausible* if removing it maintains the grammaticality and factuality of a sentence, and spans are *salient* if they contain important information from the summary. Each of these is judged by a pre-trained Transformer model, and only deletions that are both plausible and not salient can be applied. When integrated into a simple extraction-compression pipeline, our method achieves strong in-domain results on benchmark summarization datasets, and human evaluation shows that the plausibility model generally selects for grammatical and factual deletions. Furthermore, the flexibility of our approach allows it to generalize cross-domain: our system fine-tuned on only 500 samples from a new domain can match or exceed an in-domain extractive model trained on much more data.[1]

1 Introduction

Compressive summarization systems offer an appealing tradeoff between the robustness of extractive models and the flexibility of abstractive models. Compression has historically been useful in heuristic-driven systems (Knight and Marcu, 2000, 2002; Wang et al., 2013) or in systems with only certain components being learned (Martins and Smith, 2009; Woodsend and Lapata, 2012; Qian and Liu, 2013). End-to-end learning-based compressive methods are not straightforward to train: exact derivations of which compressions should be applied are not available, and deriving oracles based on ROUGE (Berg-Kirkpatrick et al., 2011; Durrett et al., 2016; Xu and Durrett, 2019; Mendes et al., 2019) optimizes only for content selection, not grammaticality or factuality of the summary. As a result, past approaches require significant engineering, such as creating a highly specific list of syntactic compression rules to identify permissible deletions (Berg-Kirkpatrick et al., 2011; Li et al., 2014; Wang et al., 2013; Xu and Durrett, 2019). Such manually specified, hand-curated rules are fundamentally inflexible and hard to generalize to new domains.

In this work, we build a summarization system that compresses text in a more data-driven way. First, we create a small set of high-recall constituency-based compression rules that cover the space of legal deletions. Critically, these rules are merely used to propose candidate spans, and the ultimate deletion decisions are controlled by two data-driven models capturing different facets of the compression process. Specifically, we model *plausibility* and *salience* of span deletions. Plausibility is a domain-independent requirement that deletions maintain grammaticality and factuality, and salience is a domain-dependent notion that deletions should maximize content selection (from the standpoint of ROUGE). In order to learn plausibility, we leverage a pre-existing sentence compression dataset (Filippova and Altun, 2013); our model learned from this data transfers well to the summarization settings we consider. Using these two models, we build a pipelined compressive system as follows: (1) an off-the-shelf extractive model highlights important sentences; (2) for each sentence, high-recall compression rules yield span candidates; (3) two pre-trained Transformer models (Clark et al., 2020) judge the plausibility and salience of spans, respectively, and only spans

[1]Code and datasets available at https://github.com/shreydesai/cups

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 6259–6274,
November 16–20, 2020.

which are both plausible and not salient are deleted.

We evaluate our approach on several summarization benchmarks. On CNN (Hermann et al., 2015), WikiHow (Koupaee and Wang, 2018), XSum (Narayan et al., 2018), and Reddit (Kim et al., 2019), our compressive system consistently outperforms strong extractive methods by roughly 2 ROUGE-1, and on CNN/Daily Mail (Hermann et al., 2015), we achieve state-of-the-art ROUGE-1 by using our compression on top of MatchSum (Zhong et al., 2020) extraction. We also perform additional analysis of each compression component: human evaluation shows plausibility generally yields grammatical and factual deletions, while salience is required to weigh the content relevance of plausible spans according to patterns learned during training.

Furthermore, we conduct out-of-domain experiments to examine the cross-domain generalizability of our approach. Because plausibility is a more domain-independent notion, we can hold our plausibility model constant and adapt the extraction and salience models to a new setting with a small number of examples. Our experiments consist of three transfer tasks, which mimic real-world domain shifts (e.g., newswire → social media). By fine-tuning salience with only 500 in-domain samples, we demonstrate our compressive system can match or exceed the ROUGE of an in-domain extractive model trained on tens of thousands of document-summary pairs.

2 Plausible and Salient Compression

Our principal goal is to create a compressive summarization system that makes linguistically informed deletions in a way that generalizes cross-domain, without relying on heavily-engineered rules. In this section, we discuss our framework in detail and elaborate on the notions of plausibility and salience, two *learnable* objectives that underlie our span-based compression.

2.1 Plausibility

Plausible compressions are those that, when applied, result in grammatical and factual sentences; that is, sentences that are syntactically permissible, linguistically acceptable to native speakers (Chomsky, 1956; Schütze, 1996), and factually correct from the perspective of the original sentence. Satisfying these three criteria is challenging: acceptability is inherently subjective and measuring factuality

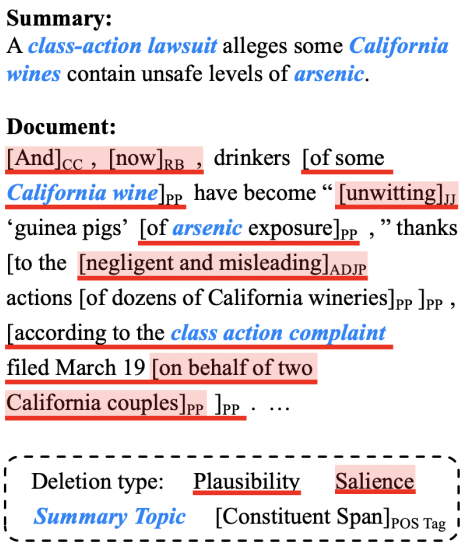

Figure 1: Decomposing span-based compression into plausibility and salience (§2). Plausible compressions (underlined) must maintain grammaticality, thus [to the ... wineries]PP is not a candidate. Salience identifies low-priority content from the perspective of this dataset (highlighted). Constituents both underlined *and* highlighted are deleted.

in text generation is a major open problem (Kryściński et al., 2020; Wang et al., 2020; Durmus et al., 2020; Goyal and Durrett, 2020). Figure 1 gives examples of plausible deletions: note that *of dozens of California wineries* would be grammatical to delete but significantly impacts factuality.

We can learn this notion of plausibility in a data-driven way with appropriately labeled corpora. In particular, Filippova and Altun (2013) construct a corpus from news headlines which can suit our purposes: these headlines preserve the important facts of the corresponding article sentence while omitting minor details, and they are written in an acceptable way. We can therefore leverage this type of supervision to learn a model that specifically identifies plausible deletions.

2.2 Salience

As we have described it, plausibility is a domain-*independent* notion that asks if a compression maintains grammaticality and factuality. However, depending on the summarization task, a compressive system may not want to apply all plausible compressions. In Figure 1, for instance, deleting all plausible spans results in a loss of key information. In addition to plausibility, we use a domain-*dependent* notion of **salience**, or whether a span should be included in summaries of the form we

want to produce.

Labeled oracles for this notion of content relevance (Gillick and Favre, 2009; Berg-Kirkpatrick et al., 2011, *inter alia*) can be derived from gold-standard summaries using ROUGE (Lin, 2004). We compare the ROUGE score of an extract with and without a particular span as a proxy for its importance, then learn a model to classify which spans improve ROUGE if deleted. By deleting spans which are both plausible and salient in Figure 1, we obtain a compressed sentence that captures core summary content with 28% fewer tokens, while still being fully grammatical and factual.

