Learning to Fuse Sentences with Transformers for Abstractive Summarization

Abstract

It is important for a summarizer to acquire the skill of sentence fusion, which allows two or more sentences to be combined into one effective sentence. The ability to fuse sentences is attractive as it is a key step to produce succinct and reliable abstracts. However, abstractive summarizers to date have tended to fail on sentence fusion. These summarizers produce few summary sentences by fusion, or even worse, generate incorrect fusions that lead the summary to fail to retain the original meaning. In this paper, we investigate the ability of Transformers to fuse sentences and propose novel algorithms to enhance their ability to perform sentence fusion by leveraging the knowledge of *points of correspondence* between sentences. Further, we introduce a novel dataset to support modeling and evaluation of sentence fusion, covering a number of types of correspondence provided by the theory of text cohesion. Through extensive experiments we investigate the effects of design choices on Transformer model performance. Our findings shed light on the importance of modeling points of correspondence between sentences for effective sentence fusion.

1 Introduction

Abstraction can only be achieved if a summarizer acquires the skill of sentence fusion [Barzilay and McKeown, 2005], which allows two or more sentences to be combined into one complex sentence. Despite its importance, abstractive summarizers to date have tended to fail on sentence fusion. As illustrated in Table 1, a summarizer combines two sentences with inappropriate choices of *points of correspondence* (PoC), resulting in ungrammatical or nonsensical output that confuses the reader. Halliday and Hasan's theory of text cohesion [1976] divides PoC into several types, including referencing, ellipsis, substitution, and others. Such cohesive devices are essential to tie two sentences together into a coherent text. Without a robust ability to fuse sentences based on appropriate PoC, an abstractive summarizer can render itself unusable in real-world scenarios or produce erroneous summaries that fail to retain the meaning of the original text. We thus seek to enhance the summarizer's ability to perform sentence fusion.

Source	The kind of horror represented by the Blackwater case and others like it [...] may be largely absent from public memory in the West these days, but it is being *used by the Islamic State in Iraq and Syria (ISIS)* to support its sectarian narrative. *In its propaganda, ISIS has been using* Abu Ghraib and other cases of Western abuse to legitimize its current actions [...]
Summary	*In its propaganda, ISIS is being used by the Islamic State in Iraq and Syria.*
Source	*The purpose of the lengthy project is to recreate the conditions that existed moments after the "Big Bang" – the scientific* theory said to explain the creation of the universe. By replicating the energy density and temperature, *scientists hope to uncover how the universe evolved.* [...]
Summary	*The purpose of the project is to recreate the conditions that existed moments after the "Big Bang" scientists hope to uncover how the universe evolved.*

Table 1: Summary sentences generated by neural abstractive summarizers. In both examples, the summarizer forces two sentences to be merged into a single summary sentence with improper use of *points of correspondence* between sentences, yielding ungrammatical and nonsensical output. Summaries are manually re-cased for readability.

A renewed emphasis must be placed on sentence fusion in the context of neural abstractive summarization. A majority of the systems are trained end-to-end [See *et al.*, 2017; Paulus *et al.*, 2017; Narayan *et al.*, 2018; Chen and Bansal, 2018; Gehrmann *et al.*, 2018; Liu and Lapata, 2019]. An abstractive summarizer is rewarded for generating summaries that contain the same words as human abstracts, measured by automatic metrics such as ROUGE [Lin, 2004]. A summarizer, however, is not rewarded for correctly fusing sentences. In fact, when examined more closely, only few sentences in system abstracts are generated by fusion [Falke *et al.*, 2019; Lebanoff *et al.*, 2019]. For instance, 6% of summary sentences generated by Pointer-Gen [See *et al.*, 2017] are through fusion, whereas human abstracts contain 32% fusion sentences. Moreover, sentences generated by fusion are prone to errors. They can be ungrammatical, nonsensical, or otherwise ill-formed. There is thus an urgent need to develop neural abstractive summarizers to fuse sentences properly.

The importance of sentence fusion has long been recognized by the community before the era of neural text summarization. The pioneering work of Barzilay et al. [1999] introduces an information fusion algorithm that combines similar elements across related text to generate a succinct summary. Later work,

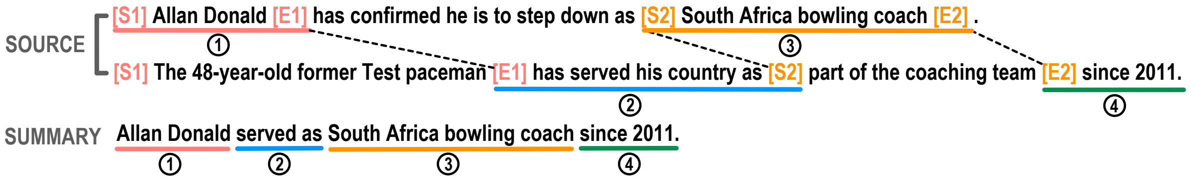

SOURCE
[S1] Allan Donald [E1] has confirmed he is to step down as [S2] South Africa bowling coach [E2] .
①
③
[S1] The 48-year-old former Test paceman [E1] has served his country as [S2] part of the coaching team [E2] since 2011.
②
④
SUMMARY
Allan Donald served as South Africa bowling coach since 2011.
① ② ③ ④

