Understanding Points of Correspondence between Sentences for Abstractive Summarization

Logan Lebanoff[†] **John Muchovej**[†] **Franck Dernoncourt**[§]
Doo Soon Kim[§] **Lidan Wang**[§] **Walter Chang**[§] **Fei Liu**[†]

[†]University of Central Florida [§]Adobe Research
{loganlebanoff,john.muchovej}@knights.ucf.edu feiliu@cs.ucf.edu
{dernonco,dkim,lidwang,wachang}@adobe.com

Abstract

Fusing sentences containing disparate content is a remarkable human ability that helps create informative and succinct summaries. Such a simple task for humans has remained challenging for modern abstractive summarizers, substantially restricting their applicability in real-world scenarios. In this paper, we present an investigation into fusing sentences drawn from a document by introducing the notion of points of correspondence, which are cohesive devices that tie any two sentences together into a coherent text. The types of points of correspondence are delineated by text cohesion theory, covering pronominal and nominal referencing, repetition and beyond. We create a dataset containing the documents, source and fusion sentences, and human annotations of points of correspondence between sentences. Our dataset bridges the gap between coreference resolution and summarization. It is publicly shared to serve as a basis for future work to measure the success of sentence fusion systems.[1]

1 Introduction

Stitching portions of text together into a sentence is a crucial first step in abstractive summarization. It involves choosing which sentences to fuse, what content from each of them to retain and how best to present that information (Elsner and Santhanam, 2011). A major challenge in fusing sentences is to establish correspondence between sentences. If there exists no correspondence, it would be difficult, if not impossible, to fuse sentences. In Table 1, we present example source and fusion sentences, where the summarizer attempts to merge two sentences into a summary sentence with improper use of *points of correspondence*. In this paper, we seek to uncover hidden correspondences between sentences, which has a great potential for improving content selection and deep sentence fusion.

[Source Sentences] Robert Downey Jr. is making headlines for walking out of an interview with a British journalist who dared to veer away from the superhero movie Downey was there to promote. The journalist instead started asking personal questions about the actor's political beliefs and "dark periods" of addiction and jail time.
[Summary] Robert Downey Jr started asking personal questions about the actor's political beliefs.
[Source Sentences] "Real Housewives of Beverly Hills" star and former child actress Kim Richards is accused of kicking a police officer after being arrested Thursday morning. A police representative said Richards was asked to leave but refused and then entered a restroom and wouldn't come out.
[Summary] Kim Richards is accused of kicking a police officer who refused to leave.
[Source Sentences] 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.

Table 1: Unfaithful summary sentences generated by neural abstractive summarizers, in-house and PG (See et al., 2017). They attempt to merge two sentences into one sentence with improper use of *points of correspondence* between sentences, yielding nonsensical output. Summaries are manually re-cased for readability.

Sentence fusion (or multi-sentence compression) plays a prominent role in automated summarization and its importance has long been recognized (Barzilay et al., 1999). Early attempts to fuse sentences build a dependency graph from sentences, then decode a tree from the graph using integer linear programming, finally linearize the tree to generate a summary sentence (Barzilay and McKeown, 2005; Filippova and Strube, 2008; Thadani and McKeown, 2013a). Despite valuable insights gained from

[1] https://github.com/ucfnlp/points-of-correspondence

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 191–198
July 5 - July 10, 2020.

PoC Type	Source Sentences	Summary Sentence
Pronominal Referencing	**[S1]** The bodies showed signs of torture. **[S2]** They were left on the side of a highway in Chilpancingo, about an hour north of the tourist resort of Acapulco in the state of Guerrero.	• The bodies of the men, which showed signs of torture, were left on the side of a highway in Chilpancingo.
Nominal Referencing	**[S1]** Bahamian R&B singer Johnny Kemp, best known for the 1988 party anthem "Just Got Paid," died this week in Jamaica. **[S2]** The singer is believed to have drowned at a beach in Montego Bay on Thursday, the Jamaica Constabulatory Force said in a press release.	• Johnny Kemp is "believed to have drowned at a beach in Montego Bay," police say.
Common-Noun Referencing	**[S1]** A nurse confessed to killing five women and one man at hospital. **[S2]** A former nurse in the Czech Republic murdered six of her elderly patients with massive doses of potassium in order to ease her workload.	• The nurse, who has been dubbed "nurse death" locally, has admitted killing the victims with massive doses of potassium.
Repetition	**[S1]** Stewart said that she and her husband, Joseph Naaman, booked Felix on their flight from the United Arab Emirates to New York on April 1. **[S2]** The couple said they spent $1,200 to ship Felix on the 14-hour flight.	• Couple spends $1,200 to ship their cat, Felix, on a flight from the United Arab Emirates.
Event Triggers	**[S1]** Four employees of the store have been arrested, but its manager was still at large, said Goa police superintendent Kartik Kashyap. **[S2]** If convicted, they could spend up to three years in jail, Kashyap said.	• The four store workers arrested could spend 3 years each in prison if convicted.