2.3 Syntactic Compression Rules

The base set of spans which we judge for plausibility and salience comes from a recall-oriented set of compression rules over a constituency grammar; that is, they largely cover the space of valid deletions, but include invalid ones as well.

Our rules allow for deletion of the following: (1) parentheticals (PRN) and fragments (FRAG); (2) adjectives (JJ) and adjectival phrases (ADJP); (3) adverbs (RB) and adverbial phrases (ADVP); (4) prepositional phrases (PP); (5) appositive noun phrases (NP_1–[,–NP_2–,]); (6) relative clauses (SBAR); and (7) conjoined noun phrases (e.g., NP_1–[CC–NP_2]), verb phrases (e.g., VP_1–[CC–VP_2]), and sentences (e.g., S_1–[CC–S_2]). Brackets specify the constituent span(s) to be deleted, e.g., CC–NP_2 in NP_1–[CC–NP_2].

Much more refined rules would be needed to ensure grammaticality: for example, in *She was [at the tennis courts]$_{PP}$*, deletion of the PP leads to an unacceptable sentence. However, this base set of spans is nevertheless a good set of building blocks, and reliance on syntax gives a useful inductive bias for generalization to other domains (Swayamdipta et al., 2018).

3 Summarization System

We now describe our compressive summarization system that leverages our notions of plausibility and salience. For an input document, an off-the-shelf extractive model first chooses relevant sentences, then for each extracted sentence, our two compression models decide which sub-sentential spans to delete. Although the plausibility and salience models have different objectives, they both output a posterior over constituent spans, and thus use the same base model architecture.

We structure our model's decisions in terms of separate sentence extraction and compression decisions. Let $S_1, \ldots, S_n$ denote random variables for sentence extraction where $S_i = 1$ indicates that the ith sentence is selected to appear in the summary. Let $C^{\text{PL}}_{11}, \ldots, C^{\text{PL}}_{nm}$, denote random variables for the plausibility model, where $C^{\text{PL}}_{ij} = 1$ indicates that the jth span of the ith sentence is plausible. An analogous set of C^{SAL}_{ij} is included for the salience model. These variables are modeled independently and fully specify a compressive summary; we describe this process more explicitly in Section 4.4.

3.1 Preprocessing

Our system takes as input a document D with sentences $s_1, \cdots, s_n$, where each sentence s_i has words $w_{i1}, \cdots, w_{im}$. We constrain n to be the maximum number of sentences that collectively have less than 512 wordpieces when tokenized. Each sentence has an associated constituency parse T_i (Kitaev and Klein, 2018) comprised of constituents $c = (t, i', j')$ where t is the constituent's part-of-speech tag and (i', j') are the indices of the text span. Let $R(T_i)$ denote the set of spans proposed for deletion by our compression rules (see Section 2.3).

3.2 Extraction

Our extraction model is a re-implementation of the BERTSum model (Liu and Lapata, 2019), which predicts a set of sentences to select as an extractive summary. The model encodes the document sentences $s_1, \cdots, s_n$ using BERT (Devlin et al., 2019), also prepending `[CLS]` and adding `[SEP]` as a delimiter between sentences.[2] We denote the token-level representations thus obtained as: $[\mathbf{h}^{\text{doc}}_{11}, \cdots, \mathbf{h}^{\text{doc}}_{nm}] = \text{Encoder}([s_1, \cdots, s_n])$

During fine-tuning, the `[CLS]` tokens are treated as sentence-level representations. We collect the `[CLS]` vectors over all sentences $\mathbf{h}^{\text{doc}}_{i1}$, dot each with a weight vector $\mathbf{w} \in \mathbb{R}^d$, and use a sigmoid to obtain selection probabilities: $P(S_i = 1|D) = \sigma(\mathbf{h}^{\text{doc}\top}_{i1}\mathbf{w})$

3.3 Compression

Depicted in Figure 2, the compression model (instantiated twice; once for plausibility and once for salience) is a sentence-level model

[2] BERT can be replaced with other pre-trained encoders, such as ELECTRA (Clark et al., 2020), which we use for most experiments.

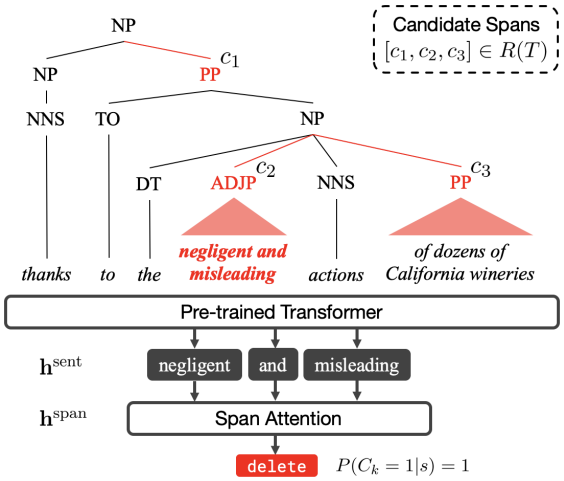

Figure 2: Compression model used for plausibility and salience modeling (§3.3). We extract candidate spans $c_i \in C(T)$ to delete, then compute span embeddings with pre-trained encoders (only one span embedding shown here). This embedding is then used to predict whether the span should be kept or deleted.

that judges which constituent spans should be deleted. We encode a single sentence s_i at a time, adding `[CLS]` and `[SEP]` as in the extraction model. We obtain token-level representations using a pre-trained Transformer encoder:[3]
$[\mathbf{h}^{\text{sent}}_{i1}, \cdots, \mathbf{h}^{\text{sent}}_{im}] = \text{Encoder}([s_i])$

We create a span representation for each constituent $c_k \in C(T_i)$. For the kth constituent, using its span indices (i', j'), we select its corresponding token representations $[\mathbf{h}^{\text{sent}}_{ii'}, \cdots, \mathbf{h}^{\text{sent}}_{ij'}]_k \in \mathbb{R}^{(j'-i')\times d}$. We then use span attention (Lee et al., 2017) to reduce this span to a fixed-length vector $\mathbf{h}^{\text{span}}_k$. Finally, we compute deletion probabilities using a weight vector $\mathbf{w} \in \mathbb{R}^d$ as follows: $P(C^X_k = 1|s_j) = \sigma(\mathbf{h}^{\text{span}\top}_k \mathbf{w})$, where C^X_k is either a plausibility or salience random variable.

3.4 Postprocessing

As alluded to in Section 2.3, there are certain cases where the syntactic compression rules license deleting a chain of constituents rather than individual ones. A common example of this is in conjoined noun phrases (NP_1–[CC–NP_2]) where if the second noun phrase NP_2 is deleted, its preceding coordinating conjunction CC can also be deleted without affecting the grammaticality of the sentence. To avoid changing the compression model substantially, we relegate secondary deletions to a post-processing step, where if a primary constituent like NP_2 is deleted at test-time, its secondary constituents are also automatically deleted.

[3]The encoders between the extraction and compression modules are fine-tuned separately; in other words, our modules do not share any parameters.