Figure 1: Sentence fusion involves determining what content from each sentence to retain, and how best to weave the text pieces together into a well-formed summary sentence. Points of correspondence (PoC) between sentences are text chunks that convey the same or similar meanings, e.g., *Allan Donald* and *The 48-year-old former Test paceman*, *South Africa bowling coach* and *part of the coaching team*. Recognizing points of correspondence allows a sentence fusion system to indiscriminately treat these text chunks during summary generation.

such as [Marsi and Krahmer, 2005; Filippova and Strube, 2008; Elsner and Santhanam, 2011; Thadani and McKeown, 2013; Mehdad *et al.*, 2013], builds a dependency or word graph by combining syntactic trees of similar sentences, then employs integer linear programming to decode a summary sentence from the graph. Most of these studies have assumed a set of *similar* sentences as input, where fusion is necessary to reduce repetition and increase fluency. However, humans do not limit themselves to combine similar sentences. In this paper, we pay particular attention to fuse *disparate* sentences that contain fundamentally different content but remain related to make fusion sensible [Elsner and Santhanam, 2011].

We address the challenge of fusing disparate sentences by enhancing the Transformer architecture [Vaswani *et al.*, 2017] with *points of correspondence* between sentences, thereby guiding the fusion process with Halliday and Hasan's theory of text cohesion [1976]. The task of sentence fusion involves choosing content from each sentence and weaving the content pieces together into an output sentence that is linguistically plausible and semantically truthful to the original input. It is distinct from prior studies [Geva *et al.*, 2019] that connect two sentences with discourse markers. More importantly, we introduce a new sentence fusion dataset with PoC annotations as test bed for our experiments, that serves as a basis for future research to measure the success of sentence fusion systems. We summarize the contributions of our work as follows.

- We make crucial use of *points of correspondence* (PoC) between sentences for information fusion. Our use of PoC was initiated by the current lack of understanding of how sentences are combined in neural text summarization. We enrich Transformers with PoC information and investigate the effect of various design choices on model performance. Our findings shed light on the importance of modeling points of correspondence for effective sentence fusion.
- We present a new dataset comprised of pairs of sentences drawn from 1,174 documents and their fusions. The data are annotated for inter-sentence points of correspondence, i.e., text expressions that convey the same or similar meanings. We use the insights gained from this study to inform the design of sentence fusion systems that combine two sentences into one with an awareness of sentence correspondences. Our data and annotations will be made publicly available.

2 Related Work

Widespread adoption of abstractive summarization techniques remains hampered by the lack of trust associated with system abstracts [Kryściński *et al.*, 2019]. The distrust originates in part from a summarizer's inability to perform sentence fusion [Falke *et al.*, 2019], where systems may produce ill-formed fusion sentences that fail to retain the original meaning [See *et al.*, 2017; Fan *et al.*, 2018; Celikyilmaz *et al.*, 2018; Sharma *et al.*, 2019]. Recent years have seen a surge of interest in unsupervised pretraining for abstractive summarization [Liu and Lapata, 2019; Dong *et al.*, 2019; Raffel *et al.*, 2019; Lewis *et al.*, 2019]. However, without an understanding of the mechanisms underlying sentence fusion, modern abstractive summarizers cannot reliably generate abstracts, making it hard to deploy them in various real-world applications.

An essential and non-trivial task, sentence fusion allows two or more sentences to be combined into one succinct sentence. It involves choosing what content from each sentence to retain and how best to weave the content pieces together into a succinct output sentence. Traditional methods identify salient text units from source sentences, arrange them in a compact form, such as a word or dependency graph, and subsequently synthesize a natural language sentence from it [Barzilay *et al.*, 1999; Filippova and Strube, 2008; Thadani and McKeown, 2013]. Nonetheless, it is not straightforward to compare methods of sentence fusion due to the lack of freely and publicly available benchmarks. In this paper we address the problem by introducing a new labelled dataset along with algorithms for sentence fusion, which has the added benefit of advancing abstractive summarization.

3 Method

Not all pairs of sentences are fusable. There must exist some form of grammatical or lexical linking that holds the sentences together for them to be felicitously combined, as stipulated by Halliday and Hasan's theory of text cohesion [1976]. We investigate lexical linking in this paper, which is characterized by a set of *points of correspondence* (PoC) between sentences. A PoC corresponds to a pair of text chunks that express the same or similar meanings. E.g., *Allan Donald* vs. *The 48-year-old former Test paceman*, *South Africa bowling coach* vs. *part of the coaching team*, as shown in Figure 1 (PoC types will be elaborated in §4). Use of alternative expressions for conveying the same meanings is standard practice in writing, as it increases lexical variety and reduces redundancy. However, if a summarizer cannot make effective use of these expressions to establish correspondence between sentences, it may produce ungrammatical and nonsensical outputs.