Table 2: Types of sentence correspondences. Text cohesion can manifest itself in different forms.

these attempts, experiments are often performed on small datasets and systems are designed to merge sentences conveying *similar* information. Nonetheless, humans do not restrict themselves to combine similar sentences, but also *disparate* sentences containing fundamentally different content but remain related to make fusion sensible (Elsner and Santhanam, 2011). We focus specifically on analyzing fusion of *disparate* sentences, which is a distinct problem from fusing a set of *similar* sentences.

While fusing disparate sentences is a seemingly simple task for humans to do, it has remained challenging for modern abstractive summarizers (See et al., 2017; Celikyilmaz et al., 2018; Chen and Bansal, 2018; Liu and Lapata, 2019). These systems learn to perform content selection and generation through end-to-end learning. However, such a strategy is not consistently effective and they struggle to reliably perform sentence fusion (Falke et al., 2019; Kryściński et al., 2019). E.g., only 6% of summary sentences generated by pointer-generator networks (See et al., 2017) are fusion sentences; the ratio for human abstracts is much higher (32%). Further, Lebanoff et al. (2019a) report that 38% of fusion sentences contain incorrect facts. There exists a pressing need for—and this paper contributes to–broadening the understanding of points of correspondence used for sentence fusion.

We present the first attempt to construct a sizeable sentence fusion dataset, where an instance in the dataset consists of a pair of input sentences, a fusion sentence, and human-annotated *points of correspondence* between sentences. Distinguishing our work from previous efforts (Geva et al., 2019), our input contains *disparate* sentences and output is a fusion sentence containing important, though not equivalent information of the input sentences. Our investigation is inspired by Halliday and Hasan's theory of *text cohesion* (1976) that covers a broad range of points of correspondence, including entity and event coreference (Ng, 2017; Lu and Ng, 2018), shared words/concepts between sentences and more. Our contributions are as follows.

- We describe the first effort at establishing points of correspondence between disparate sentences. Without a clear understanding of points of correspondence, sentence fusion remains a daunting challenge that is only sparsely and sometimes incorrectly performed by abstractive summarizers.
- We present a sizable dataset for sentence fusion containing human-annotated corresponding regions between pairs of sentences. It can be used as a testbed for evaluating the ability of summarization models to perform sentence fusion. We report on the insights gained from annotations to suggest important future directions for sentence fusion. Our dataset is released publicly.

2 Annotating Points of Correspondence

We cast sentence fusion as a constrained summarization task where portions of text are selected from each source sentence and stitched together to form a fusion sentence; rephrasing and reordering are allowed in this process. We propose guidelines for annotating *points of correspondence* (PoC) between sentences based on Halliday and Hasan's theory of cohesion (1976).

We consider points of correspondence as cohesive phrases that tie sentences together into a coherent text. Guided by text cohesion theory, we categorize PoC into five types, including pronominal referencing ("*they*"), nominal referencing ("*Johnny Kemp*"), common-noun referencing ("*five women*"),

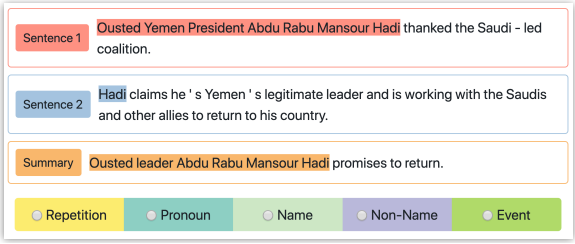

Figure 1: An illustration of the annotation interface. A human annotator is asked to highlight text spans referring to the same entity, then choose one from the five pre-defined PoC types.

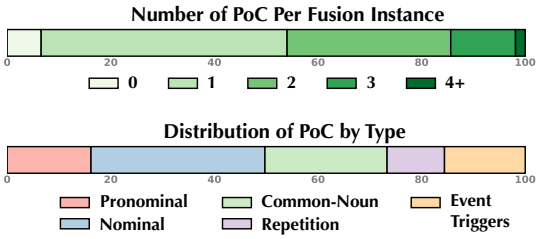

Figure 2: Statistics of PoC occurrences and types.

repetition, and event trigger words that are related in meaning ("*died*" and "*drowned*"). An illustration of PoC types is provided in Table 2. Our categorization emphasizes the lexical linking that holds a text together and gives it meaning.