4 Training and Inference

The extraction and compression models in our summarization system are trained separately, but both used in a pipeline during inference. Because the summarization datasets we use do not come with labels for extraction and compression, we chiefly rely on structured oracles that provide supervision for our models. In this section, we describe our oracle design decisions, learning objectives, and inference procedures.[4]

4.1 Extraction Supervision

Following Liu and Lapata (2019), we derive an oracle extractive summary using a greedy algorithm that selects up to k sentences in a document that maximize ROUGE (Lin, 2004) with respect to the reference summary.[5]

4.2 Compression Supervision

Because plausibility and salience are two different views of compression, as introduced in Section 2.3, we have different methods for deriving their supervision. However, their oracles share the same high-level structure, which procedurally operate as follows: an oracle takes in as input an uncompressed sentence x, compressed sentence or paragraph y, and a similarity function f. Using the list of available compression rules $R(T_x)$ for x, if x without a constituent $c_k \in R(T_x)$ results in $f(x \backslash c_k, y) > f(x, y)$, we assign c_k a positive "delete" label, otherwise we assign it a negative "keep" label. Intuitively, this oracle measures whether the deletion of a constituent causes x to become closer to y. We set f to ROUGE (Lin, 2004), primarily for computational efficiency, although more complex similarity functions such as BERTScore (Zhang et al., 2020b) could be used without modifying our core approach. Below, we elaborate on the nature of x and y for plausibility and salience, respectively.

Plausibility. We leverage labeled, parallel sentence compression data from news headlines to

[4]See Appendices B and C for training and inference hyperparameters, respectively.

[5]We found that using beam search to derive the oracle yielded higher oracle ROUGE, but also a significantly harder learning problem, and the extractive model trained on this oracle actually performed worse at test time.

learn plausibility. Filippova and Altun (2013) create a dataset of 200,000 news headlines and the lead sentence of its corresponding article, where each headline x is a compressed extract of the lead sentence y. Critically, the headline is a subtree of the dependency relations induced by the lead sentence, ensuring that x and y will have very similar syntactic structure. Filippova and Altun (2013) further conduct a human evaluation of the headline and lead sentence pairs and conclude that, with 95% confidence, annotators find the pairs "indistinguishable" in terms of readability and informativeness. This dataset therefore suits our purposes for plausibility as we have defined it.

Salience. Though the sentence compression data described above offers a reasonable prior on span-level deletions, the salience of a particular deletion is a domain-dependent notion that should be learned from in-domain data. One way to approximate this is to consider whether the deletion of a span in a sentence x_i of an extractive summary increases ROUGE with the reference summary y (Xu and Durrett, 2019), allowing us to estimate what types of spans are *likely* or *unlikely* to appear in a summary. We can therefore derive salience labels directly from labeled summarization data.

4.3 Learning

In aggregate, our system requires training three models: an extraction model (θ_{E}), a plausibility model (θ_{P}), and a salience model (θ_{S}).

The extraction model optimizes log likelihood over each selection decision S_j in document D_i, defined as $\mathcal{L}_{\text{EXT}} = -\sum_{i=1}^{n}\sum_{j\in D_i}\log P(S_j^{(i)} = S_j^{(i)*}|D_i)$ where $S_j^{(i)*}$ is the gold label for selecting the jth sentence in the ith document.

The plausibility model optimizes log likelihood over the oracle decision $C_{jk}^{\text{PL}(i)*}$ for each constituent $c_k \in R(T_j)$ in sentence j, defined as $\mathcal{L}_{\text{CMP}} = -\sum_{j=1}^{m}\sum_{c_k\in R(T_j)}\log P(C_{jk}^{\text{PL}(i)} = C_{jk}^{\text{PL}(i)*}|s_j^{(i)})$. The salience model operates analogously over the C^{SAL} variables.

4.4 Inference

While our sentence selection and compression stages are modeled independently, structurally we need to combine these decisions to yield a coherent summary, recognizing that these models have not been optimized directly for ROUGE.

Our pipeline consists of three steps: (1) For an input document D, we select the top-k sentences with the highest posterior selection probabilities: $\text{argmax}_k P(S_i = 1|D;\theta_{\text{E}})$. (2) Next, for each selected sentence j, we obtain plausible compressions $Z_{\text{P}} = \{c_k|P(C_{jk}^{\text{PL}} = 1|s_j;\theta_{\text{P}}) > \lambda_{\text{P}}, \forall c_k \in R(T_j)\}$ and salient compressions $Z_{\text{S}} = \{c_k|P(C_{jk}^{\text{SAL}} = 1|s_j;\theta_{\text{S}}) > \lambda_{\text{S}}, \forall c_k \in R(T_j)\}$, where λ_{P} and λ_{S} are hyperparameters discovered with held-out samples. (3) Finally, we only delete constituent spans licensed by both the plausibility and salience models, denoted as $Z_{\text{P}} \cap Z_{\text{S}}$, for each sentence. The remaining tokens among all selected sentences form the compressive summary.[6]

We do not perform joint inference over the plausibility and salience models because plausibility is a necessary precondition in span-based deletion, as defined in Section 2.1. If, for example, a compression has a low plausibility score but high salience score, it will get deleted during joint inference, but this may negatively affect the well-formedness of the summary. As we demonstrate in Section 6.3, the plausibility model enforces strong guardrails that prevent the salience model from deleting arbitrary spans that result in higher ROUGE but at the expense of syntactic or semantic errors.

5 Experimental Setup

We benchmark our system first with an automatic evaluation based on ROUGE-1/2/L F_1 (Lin, 2004).[7] Our experiments use the following English datasets: CNN/DailyMail (Hermann et al., 2015), CNN (subset of CNN/DM), New York Times (Sandhaus, 2008), WikiHow (Koupaee and Wang, 2018), XSum (Narayan et al., 2018), and Reddit (Kim et al., 2019).[8]

We seek to answer three questions: (1) How does our compressive system stack up against our own extractive baseline and past extractive approaches? (2) Do our plausibility and salience modules successfully model their respective phenomena? (3) How can these pieces be used to improve cross-domain summarization?

[6] Our pipeline overall requires 3x more parameters than a standard Transformer-based extractive model (e.g., BERTSum). However, our compression module (which accounts for 2/3 of these parameters) can be applied on top of any off-the-shelf extractive model, so stronger extractive models with more parameters can be combined with our approach as well.

[7] Following previous work, we use `pyrouge` with the default command-line arguments: `-c 95 -m -n 2`

[8] See Appendix A for dataset splits.