Let $\mathbf{x}$=$\{x_1, \ldots, x_{|\mathbf{x}|}\}$ be the source sequence consisting of a concatenation of two input sentences and $\mathbf{y}$=$\{y_1, \ldots, y_{|\mathbf{y}|}\}$ the target sequence corresponding to the summary sentence. Existing sequence-to-sequence or text-to-text models [Raffel *et al.*,

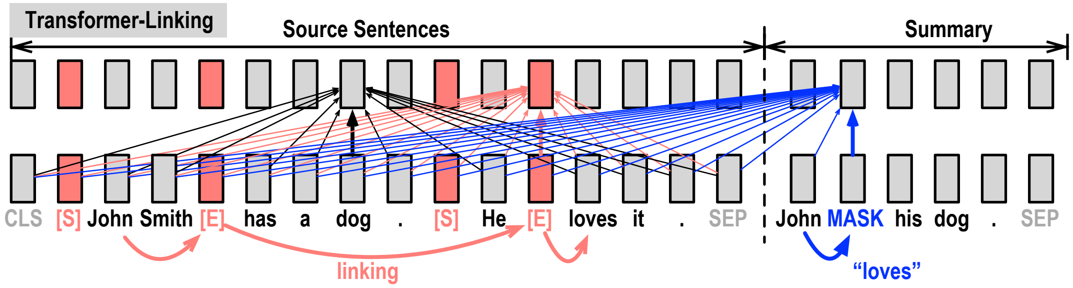


Figure 2: Our TRANS-LINKING model provides a soft mechanism to allow mentions of the same PoC to be used interchangeably in summary generation. Without linking, the model attention has to shift a long distance from "John" to "loves" to generate the next summary word, as the two words appear far apart in the source text and such long-range dependency is not always effectively captured by the model. The TRANS-LINKING model facilitates generation by reducing the shifting distance, allowing the model to hop to special tokens then to "loves."

2019] are agnostic to points of correspondence between sentences, thus fusion can happen somewhat arbitrarily between sentences with or without lexical ties. In contrast, we aim to design new sentence fusion models that *explicitly* leverage the knowledge of points of correspondence between document sentences to generate reasonable summary sentences. In what follows, we describe our fusion model in detail, which is flexible enough to permit a varying number of PoC per fusion instance; we discuss the recognition of PoC in §4.

3.1 Transformer with Linking

It is advantageous for a Transformer model to make use of PoC information for sentence fusion. PoC are alternative expressions bearing the same semantic meaning or references to the same entity. These mentions are usually used interchangeably by writers to create abstractive summaries. While Transformer-based pretrained models have had considerable success [Devlin *et al.*, 2018], they primarily feature pairwise relationships between *tokens* through the multi-head self-attention mechanism, but not PoC mentions, which are are *text expressions* of varying size. Only to a limited extent do these models embed knowledge of coreference links in their model weights [Clark *et al.*, 2019]. There is thus a growing need for incorporating PoC linkages explicitly in a Transformer model to enhance its ability to perform sentence fusion.

We propose to enrich Transformer's source sequence with markups that indicate PoC linkages. Here PoC information is assumed to be available for any fusion instance. We introduce special tokens ([S$_k$] and [E$_k$]) to mark the start and end of each PoC mention; all mentions pertaining to the k-th PoC share the same start/end tokens. An example is illustrated in Figure 1, where *Allan Donald* and *The 48-year-old former Test paceman* are enriched with the same special tokens. We expect special tokens to assist in linking coreferring mentions, creating long-range dependencies between them and encouraging the model to use these mentions interchangeably in summary generation. The model is called "TRANS-LINKING."

Our Transformer takes as input a sequence $\mathcal{S}$ formed by concatenating the source and summary sequences, where the source sequence has been enriched with PoC information. Followingpractice of Devlin et al. [2018], we append CLS and SEP to the source and SEP to the summary sequence (Eq. (1)). An embedding layer is employed to convert $\mathcal{S}$ to a sequence of embeddings, denoted by $\mathbf{H}^0 = [\mathbf{h}_1^0, \dots, \mathbf{h}_{|\mathcal{S}|}^0]$ (l=0). The Transformer then takes hidden representations from the (l-1)-th layer to construct representations of the l-th layer, $\mathbf{H}^l = [\mathbf{h}_1^l, \dots, \mathbf{h}_{|\mathcal{S}|}^l]$, using the multi-head attention mechanism.

$$\mathcal{S} = [\underbrace{\text{CLS}, x_1, \dots, x_{|\mathbf{x}|}, \text{SEP}}_{\text{Source }(\mathbf{x})}, \underbrace{y_1, \dots, y_{|\mathbf{y}|}, \text{SEP}}_{\text{Summary }(\mathbf{y})}] \tag{1}$$

An attention head transforms each hidden vector $\mathbf{h}_i^{l-1} \in \mathbb{R}^{d_h}$ into $\mathbf{q}_i, \mathbf{k}_i, \mathbf{v}_i \in \mathbb{R}^{d_k}$ respectively denoting query, key and value vectors. This is achieved by linear transformations (Eq. (2)), where $\mathbf{W}_l^Q, \mathbf{W}_l^K, \mathbf{W}_l^V \in \mathbb{R}^{d_k \times d_h}$ are model parameters. The attention weight $\alpha_{i,j}$ is computed for all pairs of tokens by taking the dot product of query and key vectors and applying softmax over the output (Eq. (3)). $\alpha_{i,j}$ indicates the importance of every other token j to constructing the hidden representation $\mathbf{h}_i^l$ of the current token i.

$$\mathbf{q}_i = \mathbf{W}_l^Q \mathbf{h}_i^{l-1} \quad \mathbf{k}_i = \mathbf{W}_l^K \mathbf{h}_i^{l-1} \quad \mathbf{v}_i = \mathbf{W}_l^V \mathbf{h}_i^{l-1} \tag{2}$$

$$\alpha_{i,j} = \frac{\exp(\mathbf{q}_i^\top \mathbf{k}_j / \sqrt{d_k} + \mathcal{M}_{i,j})}{\sum_{j'=1}^{|\mathcal{S}|} \exp(\mathbf{q}_i^\top \mathbf{k}_{j'} / \sqrt{d_k} + \mathcal{M}_{i,j'})} \tag{3}$$

Importantly, we incorporate a mask $\mathcal{M} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ to control the attention of the model (Eq. (4)). It is a square matrix where the i-th row represents the mask of the i-th token of $\mathcal{S}$. $\mathcal{M}_{i,j} = 0$ allows token i to attend to j, whereas $\mathcal{M}_{i,j} = -\infty$ prevents i from attending to j as it leads $\alpha_{i,j}$ to be zero after softmax normalization. Similar to [Dong *et al.*, 2019], a source token ($i \le |\mathbf{x}|$) is allowed to attend to all source tokens ($\mathcal{M}_{i,j} = 0$ for $j \le |\mathbf{x}|$), whereas a summary token ($i > |\mathbf{x}|$) can attend to all tokens including itself and those prior to it ($\mathcal{M}_{i,j} = 0$ for $j \le i$). The mask $\mathcal{M}$ provides desired flexibility in terms of building hidden representations for tokens in $\mathcal{S}$. The output of the attention head is a weighted sum of the value vectors $\mathbf{h}_i^l = \sum_{j=1}^{|\mathcal{S}|} \alpha_{i,j} \mathbf{v}_j$. When there are multiple attention heads, their outputs are concatenated to form $\mathbf{h}_i^l$.

$$\mathcal{M}_{i,j} = \begin{cases} 0 & \text{if } j \le \max(i, |\mathbf{x}|) \\ -\infty & \text{otherwise} \end{cases} \tag{4}$$

We opt for a decoder-only Transformer architecture rather than an encoder-decoder architecture [Raffel *et al.*, 2019] as it allows all model parameters to be warm-started, which is crucial to task success [Khandelwal *et al.*, 2019]. We fine-tune the TRANS-LINKING model on our sentence fusion dataset (§4) using a denoising objective, where 70% of the summary tokens are randomly sampled then masked out from $\mathcal{S}$. These tokens are replaced by a MASK token. The model is then trained to predict the original tokens given hidden representations of MASK tokens: $\mathbf{o} = \text{softmax}(\mathbf{W}^O \text{GeLU}(\mathbf{W}^h \mathbf{h}^L_{\text{MASK}})))$, where parameters of the output layer $\mathbf{W}^O$ are tied with token embeddings. At test time, a summary is unrolled one word at a time. This is achieved by appending MASK to the partial summary, starting from $\mathcal{S}' = [\text{CLS}, x_1, \ldots, x_{|\mathbf{x}|}, \text{SEP}, \text{MASK}]$, then predicting the next summary word given the hidden representation $\mathbf{h}^L_{\text{MASK}}$. This process continues until SEP is encountered which signals the end of the decoding process.

Our model provides a soft linking mechanism to allow mentions of the same PoC to be used interchangeably in summary generation. As illustrated in Figure 2, without PoC linking, the focus of model attention has to shift a long distance from "John" to "loves" to generate the next summary word. These two words occur far apart in the source text; their long-range dependency is not always effectively captured by the model. In contrast, our TRANS-LINKING model substantially reduces the shifting distance, allowing the model to hop to the special token "[E]" then to "loves," facilitating summary generation.

3.2 Transformer with Shared Representations

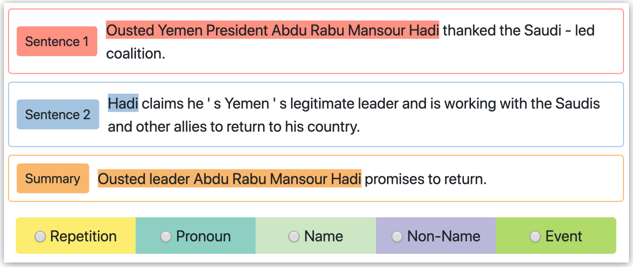
An entity may be referred to by various linguistic expressions. These expressions are related by complex morpho-syntactic, syntactic or semantic constraints [Grosz *et al.*, 1995]. We explore an alternative method to allow mentions of the same PoC to be connected with each other. Particularly, we direct one attention head of the Transformer model to focus on tokens belonging to the same PoC, allowing these tokens to share semantic representations, similar to Strubell et al. [2018].