		CNN			WikiHow			XSum			Reddit		
Type	Model	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
ext	Lead-k	29.80	11.40	26.45	24.96	5.83	23.23	17.02	2.72	13.79	19.64	2.40	14.79
ext	BERTSum	—	—	—	30.31	8.71	28.24	22.86	4.48	17.16	23.86	5.85	19.11
ext	MatchSum$^{\diamondsuit}$	—	—	—	31.85	8.98	29.58	24.86	4.66	18.41	25.09	6.17	20.13
abs	PEGASUS$_{\text{BASE}}$	—	—	—	36.58	15.64	30.01	39.79	16.58	31.70	24.36	6.09	18.75
abs	PEGASUS$^{\heartsuit}_{\text{LARGE}}$	—	—	—	**43.06**	**19.71**	**34.80**	**47.21**	**24.56**	**39.25**	**26.63**	**9.01**	**21.60**
ext	CUPS$_{\text{EXT}}$	33.12	13.88	29.51	30.94	9.06	28.81	24.23	4.95	18.30	24.42	6.10	19.57
cmp	CUPS	**35.22**	**14.19**	**31.51**	**32.43**	**9.44**	**30.24**	**26.04**	**5.36**	**19.90**	**25.99**	**6.57**	**21.08**

Table 1: **Results on CNN, WikiHow, XSum, and Reddit.** Our system consistently achieves higher ROUGE than extraction-only baselines. Additionally, our system achieves higher ROUGE-L than PEGASUS$_{\text{BASE}}$ on WikiHow and Reddit without summarization-specific pre-training. $^{\diamondsuit}$Extractive SOTA; $^{\heartsuit}$Abstractive SOTA.

Type	Model	R1	R2	RL
ext	Lead-3	40.42	17.62	36.67
ext	BERTSum	43.25	20.24	39.63
ext	MatchSum$^{\diamondsuit}$	**44.41**	20.86	40.55
abs	PEGASUS$_{\text{BASE}}$	41.79	18.81	38.93
abs	PEGASUS$^{\heartsuit}_{\text{LARGE}}$	44.17	**21.47**	**41.11**
ext	CUPS$_{\text{EXT}}$ (BERT)	43.16	20.10	39.52
ext	CUPS$_{\text{EXT}}$	43.65	20.57	40.02
cmp	CUPS	44.02	20.57	40.38
cmp	MatchSum + CUPS$_{\text{CMP}}$	**44.69**	**20.71**	**40.86**

Table 2: **Results on CNN/DM.** Notably, a pipeline with MatchSum (Zhong et al., 2020) extraction and our compression module achieves state-of-the-art ROUGE-1. $^{\diamondsuit}$Extractive SOTA; $^{\heartsuit}$Abstractive SOTA.

Systems for Comparison. We refer to our full compressive system as **CUPS**[9], which includes CUPS$_{\text{EXT}}$ and CUPS$_{\text{CMP}}$, the extraction and compression components, respectively. CUPS$_{\text{EXT}}$ is a re-implementation of BERTSum (Liu et al., 2019) and CUPS$_{\text{CMP}}$ is a module consisting of both the plausibility and salience models. The pre-trained encoders in the extraction and compression modules are set to ELECTRA$_{\text{BASE}}$ (Clark et al., 2020), unless specified otherwise.

Because our approach is fundamentally extractive (albeit with compression), we chiefly compare against state-of-the-art extractive models: **BERTSum** (Liu et al., 2019), the canonical architecture for sentence-level extraction with pre-trained encoders, and **MatchSum** (Zhong et al., 2020), a summary-level semantic matching model that uses BERTSum to prune irrelevant sentences. These models outperform recent compressive systems (Xu and Durrett, 2019; Mendes et al., 2019); updating the architectures of these models and extending their oracle extraction procedures to the range of datasets we consider is not straightforward.

To contextualize our results, we also compare against a state-of-the-art abstractive model, **PEGASUS** (Zhang et al., 2020a), a seq2seq Transformer pre-trained with "gap-sentences." This comparison is not entirely apples-to-apples, as this pre-training objective uses very large text corpora (up to 3.8TB) in a summarization-specific fashion. We expect our approach to stack with further advances in pre-training.

Extractive, abstractive, and compressive approaches are typed as `ext`, `abs`, and `cmp`, respectively, throughout the experiments.

6 In-Domain Experiments

6.1 Benchmark Results

Table 1 (CNN, WikiHow, XSum, Reddit) and 2 (CNN/DM) show ROUGE results. From these tables, we make the following observations:

Compression consistently improves ROUGE, even when coupled with a strong extractive model. Across the board, we see improvements in ROUGE when using CUPS. Our results particularly contrast with recent trends in compressive summarization where span-based compression (in joint and pipelined forms) *decreases* ROUGE over sentence extractive baselines (Zhang et al., 2018; Mendes et al., 2019). Gains are especially pronounced on datasets with more abstractive summaries, where applying compression roughly adds +2 ROUGE-1; however, we note there is a large gap between extractive and abstractive approaches on tasks like XSum due to the amount of paraphrasing in reference summaries (Narayan et al., 2018). Nonetheless, our system outperforms strong extractive models on these datasets, and also yields

[9] **C**ompressive **Su**mmarization with **P**lausibility and **S**alience

opening statements in the murder trial of movie theater massacre suspect james holmes are scheduled for april 27~~, more than a month ahead of schedule~~, a colorado court spokesman said. holmes~~, 27,~~ is charged as the ~~sole~~ gunman who stormed a crowded movie theater ~~at a midnight showing of "the dark knight rises" in aurora, colorado~~, and opened fire~~, killing 12 people and wounding 58 more in july 2012~~. holmes~~, a one-time neuroscience doctoral student,~~ faces 166 counts~~, including murder and attempted murder charges~~.
the accident happened in santa ynez california~~, near where crosby lives~~. crosby was driving at ~~approximately~~ 50 mph when he struck the jogger, according to california highway patrol spokesman don clotworthy. the jogger suffered multiple fractures, and was airlifted to a hospital ~~in santa barbara~~, clotworthy said.
update: jonathan hyla said ~~in an phone interview~~ monday that his interview with cate blanchett was mischaracterized ~~when an edited version went viral around the web last week~~. "she wasn't upset," he told cnn. blanchett ended the interview laughing~~, hyla said,~~ and "she was in on the joke."

Table 3: CUPS-produced summaries on CNN, where ~~strikethrough text~~ implies the span is deleted as judged by the plausibility and salience models. The base sentences before applying compression are derived from CUPS_{EXT}, the sentence extractive model.

competitive results on CNN/DM. In addition, Table 3 includes representative summaries produced by our compressive system. The summaries are highly compressive: spans not contributing to the main event or story are deleted, while maintaining grammaticality and factuality.

Our compression module can also improve over other off-the-shelf extractive models. The pipelined nature of our approach allows us to replace the current BERTSum (Liu and Lapata, 2019) extractor with any arbitrary, black-box model that retrieves important sentences. We apply our compression module on system outputs from MatchSum (Zhong et al., 2020), the current state-of-the-art extractive model, and also see gains in this setting with no additional modification to the system.

6.2 Plausibility Study

Given that our system achieves high ROUGE, we now investigate whether its compressed sentences are grammatical and factual. The plausibility model is responsible for modeling these phenomena, as defined in Section 2.1, thus we analyze its compression decisions in detail. Specifically, we run the plausibility model on 50 summaries from each of CNN and Reddit, and have annotators judge whether the predicted plausible compressions are grammatical and factual with respect to the original sentence.[10] By nature, this evaluates the *precision* of span-based deletions.

Because the plausibility model uses candidate spans from the high-recall compression rules (defined in Section 2.3), we compare our plausibility model against the baseline consisting of simply the spans identified by these rules. The results

[10]See Appendix D for further information on the annotation task and agreement scores.