Let $\mathbf{z}$=$\{z_1, \ldots, z_{|\mathbf{z}|}\}$ be a sequence containing PoC information. $\mathbf{z}$ has the same length as $\mathbf{x}$. $z_i \in \{0, \ldots, \mathsf{K}\}$ indicates the index of PoC to which the token $\mathbf{x}_i$ belongs. z_i=0 indicates $\mathbf{x}_i$ is not associated with any PoC. Our TRANS-SHAREREPR model selects an attention head h from the l-th layer of the Transformer model. The attention head h governs tokens that are part of any PoC ($z_i \neq 0$). In particular, the hidden representation $\mathbf{h}^l_i$ is computed by modeling pairwise relationships between token i and any token j of the same PoC ($z_i = z_j$), while other tokens are excluded from consideration, as illustrated in Eqs. (5-6). The model encourages tokens of the same PoC to be assigned similar representations.

$$\alpha^h_{i,j} = \frac{\exp(\mathbf{q}_i^\top \mathbf{k}_j / \sqrt{d_k} + \mathcal{M}^h_{i,j})}{\sum_{j'=1}^{|\mathcal{S}|} \exp(\mathbf{q}_i^\top \mathbf{k}_{j'} / \sqrt{d_k} + \mathcal{M}^h_{i,j'})} \tag{5}$$

$$\mathcal{M}^h_{i,j} = \begin{cases} 0 & \text{if } i, j \leq |\mathbf{x}| \;\&\; z_i = z_j \\ -\infty & \text{otherwise} \end{cases} \tag{6}$$

where $\mathcal{M}^h_{i,j} = 0$ allows token i to attend to j. It happens when i and j are both source tokens and they pertain to mentions of the same PoC; otherwise token i cannot attend to j under the governance of attention head h and $\mathcal{M}^h_{i,j} = -\infty$. As an example, "*Allan Donald*" and "*The 48-year-old former Test paceman*" are co-referring mentions. TRANS-SHAREREPR allows these tokens to only attend to each other when learning representations using attention head h. These tokens are likely to yield similar representations. TRANS-SHAREREPR thus accomplishes a similar goal as TRANS-LINKING to allow *tokens* of the same PoC to be treated equivalently during summary generation; we explore the selection of attention heads in §5. Comparing the two models, we are particularly interested in understanding the necessity of introducing special markup tokens and how that may facilitate the shift of attention. Both models allow for multiple PoC per fusion instance. They vary in terms of how mentions of the same PoC are processed by the Transformer architecture.



Sentence 1	Ousted Yemen President Abdu Rabu Mansour Hadi thanked the Saudi - led coalition.
Sentence 2	Hadi claims he ' s Yemen ' s legitimate leader and is working with the Saudis and other allies to return to his country.
Summary	Ousted leader Abdu Rabu Mansour Hadi promises to return.

○ Repetition ○ Pronoun ○ Name ○ Non-Name ○ Event

Figure 3: An illustration of annotation interface for points of correspondence (POC) between sentences. A human annotator is asked to highlight text spans referring to the same entity, then choose one from the five pre-defined PoC types.

We assume points of correspondence (PoC) between a pair of sentences are available. The Transformer model learns to fuse two sentences into a single summary sentence by retaining the essential content and considering cohesive ties between sentences. In the following we introduce a new sentence fusion dataset with PoC annotations, which extends previous work to a larger scale and serves as a basis for future research to measure the success of sentence fusion systems.

4 Data and Annotation

We propose guidelines for annotating *points of correspondence* between sentences based on Halliday and Hasan's theory of cohesion. We consider points of correspondence as cohesive phrases that tie sentences together into a coherent text. PoC are categorized into five types, including pronominal coreference ("*John*" and "*he*"), nominal coreference ("*Johnny Kemp*" and "*The Bahamian R&B singer*"), common-noun coreference ("*five women*" and "*the five patients*"), repetition, and event trigger words that are related in meaning ("*died*" and "*drowned*"). Our categorization emphasizes the lexical linking that holds a text together and gives it meaning.

Pairs of input sentences and fusions are obtained from the CNN/DailyMail news corpus [See *et al.*, 2017]. We take a human summary sentence as an anchor point (i.e., a fusion sentence) to find two document sentences that are most similar to it, using a heuristic based on ROUGE similarity [Lebanoff *et al.*, 2019]. It becomes an instance containing a pair of input sentences and their fusion. This method allows us to identify a large quantity of candidate fusion instances.

A human annotator is instructed to identify a text span from each of the input sentences and fusion sentence, thus establishing a point of correspondence between input sentences, and between input and fusion sentences. If multiple PoC co-exist

Coref Resolver	P(%)	R(%)	F(%)
SpaCy	**59.2**	20.1	30.0
AllenNLP	49.0	24.5	32.7
Stanford	54.2	**26.2**	**35.3**

Table 2: Results of coreference resolvers on our sentence fusion dataset with points of correspondence (PoC)

in an example, an annotator is expected to label them all. An illustration of the interface is provided in Figure 3. We are particularly interested in annotating inter-sentence PoC. If entity mentions ("*John*" and "*he*") are found in the same sentence, we do not label them but assume intra-sentence coreference links can be captured by an existing resolver.