	CNN		Reddit	
System	G	F	G	F
Compression Rules	87.9	75.7	73.5	60.8
+ Plausibility Model	**96.0**	**89.7**	**93.1**	**66.7**

Table 4: Human evaluation of grammaticality (G) and factuality (F) of summaries, comparing the precision of span deletions from our compression rules (§2.3) before and after applying the plausibility model (§2.1).

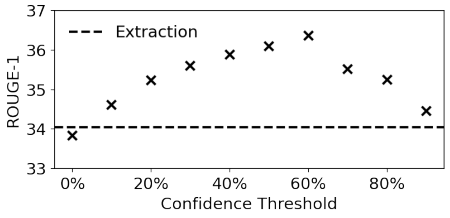

Figure 3: Varying the salience threshold $\lambda_S \in [0, 1)$ (depicted as % confidence) and its impact on ROUGE upon deleting spans $Z_P \cap Z_S$.

are shown in Table 4. On both CNN and Reddit, the plausibility model's deletions are highly grammatical, and we also see evidence that the plausibility model makes more semantically-informed deletions to maintain factuality, especially on CNN.

Factuality performance is lower on Reddit, but incorporating the plausibility model on top of the compression rules results in a 6% gain in precision. There is still, however, a large gap between factuality in this setting and factuality on CNN, which we suspect is because Reddit summaries are different in style and structure than CNN summaries: they largely consist of short event narratives (Kim et al., 2019), and so annotators may disagree on the degree to which deleting spans such as subordinate clauses impact the meaning of the events described.

Type	Model	NYT → CNN R1	NYT → CNN R2	NYT → CNN RL	CNN → Reddit R1	CNN → Reddit R2	CNN → Reddit RL	XSum → WikiHow R1	XSum → WikiHow R2	XSum → WikiHow RL	Average R1	Average R2	Average RL
In-Domain													
ext	Lead-k	29.80	11.40	26.45	19.64	2.40	14.79	24.96	5.83	23.23	24.80	6.54	21.49
ext	CUPS$_{\text{EXT}}$	**33.12**	**13.88**	**29.51**	**24.42**	**6.10**	**19.57**	**30.94**	**9.06**	**28.81**	**29.49**	**9.68**	**25.96**
Out-of-Domain													
ext	CUPS$_{\text{EXT}}$	31.05	12.46	27.64	21.32	4.54	17.08	28.32	7.54	26.35	26.90	12.27	23.69
	+ Fine-Tune (500)	31.90	13.04	28.42	23.76	5.66	18.95	29.44	8.25	27.41	28.37	8.98	24.93
cmp	CUPS	31.98	12.77	28.53	22.25	4.82	17.94	29.17	7.65	27.28	27.80	8.41	24.59
	+ Fine-Tune (500)	**33.98**	**13.25**	**30.39**	**25.01**	**5.96**	**20.10**	**30.52**	**8.44**	**28.48**	**29.84**	**9.22**	**26.32**

Table 5: **Results on out-of-domain transfer tasks.** Fine-tuning results are averaged across 5 runs, each with a random batch of 500 target domain samples. Variance among these runs is very low; see Appendix H.

6.3 Compression Analysis

The experiments above demonstrate the plausibility model generally selects spans that, if deleted, preserve grammaticality and factuality. In this section, we dive deeper into how the plausibility and salience models work together in the final trained summary model, presenting evidence of typical compression patterns. We analyze (1) our default system CUPS, which deletes spans $Z_{\text{P}} \cap Z_{\text{S}}$; and (2) a variant CUPS-NoPl (*without* plausibility but *with* salience), which only deletes spans Z_{S}, to specifically understand what compressions the salience model makes without the plausibility model's guardrails. Using 100 randomly sampled documents from CNN, we conduct a series of experiments detailed below.

On average, per sentence, 16% of candidate spans deleted by the salience model alone are not plausible. For each sentence, our system exposes a list of spans for deletion, denoted by $Z_{\text{P}} \cap Z_{\text{S}}$ and Z_{S} for CUPS and CUPS-NoPl, respectively. Because Z_{S} is identical across both variants, we can compute the plausibility model's *rejection rate* (16%), defined as $|Z_{\text{S}} \cap Z_{\text{P}}^{C}|/|Z_{\text{S}}|$. Put another way, how many compressions does the plausibility model *reject* if partnered with the salience model? On average, per sentence, the plausibility model rejects 16% of spans approved by the salience model alone, so it does non-trivial filtering of the compressions. We observe a drop in the token-level compression ratio, from 26% in CUPS to 24% in CUPS-NoPl, which is partially a result of this. From a ROUGE-1/2 standpoint, the slight reduction in compression yields a peculiar effect: on this subset of summaries, CUPS achieves 36.23/14.61 while CUPS-NoPl achieves 36.1/14.79, demonstrating the plausibility model trades off some salient deletions (-R1) for overall grammaticality (+R2) (Paulus et al., 2018).

Using salience to discriminate between plausible spans increases ROUGE. With CUPS, we perform a line search on $\lambda_{\text{S}} \in [0, 1)$, which controls the confidence threshold for deleting non-salient spans as described in Section 4.4.[11] Figure 3 shows ROUGE-1 across multiple salience cutoffs. When $\lambda_{\text{S}} = 0$, all plausible spans are deleted; in terms of ROUGE, this setting underperforms the extractive baseline, indicating we end up deleting spans that contain pertinent information. In contrast, at the peak when $\lambda_{\text{S}} = 0.6$, we delete non-salient spans with at least 60% confidence, and obtain considerably better ROUGE. These results indicate that the spans selected by the plausibility model are fundamentally good, but the ability to weigh the content relevance of these spans is critical to end-task performance.

7 Out-of-Domain Experiments

Additionally, we examine the cross-domain generalizability of our compressive summarization system. We set up three source → target transfer tasks guided by real-world settings: (1) NYT → CNN (one newswire outlet to another), (2) CNN → Reddit (newswire to social media, a low-resource domain), and (3) XSum → WikiHow (single to multiple sentence summaries with heavy paraphrasing).

For each transfer task, we experiment with two types of settings: (1) **zero-shot** transfer, where our system with parameters $[\theta_{\text{E}}; \theta_{\text{P}}; \theta_{\text{S}}]$ is directly evaluated on the target test set; and (2) **fine-tuned** transfer, where $[\theta_{\text{E}}; \theta_{\text{S}}]$ are fine-tuned with 500 target

[11] Our assumption is that posterior probabilities are calibrated, which holds true for various pre-trained Transformers across a range of tasks (Desai and Durrett, 2020).

samples, then the resulting system with parameters $[\theta'_E; \theta_P; \theta'_S]$ is evaluated on the target test set. As defined in Section 2.1, plausibility is a domain-independent notion, thus we do not fine-tune θ_P.

Table 5 shows the results. Our system maintains strong zero-shot out-of-domain performance despite distribution shifts: extraction outperforms the lead-k baseline, and compression adds roughly +1 ROUGE-1. This increase is largely due to compression improving ROUGE precision: extraction is adept at retrieving content-heavy sentences with high recall, and compression helps focus on salient content within those sentences.

More importantly, we see that **performance via fine-tuning on 500 samples matches or exceeds in-domain extraction ROUGE.** On NYT $\rightarrow$ CNN and CNN $\rightarrow$ Reddit, our system outperforms in-domain extraction baselines (trained on tens of thousands of examples), and on XSum $\rightarrow$ WikiHow, it comes within 0.3 in-domain average ROUGE. These results suggest that our system could be applied widely by crowdsourcing a relatively small number of summaries in a new domain.

8 Related Work

Compressive Summarization. Our work follows in a line of systems that use auxiliary training data or objectives to learn sentence compression (Martins and Smith, 2009; Woodsend and Lapata, 2012; Qian and Liu, 2013). Unlike these past approaches, our compression system uses *both* a plausibility model optimized for grammaticality and a salience model optimized for ROUGE. Almeida and Martins (2013) leverage such modules and learn them jointly in a multi-task learning setup, but face an intractable inference problem in their model which needs sophisticated approximations. Our approach, by contrast, does not need such approximations or expensive inference machinery like ILP solvers (Martins and Smith, 2009; Berg-Kirkpatrick et al., 2011; Durrett et al., 2016). The highly decoupled nature of our pipelined compressive system is an advantage in terms of training simplicity: we use only simple MLE-based objectives for extraction and compression, as opposed to recent compressive methods that use joint training (Xu and Durrett, 2019; Mendes et al., 2019) or reinforcement learning (Zhang et al., 2018). Moreover, we demonstrate our compression module can stack with state-of-the-art sentence extraction models, achieving additional gains in ROUGE.

One significant line of prior work in compressive summarization relies on heavily engineered rules for syntactic compression (Berg-Kirkpatrick et al., 2011; Li et al., 2014; Wang et al., 2013; Xu and Durrett, 2019). By relying on our data-driven objectives to ultimately perform compression, our approach can rely on a leaner, much more minimal set of constituency rules to extract candidate spans.

Gehrmann et al. (2018) also extract sub-sentential spans in a "bottom-up" fashion, but their method does not incorporate grammaticality and only works best with an abstractive model; thus, we do not compare to it in this work.

Discourse-based Compression. Recent work also demonstrates elementary discourse units (EDUs), spans of sub-sentential clauses, capture salient content more effectively than entire sentences (Hirao et al., 2013; Li et al., 2016; Durrett et al., 2016; Xu et al., 2020). Our approach is significantly more flexible because it does not rely on an a priori chunking of a sentence, but instead can delete variably sized spans based on what is contextually permissible. Furthermore, these approaches require RST discourse parsers and in some cases coreference systems (Xu et al., 2020), which are less accurate than the constituency parsers we use.

9 Conclusion

In this work, we present a compressive summarization system that decomposes span-level compression into two learnable objectives, plausibility and salience, on top of a minimal set of rules derived from a constituency tree. Experiments across both in-domain and out-of-domain settings demonstrate our approach outperforms strong extractive baselines while creating well-formed summaries.

Acknowledgments

This work was partially supported by NSF Grant IIS-1814522, NSF Grant SHF-1762299, a gift from Salesforce Inc., and an equipment grant from NVIDIA. The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources used to conduct this research. Results presented in this paper were obtained using the Chameleon testbed supported by the National Science Foundation. Thanks as well to the anonymous reviewers for their helpful comments.

References

Miguel Almeida and André Martins. 2013. Fast and Robust Compressive Summarization with Dual Decomposition and Multi-Task Learning. In *Proceedings of the Annual Meeting of the Association for Computational Lingusitics (ACL)*.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly Learning to Extract and Compress. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Noam Chomsky. 1956. Syntactic Structures.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Shrey Desai and Greg Durrett. 2020. Calibration of Pre-trained Transformers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A Question Answering Evaluation Framework for Faithfulness Assessment in Abstractive Summarization. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.

Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. 2016. Learning-Based Single-Document Summarization with Compression and Anaphoricity Constraints. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Katja Filippova and Yasemin Altun. 2013. Overcoming the Lack of Parallel Data in Sentence Compression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. 2018. Bottom-Up Abstractive Summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Dan Gillick and Benoit Favre. 2009. A Scalable Global Model for Summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing (ILP for NLP)*.

Tanya Goyal and Greg Durrett. 2020. Evaluating Factuality in Generation with Dependency-level Entailment. In *Findings of the Conference on Empirical Methods in Natural Language Processing (Findings of EMNLP)*.

Karl Moritz Hermann, Tomás Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*.

Tsutomu Hirao, Yasuhisa Yoshida, Masaaki Nishino, Norihito Yasuda, and Masaaki Nagata. 2013. Single-Document Summarization as a Tree Knapsack Problem. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive Summarization of Reddit Posts with Multi-level Memory Networks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Nikita Kitaev and Dan Klein. 2018. Constituency Parsing with a Self-Attentive Encoder. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Kevin Knight and Daniel Marcu. 2000. Statistics-Based Summarization—Step One: Sentence Compression. In *Proceedings of the National Conference on Artificial Intelligence (AAAI) and Conference on Innovative Applications of Artificial Intelligence (IAAI)*.

Kevin Knight and Daniel Marcu. 2002. Summarization beyond Sentence Extraction: A Probabilistic Approach to Sentence Compression. *Artificial Intelligence*.

Mahnaz Koupaee and William Yang Wang. 2018. WikiHow: A Large Scale Text Summarization Dataset. *arXiv preprint arXiv:1810.09305*.

Klaus Krippendorff. 1980. *Content Analysis: An Introduction to Its Methodology*. Sage.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the Factual Consistency of Abstractive Text Summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end Neural Coreference Resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Chen Li, Yang Liu, Fei Liu, Lin Zhao, and Fuliang Weng. 2014. Improving Multi-documents Summarization by Sentence Compression based on Expanded Constituent Parse Trees. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Junyi Jessy Li, Kapil Thadani, and Amanda Stent. 2016. The Role of Discourse Units in Near-Extractive

Summarization. In *Proceedings of the Annual Meeting of the Special Interest Group on Discourse and Dialogue*.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yang Liu and Mirella Lapata. 2019. Text Summarization with Pretrained Encoders. In *Proceedings of the Conference on Empirical Methods of Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Yang Liu, Ivan Titov, and Mirella Lapata. 2019. Single Document Summarization as Tree Induction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

André Martins and Noah A. Smith. 2009. Summarization with a Joint Model for Sentence Extraction and Compression. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing (ILP for NLP)*.

Afonso Mendes, Shashi Narayan, Sebastião Miranda, Zita Marinho André F. T. Martins, and Shay B. Cohen. 2019. Jointly Extracting and Compressing Documents with Summary State Representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A Deep Reinforced Model for Abstractive Summarization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Xian Qian and Yang Liu. 2013. Fast Joint Compression and Summarization via Graph Cuts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Evan Sandhaus. 2008. The New York Times Annotated Corpus.

Carson Schütze. 1996. The Empirical Base of Linguistics: Grammaticality Judgments and Linguistic Methodology.

Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A. Smith. 2018. Syntactic Scaffolds for Semantic Structures. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and Answering Questions to Evaluate the Factual Consistency of Summaries. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.

Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2013. A Sentence Compression Based Framework to Query-Focused Multi-Document Summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv preprint arXiv:1910.03771*.