Annotation proceeds in two stages. Stage one removes all spurious pairs that are generated by the heuristic, i.e. a fusion sentence that is not a valid fusion of the corresponding two input sentences. Human annotators are given a pair of sentences and a fusion sentence and are asked whether it represents a valid fusion. The pairs identified as valid fusions by a majority of annotators move on to stage two. Stage two identifies the corresponding regions in the sentences. Annotators are again given an input pair and fusion and are tasked with highlighting the corresponding regions between each sentence. They must also choose one of the five PoC types (pronominal, nominal, etc.) for the set of corresponding regions.

We use Amazon Mechanical Turk, allowing only workers with 95% approval rate and at least 5,000 accepted tasks. To ensure high quality annotations, we first run a qualification round of 10 tasks. Workers performing sufficiently on the task may continue with the full set of tasks. In total, 2,221 fusion instances were evaluated in stage one and 727 of them were filtered out. Finally, we obtain points of correspondence from **1,494 instances, taken from 1,174 documents** in the test and valid splits of CNN/DM. We calculate Fleiss' Kappa judged on each token (highlighted or not), yielding substantial inter-annotator agreement (0.58).

Importantly, coreference resolution is very similar to the task of identifying points of correspondence. It is thus a natural step to analyze how well coreference resolvers can be used for PoC identification. If coreference resolvers can perform reasonably well, then these resolvers can be used to automatically extract PoC to enhance fusion models. We compare three coreference resolvers on this task: Stanford CoreNLP, SpaCy, and AllenNLP.[1] The results are presented in Table 2. The three resolvers have similar F-scores, but they all perform poorly at identifying points of correspondence. The SpaCy resolver has the highest precision (59.2%), but significantly lower recall. Our results indicate that coreference resolution models generally struggle to use the high-level reasoning that humans use to determine what connects two sentences together.

5 Experiments

Models We proceed by investigating the effectiveness of various sentence fusion models, including (a) **Pointer-Generator** [See *et al.*, 2017] that employs an encoder-decoder architecture to condense input sentences to a vector representation, then decode it into a fusion sentence. (b) **Transformer**, our baseline Transformer architecture w/o PoC information. It is a strong baseline that resembles the UniLM model described in [Dong *et al.*, 2019]. The following models differ from the baseline only on the incorporation of PoC. (c) **Trans-Linking** uses special tokens to mark the boundaries of PoC mentions so that they can be used interchangeably during generation (§3.1). (d) **Trans-ShareRepr** allows tokens of the same PoC to share representations by dedicating an attention head of the Transformer to this purpose (§3.2). All models are trained (or fine-tuned) on the same training set containing 107k fusion instances from the training split of CNN/DM; PoC are identified by the spaCy resolver. We evaluate fusion models on two test sets, including a "heuristic set" containing testing instances and automatically identified PoC via spaCy, and a final test set containing 1,494 instances with human-labelled PoC.[2]

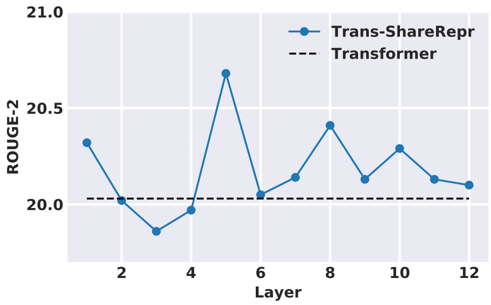


Figure 4: An attention head from the l-th layer is dedicated to coreference. The head encourages tokens of the same PoC to share similar representations. Our results suggest that the attention head of the 5-th layer achieves competitive performance, while most heads perform better than the baseline. The findings are congruent with [Clark *et al.*, 2019] that provides a detailed analysis of BERT's attention.

Results We experiment with a number of automatic evaluation metrics including ROUGE [Lin, 2004], BLEU [Papineni *et al.*, 2002] and BERTScore [Zhang *et al.*, 2019] that compares system output and reference based on BERT similarity. Results are presented in Table 3. We observe that all Transformer models outperform PG, suggesting that these models can benefit substantially from unsupervised pretraining on a large corpus of text. On the heuristic test set where training and testing conditions match (they both use automatically identified PoC), **Trans-Linking** performs better than **Trans-ShareRepr**, and vice versa on the final test set. We conjecture that this is because the linking model has a stronger requirement on PoC boundaries and the training/testing conditions must match for it to be effective. In contrast, **Trans-ShareRepr** is more lenient with mismatched conditions. We include a **Concat-Baseline** that creates a fusion by simply concatenating two input sentences. Its output contains 52 tokens on average, while other model outputs contain 15 tokens. This is a 70% compression rate, which adds to the challenge of content selection [Daumé III

[1] https://stanfordnlp.github.io/CoreNLP/, spacy.io, allennlp.org

[2] All Transformer models are fine-tuned based on BERT-BASE for 11 epochs, with a batch size of 32. The source and fusion sentences are tokenized with byte-pair encoding, and the combined input/output sequence is truncated to 128 tokens. We use the Adam optimizer with a learning rate of 2e-5 with warm-up. For PG, we use the default settings and truncate the output sequences to 60 tokens.

	Heuristic Set				Sentence Fusion w/ PoC Test Set						
System	R-1	R-2	R-L	BLEU	R-1	R-2	R-L	BLEU	B-Score	#Tkns	%Fuse
Concat-Baseline	37.2	20.0	28.7	25.0	36.1	18.6	27.8	24.6	60.4	52.0	99.7
Pointer-Generator	35.8	18.2	31.8	41.9	33.7	16.3	29.3	40.3	57.3	14.3	38.7
Transformer	39.6	20.9	35.3	47.2	38.8	20.0	33.8	45.8	61.3	15.1	50.7
Trans-Linking	**39.8**	**21.1**	**35.3**	**47.3**	38.8	20.1	33.9	45.5	61.1	15.1	**55.8**
Trans-ShareRepr	39.4	20.9	35.2	46.9	**39.0**	**20.2**	**33.9**	**45.8**	61.2	**15.2**	46.5

Table 3: Results of various sentence fusion systems evaluated by ROUGE, BLEU and BERTScore. Furthermore, we report the percentage of output sentences that are generated by fusion (%Fuse) and the average number of tokens per output sentence (#Tkns).

		Extractiveness		
System	Truthful.	1-gram	2-gram	3-gram
Pointer-Generator	63.6	97.5	83.1	72.8
Transformer	71.7	91.9	68.6	54.2
Trans-ShareRepr	70.9	92.0	70.1	56.4
Reference	67.2	72.0	34.9	20.9

Table 4: Fusion sentences are evaluated by their level of truthfulness and extractivenss. System-generated sentences have a high level of extractiveness. Our Trans-ShareRepr is most similar to Reference.

and Marcu, 2004]. Despite that all models are trained to fuse sentences, their outputs are not guaranteed to be fusions and shortening of single sentences is possible. We observe that **Trans-Linking** has the highest rate of producing fusions (56%).

Attention head Our **Trans-ShareRepr** model contains a total of 12 layers and there are 12 heads per layer. While dedicating a head to attend to PoC mentions has demonstrated promising performance, as shown in Table 3 (containing head-averaged results), it remains to be seen whether there is any principled way of choosing the head. In Figure 4 we examine the effect of different design choices, where the first attention head is selected from the l-th layer and dedicated to PoC. Our results suggest that the head from the 5-th layer achieves competitive performance, while most heads perform better than the baseline. Our findings are congruent with [Clark *et al.*, 2019] that provides a detailed analysis of BERT's attention.[3]

Human evaluation We finally gauge the quality of fusions with human evaluation. The models we use for comparison include (a) **Pointer-Generator**, (b) **Transformer**, (c) **Trans-ShareRepr** and (d) reference fusion sentences (§4). Experiments are performed on the final test where we sample 200 instances and take an extra step to make sure all model outputs for selected instances contain fusion sentences, as opposed to shortening of single sentences. A human evaluator is presented with two input sentences and a fusion sentence, and asked to evaluate if the fusion sentence has successfully retained the original meaning and not introduced any new meaning not present in the source. Each instance is judged by five human evaluators and results are presented in Table 4. In addition to *truthfulness* of fusions, we measure their *extractiveness* by reporting on the percentage of n-grams (n=1/2/3) that appear in the source. Interestingly, we observe that humans do not always rate reference fusions as truthful. This is in part because reference fusions exhibit a high level of abstraction and they occasionally contain content that is non-existent in the source (Table 5). If fusion sentences are less extractive, humans sometimes perceive that as less truthful, especially when compared to fusions that reuse the source text. Our findings suggest that there is an increasing need for better evaluation metrics for system-generated text, including question-answering-based [Narayan *et al.*, 2018] or providing more context to evaluators during evaluation [Daumé III and Marcu, 2004].

Source: Later that month, the ICC opened a preliminary examination into the situation in Palestinian territories, paving the way for possible war crimes investigations against Israelis.

Israel and the United States, neither of which is an ICC member, opposed the Palestinians' efforts to join the body.

Pointer-Generator: *ICC opened a preliminary examination into the situation in Palestinian territories .*

Transformer: *Israel, U.S. and the United States are investigating possible war crimes, paving way for war crimes.*

Transformer-ShareRepr: *Israel and U.S. opposed the ICC's investigation into the situation in Palestinian territories.*

Reference: *Israel and the United States opposed the move, which could open the door to war crimes investigations against Israelis.*

Table 5: Example output of sentence fusion systems. **PG** only performs sentence shortening rather than fusion. **Transformer** fails to retain the original meaning and **Transformer-ShareRepr** performs best. **Reference** demonstrates a high level of abstraction. Sentences are manually de-tokenized for readability.