Kristian Woodsend and Mirella Lapata. 2012. Multiple Aspect Summarization Using Integer Linear Programming. *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing (EMNLP) and Computational Natural Language Learning (CoNLL)*.

Jiacheng Xu and Greg Durrett. 2019. Neural Extractive Text Summarization with Syntactic Compression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-Aware Neural Extractive Text Summarization. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020a. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. *Proceedings of the International Conference on Machine Learning (ICML)*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Weinberger, and Yoav Artzi. 2020b. BERTScore: Evaluating Text Generation with BERT. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou. 2018. Neural Latent Extractive Document Summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive Summarization as Text Matching. In *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*.

A Summarization Datasets

Table 1 lists training, development, and test splits for each dataset used in our experiments.

Dataset	k	Train	Dev	Test
CNN/Daily Mail	3	287,084	13,367	11,489
CNN	3	90,266	1,220	1,093
New York Times	3	137,772	17,222	17,220
XSum	2	203,028	11,273	11,332
WikiHow	4	168,126	6,000	6,000
Reddit	2	41,675	645	645

Table 1: Training, development, and test dataset sizes for CNN/Daily Mail (Hermann et al., 2015), CNN (subset of CNN/DM), New York Times (Sandhaus, 2008), XSum (Narayan et al., 2018), WikiHow (Koupaee and Wang, 2018), and Reddit (Kim et al., 2019). For each dataset, the extraction model selects the top-k sentences to form the basis of the compressive summary.

B Training Details

Table 2 details the hyperparameters for training the extraction and compression models. These hyperparameters largely borrowed from previous work (Devlin et al., 2019), and we do not perform any additional grid searches in the interest of simplicity. The pre-trained encoders are set to either `bert-base-uncased` or `google/electra-base-discriminator` from HuggingFace Transformers (Wolf et al., 2019). Following previous work (Liu et al., 2019; Zhong et al., 2020), we use the best performing model among the top three validation checkpoints.

C Inference Details

Our system uses two hyperparameters at test-time to control the level of compression performed by the plausibility and salience models. Table 3 shows the BERT- and ELECTRA-based system hyperparameters, respectively. We sweep the salience model threshold $\lambda_S \in [0.1, 0.9]$ with a granularity of 0.05; across all datasets used in the in-domain experiments (CNN/DM, CNN, WikiHow, XSum, and Reddit), this process takes roughly 8 hours on a 32GB NVIDIA V100 GPU.

Furthermore, there are certain cases where the syntactic compression rules license deleting a chain of constituents rather than individual ones. A common example of this is in conjoined noun phrases (NP_1–[CC–NP_2]) where if the second noun phrase NP_2 is deleted, its preceding coordinating conjunction CC should also be deleted. To avoid changing the compression model substantially, we relegate secondary deletions to a postprocessing step, where if a primary constituent like NP_2 is deleted at test-time, its secondary constituents are also automatically deleted.

Hyperparameter	Extraction	Compression
Train Steps	10,000	10,000
Eval Steps	1,000	1,000
Eval Interval	1,000	1,000
Batch Size	16	16
Learning Rate	1e-5	1e-5
Optimizer	AdamW	AdamW
Weight Decay	0	0
Gradient Clip	1.0	1.0
Max Sequence Length	512	256
Max Spans	—	50

Table 2: Training hyperparameters for the extraction and compression models (§3).

Encoder	CNN/DM	CNN	WikiHow	XSum	Reddit
Hyperparameter: Plausibility (λ_P)					
BERT	0.6	0.6	0.6	0.6	0.6
ELECTRA	0.6	0.6	0.6	0.6	0.6
Hyperparameter: Salience (λ_S)					
BERT	0.7	0.5	0.4	0.55	0.65
ELECTRA	0.7	0.5	0.45	0.6	0.7

Table 3: BERT- and ELECTRA-based system hyperparameters for the plausibility (§2.1) and salience models (§2.2). We fix the plausibility threshold at 0.6 and only optimize the salience thresold.

D Plausibility Study

Study	CNN	Reddit
Grammaticality	0.24	0.17
Factuality	0.28	0.34

Table 4: Annotator agreement for grammaticality and factuality studies on CNN and Reddit. Values displayed are computed using Krippendorff's α (Krippendorff, 1980).

We conduct our human evaluation on Amazon Mechanical Turk, and set up the following requirements: annotators must (1) reside in the US; (2) have a HIT acceptance rate $\geq 95\%$; and (3) complete at least 50 HITs prior to this one. Each HIT comes with detailed instructions (including a set of representative examples) and 6 assignments. One of these assignments is a randomly chosen example from the instructions (the *challenge* question), and the other five are samples we use in our actual

		CNN/DM			CNN			WikiHow			XSum			Reddit		
Type	Model	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
ext	CUPS_{EXT}	43.16	20.10	39.52	32.41	13.59	28.93	30.45	8.74	28.34	23.59	4.55	17.81	23.87	5.84	19.27
cmp	CUPS	**43.55**	**20.11**	**39.93**	**34.54**	**13.67**	**31.00**	**31.98**	**8.95**	**29.88**	**25.59**	**4.93**	**19.67**	**25.24**	**6.12**	**20.60**

Table 5: Results on CNN/DM, CNN, WikiHow, XSum, and Reddit with initializing the pre-trained encoders in CUPS to $\text{BERT}_{\text{BASE}}$ as opposed to $\text{ELECTRA}_{\text{BASE}}$.

		WikiHow			XSum			Reddit		
Type	Model	R1	R2	RL	R1	R2	RL	R1	R2	RL
cmp	CUPS	32.43	**9.44**	30.24	26.04	**5.36**	**19.90**	25.99	6.57	21.08
cmp	MatchSum + CUPS_{CMP}	**32.83**	9.24	**30.53**	**26.42**	5.09	19.76	**26.60**	**6.60**	**21.43**

Table 6: Results on WikiHow, XSum, and Reddit with replacing CUPS_{EXT} with MatchSum (Zhong et al., 2020), a state-of-the-art extractive model.

study. In each assignment, annotators are presented with the original sentence and a candidate span, and asked if deleting the span negatively impacts the grammaticality and factuality of the resulting, compressed sentence. Each annotator is paid 50 cents upon completing the HIT; this pay rate was calibrated to pay roughly $10/hour.

After all assignments are completed, we filter low-quality annotators according to two heuristics. An annotator is removed if he/she completes the assignment in under 60 seconds or answers the challenge question incorrectly. We see a substantial increase in agreement for both the grammaticality and factuality studies among the remaining annotators. The absolute agreement scores, as measured by Krippendorff's α (Krippendorff, 1980), are shown in Table 4. Consistent with prior grammaticality evaluations in summarization (Xu and Durrett, 2019; Xu et al., 2020), agreement scores are objectively low due to the difficulty of the tasks, thus we compare the annotations with expert judgements. An expert annotator (one of the authors of this paper uninvolved with the development of the plausibility model) performed the CNN annotation task; we find, by using the majority vote among the crowdsourced annotations, the regular and expert annotators concur 80% of the time on grammaticality and 60% of the time on factuality; this establishes a higher degree of confidence in the crowdsourced annotations when aggregated.

E System Results with BERT

Table 5 (CNN/DM, CNN, WikiHow, XSum, Reddit) shows results using $\text{BERT}_{\text{BASE}}$ as the pre-trained encoder. While the absolute ROUGE results with $\text{BERT}_{\text{BASE}}$ are lower than with $\text{ELECTRA}_{\text{BASE}}$, we still see a large improvement compared to the sentence extractive baseline.

F Extended MatchSum Results

On WikiHow, XSum, and Reddit, we additionally experiment with replacing the sentences extracted from CUPS_{EXT} with MatchSum (Zhong et al., 2020) system outputs. From the results (see Table 6), we see that our system with MatchSum extraction achieves the most gains on Reddit, but its average performance on WikiHow and XSum is more comparable to the standard CUPS system.

G Plausibility Ablation

Table 7 shows results on CNN, WikiHow, XSum, and Reddit with removing the plausibility model in CUPS_{CMP}. Consistent with the analysis in Section 6.3, we see the plausibility model is primarily responsible for gains in ROUGE-2, but in its absence, the salience model can delete arbitrary spans, resulting in gains in ROUGE-1 and ROUGE-L. This ablation demonstrates the need to analyze summaries outside of ROUGE since notions of grammaticality and factuality cannot easily be ascertained by computing lexical overlap with a reference summary.

H Out-of-Domain Results

In Tables 8, 9, and 10, we show ROUGE results with standard deviations across 5 independent runs, for the fine-tuning experiments on NYT $\rightarrow$ CNN, CNN $\rightarrow$ Reddit, and XSum $\rightarrow$ WikiHow, respectively. Despite fine-tuning with a random batch of 500 samples each time, we consistently see low

		CNN			WikiHow			XSum			Reddit		
Type	Model	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
cmp	CUPS	35.22	**14.19**	31.51	32.43	**9.44**	30.24	26.04	**5.36**	19.90	25.99	**6.57**	21.08
	- Plausibility	**35.29**	14.03	**31.63**	**32.54**	9.34	**30.36**	**26.36**	5.35	**20.19**	**26.11**	6.56	**21.19**

Table 7: Results on CNN, WikiHow, XSum, and Reddit with removing the plausibility model in CUPS_{CMP}.

variance across the runs, demonstrating our system does not have an affinity towards particular samples in an out-of-domain setting.

Furthermore, we present an ablation of salience for the aforementioned transfer tasks in Table 11. On NYT $\rightarrow$ CNN, salience only helps increase ROUGE-L, but we see consistent increases in average ROUGE on CNN $\rightarrow$ Reddit and XSum $\rightarrow$ WikiHow. We can expect larger gains by fine-tuning salience on more samples, but even with 500 out-of-domain samples, our compression module benefits from the inclusion of the salience model.

I Reproducibility

Table 12 shows system results on the development sets of CNN/DM, CNN, WikiHow, XSum, and Reddit to aid the reproducibility of our system; both CUPS_{EXT} and CUPS are included. Furthermore, in Table 13, we report several metrics to aid the training of the extraction and compression models. These specific metrics recorded by training models on a 32GB NVIDIA V100 GPU with the hyperparameters listed in Table 2.

		NYT → CNN		
Type	Model	R1 (std)	R2 (std)	RL (std)
ext	CUPS_{EXT}	33.74 (0.08)	13.19 (0.11)	30.46 (0.11)
cmp	CUPS	33.98 (0.06)	13.25 (0.11)	30.39 (0.07)

Table 8: Results on NYT → CNN, reporting ROUGE with standard deviation across 5 independent runs with a random batch of 500 samples.

		CNN → Reddit		
Type	Model	R1 (std)	R2 (std)	RL (std)
ext	CUPS_{EXT}	24.30 (0.20)	5.78 (0.08)	19.87 (0.11)
cmp	CUPS	25.01 (0.15)	5.96 (0.08)	20.10 (0.09)

Table 9: Results on CNN → Reddit, reporting ROUGE with standard deviation across 5 independent runs with a random batch of 500 samples.

		XSum → WikiHow		
Type	Model	R1 (std)	R2 (std)	RL (std)
ext	CUPS_{EXT}	30.22 (0.05)	8.43 (0.03)	28.30 (0.03)
cmp	CUPS	30.52 (0.06)	8.44 (0.01)	28.48 (0.04)

Table 10: Results on XSum → WikiHow, reporting ROUGE with standard deviation across 5 independent runs with a random batch of 500 samples.

		NYT → CNN			CNN → Reddit			XSum → WikiHow		
Type	Model	R1	R2	RL	R1	R2	RL	R1	R2	RL
ext	CUPS_{EXT}	31.90	13.04	28.42	23.76	5.66	18.95	29.44	8.25	27.41
cmp	CUPS	**33.98**	**13.25**	30.39	**25.01**	**5.96**	**20.10**	**30.52**	**8.44**	**28.48**
	- Salience	33.74	13.19	**30.46**	24.30	5.78	19.87	30.22	8.43	28.30

Table 11: Results on NYT → CNN, CNN → Reddit, and XSum → WikiHow after removing the salience model.

		CNN/DM			CNN			WikiHow			XSum			Reddit		
Type	Model	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
Encoder: BERT																
ext	CUPS_{EXT}	43.37	20.50	39.86	31.85	12.98	28.53	30.20	8.58	28.07	23.67	4.52	17.89	24.20	5.78	18.77
cmp	CUPS	**43.68**	**20.51**	**40.16**	**34.26**	**13.63**	**30.93**	**31.55**	**8.95**	**29.42**	**25.37**	**4.93**	**19.44**	**25.51**	**6.17**	**19.96**
Encoder: ELECTRA																
ext	CUPS_{EXT}	43.97	21.03	40.45	32.50	13.40	29.09	30.75	8.90	28.57	24.44	5.03	18.48	25.09	6.40	19.42
cmp	CUPS	**44.35**	**21.07**	**40.81**	**34.87**	**13.89**	**31.35**	**32.20**	**9.34**	**30.01**	**26.24**	**5.47**	**20.06**	**26.73**	**6.90**	**20.84**

Table 12: Results on the development sets of CNN/DM, CNN, WikiHow, XSum, and Reddit using the default CUPS system, leveraging both $\text{BERT}_{\text{BASE}}$ and $\text{ELECTRA}_{\text{BASE}}$ pre-trained encoders.

Metrics	CNN/DM	CNN	NYT	WikiHow	XSum	Reddit	Google
Model: Extraction							
Train Steps	22K	15K	18K	23K	24K	10K	—
Time Elapsed (hrs/min)	6h 48m	3h 4m	5h 52m	5h 5m	6h 6m	1h 59m	—
Model: Compression							
Train Steps	26K	13K	19K	25K	25K	10K	20K
Time Elapsed (hrs/min)	3h 32m	1h 27m	2h 38m	3h 26m	3h 38m	0h 56m	1h 59m

Table 13: Number of training steps and total time elapsed for training extraction and compression models on CNN/DM, CNN, NYT, WikiHow, XSum, Reddit, and Google*. Models are benchmarked on a 32GB NVIDIA V100 GPU. *Google refers to the sentence compression dataset released by Filippova and Altun (2013), which is only used to train the plausibility compression model.