6 Conclusion

In this paper, we address the challenge of information fusion in the context of neural abstractive summarization by making crucial use of points of correspondence between sentences. We propose to enrich Transformers with PoC information and investigate the effect of various design choices on model performance. More importantly, we present a benchmark dataset comprised of pairs of sentences and their fusions. It extends previous work to a larger scale and serves as a basis for future research to measure the success of sentence fusion systems. Our findings shed light on the importance of modeling points of correspondence for effective sentence fusion.

References

[Barzilay and McKeown, 2005] Regina Barzilay and Kathleen R. McKeown. Sentence fusion for multidocument news summarization. *Comp. Ling.*, 31(3), 2005.

[Barzilay *et al.*, 1999] Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. Information fusion in the context of multi-document summarization. In *ACL*, 1999.

[3] We choose to not experiment with dedicating multiple heads to PoC at the same time due to the limitations with computing resources.

[Celikyilmaz *et al.*, 2018] Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. Deep communicating agents for abstractive summarization. In *NAACL*, 2018.

[Chen and Bansal, 2018] Yen-Chun Chen and Mohit Bansal. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proc. of ACL*, 2018.

[Clark *et al.*, 2019] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proc. of the 2019 ACL Workshop BlackboxNLP*, 2019.

[Daumé III and Marcu, 2004] Hal Daumé III and Daniel Marcu. Generic sentence fusion is an ill-defined summarization task. In *Wksp on Text Summ. Branches Out*, 2004.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*, 2018.

[Dong *et al.*, 2019] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *https://arxiv.org/abs/1905.03197*, 2019.

[Elsner and Santhanam, 2011] Micha Elsner and Deepak Santhanam. Learning to fuse disparate sentences. In *Proc. of ACL Wksp on Monolingual Text-To-Text Gen.*, 2011.

[Falke *et al.*, 2019] Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proc. of ACL*, 2019.

[Fan *et al.*, 2018] Angela Fan, David Grangier, and Michael Auli. Controllable abstractive summarization. In *Proc. of the 2nd Wksp on NMT and Generation*, 2018.

[Filippova and Strube, 2008] Katja Filippova and Michael Strube. Sentence fusion via dependency graph compression. In *Proc. of EMNLP*, 2008.

[Gehrmann *et al.*, 2018] Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. Bottom-up abstractive summarization. In *Proc. of EMNLP*, 2018.

[Geva *et al.*, 2019] Mor Geva, Eric Malmi, Idan Szpektor, and Jonathan Berant. DISCOFUSE: A large-scale dataset for discourse-based sentence fusion. In *NAACL*, 2019.

[Grosz *et al.*, 1995] Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. Centering: A framework for modeling the local coherence of discourse. *Comp. Ling.*, 1995.

[Halliday and Hasan, 1976] Michael A. K. Halliday and Ruqaiya Hasan. *Cohesion in English*. English Language Series. Longman Group Ltd., 1976.

[Khandelwal *et al.*, 2019] Urvashi Khandelwal, Kevin Clark, Dan Jurafsky, and Lukasz Kaiser. Sample efficient text summarization using a single pre-trained transformer. *arXiv:1905.08836*, 2019.

[Kryściński *et al.*, 2019] Wojciech Kryściński, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. Neural text summarization: A critical evaluation. In *Proc. of EMNLP*, 2019.

[Lebanoff *et al.*, 2019] Logan Lebanoff, John Muchovej, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. Analyzing sentence fusion in abstractive summarization. In *Proc. of the 2nd Workshop on New Frontiers in Summarization*, 2019.

[Lewis *et al.*, 2019] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *arXiv:1910.13461*, 2019.

[Lin, 2004] Chin-Yew Lin. ROUGE: a package for automatic evaluation of summaries. In *Proc. of ACL Workshop on Text Summarization Branches Out*, 2004.

[Liu and Lapata, 2019] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In *EMNLP*, 2019.

[Marsi and Krahmer, 2005] Erwin Marsi and Emiel Krahmer. Explorations in sentence fusion. In *ACL Wksp on Comp. Approaches to Semitic Languages*, 2005.

[Mehdad *et al.*, 2013] Yashar Mehdad, Giuseppe Carenini, Frank W. Tompa, and Raymond T. NG. Abstractive meeting summarization with entailment and fusion. In *Proc. of the 14th European Wksp on Nat. Lang. Gen.*, 2013.

[Narayan *et al.*, 2018] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *Proc. of EMNLP*, 2018.

[Paulus *et al.*, 2017] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. In *Proc. of EMNLP*, 2017.

[Raffel *et al.*, 2019] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv:1910.10683*, 2019.

[See *et al.*, 2017] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proc. of ACL*, 2017.

[Sharma *et al.*, 2019] Eva Sharma, Luyang Huang, Zhe Hu, and Lu Wang. An entity-driven framework for abstractive summarization. In *Proc. of EMNLP*, 2019.

[Strubell *et al.*, 2018] Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. Linguistically-informed self-attention for semantic role labeling. In *Proc. of EMNLP*, 2018.

[Thadani and McKeown, 2013] Kapil Thadani and Kathleen McKeown. Supervised sentence fusion with single-stage inference. In *Proc. of IJCNLP*, 2013.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. of NIPS*, 2017.