A human annotator is instructed to identify a text span from each of the source sentences and summary sentence, thus establishing a point of correspondence between source sentences, and between source and summary sentences. As our goal is to understand the role of PoC in sentence fusion, we do not consider the case if PoC is only found in source sentences but not summary sentence, e.g., "*Kashyap said*" and "*said Goa police superintendent Kartik Kashyap*" in Table 2. If multiple PoC co-exist in an example, an annotator is expected to label them all; a separate PoC type will be assigned to each PoC occurrence. We are particularly interested in annotating inter-sentence PoC. If entity mentions ("*John*" and "*he*") are found in the same sentence, we do not explicitly label them but assume such intra-sentence referencing can be captured by an existing coreference resolver. Instances of source sentences and summary sentences are obtained from the test and validation splits of the CNN/DailyMail corpus (See et al., 2017) following the procedure described by Lebanoff et al. (2019a). We take a human summary sentence as an anchor point to find two document sentences that are most similar to it based on ROUGE. It becomes an instance containing a pair of source sentences and their summary. The method allows us to identify a large quantity of candidate fusion instances.

Annotations are performed in two stages. Stage one removes all spurious pairs that are generated by the heuristic, i.e. a summary sentence that is not a valid fusion of the corresponding two source sentences. Human annotators are given a pair of sentences and a summary 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. As shown in Figure 1, annotators are given a pair of sentences and their summary and are tasked with highlighting the corresponding regions between each sentence. They must also choose one of the five PoC types (repetition, pronominal, nominal, common-noun referencing, and event triggers) 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 these tasks were allowed to annotate the whole dataset. For task one, 2,200 instances were evaluated and 621 of them were filtered out. In total, we annotate points of correspondence for **1,599 instances, taken from 1,174 documents**. Similar to (Hardy et al., 2019), we report Fleiss' Kappa judged on each word (highlighted or not), yielding substantial inter-annotator agreement (κ=0.58) for annotating points of correspondence. We include a reference to the original article that each instance was taken from, thus providing context for each instance.

Figure 2 shows statistics of PoC occurrence frequencies and the distribution of different PoC types. A majority of sentence pairs have one or two points of correspondence. Only a small percentage (6.5%) do not share a PoC. A qualitatively analysis shows that these sentences often have an *implicit* discourse relationship, e.g., "*The two men speak. Scott then gets out of the car, again, and runs away.*" In this example, there is no clear portion of text that is shared between the sentences; rather, the connection lies in the fact that one event happens after the other. Most of the PoC are a flavor of coreference (pronominal, nominal, or common-noun). Few are exact repetition. Further, we find that only 38% of points of correspondence in the sentence pair share

Coref Resolver	P(%)	R(%)	F(%)	Pronominal	Nominal	Comm.-Noun	Repetition	Event Trig.
SpaCy	**59.2**	20.1	30.0	30.8	23.3	10.4	39.9	2.6
AllenNLP	49.0	24.5	32.7	36.5	28.1	14.7	47.1	3.1
Stanford CoreNLP	54.2	**26.2**	**35.3**	40.0	27.3	17.4	55.1	2.3

Table 3: Results of various coreference resolvers on successfully identifying inter-sentence points of correspondence (PoC) and recall scores of these resolvers split by PoC correspondence type.

any words (lemmatized). This makes identifying them automatically challenging, requiring a deeper understanding of what connects the two sentences.

3 Resolving Coreference

Coreference resolution (Ng, 2017) is similar to the task of identifying points of correspondence. Thus, a natural question we ask is how well state-of-the-art coreference resolvers can be adapted to this task. If coreference resolvers can perform reasonably well on PoC identification, then these resolvers can be used to extract PoC annotations to potentially enhance sentence fusion. If they perform poorly, coreference performance results can indicate areas of improvement for future work on detecting points of correspondence. In this paper, we compare three coreference resolvers on our dataset, provided by open-source libraries: Stanford CoreNLP (Manning et al., 2014), SpaCy (Honnibal and Montani, 2017), and AllenNLP (Gardner et al., 2017).

We base our evaluation on the standard metric used for coreference resolution, B-CUBED algorithm (Bagga and Baldwin, 1998), with some modifications. Each resolver is run on an input pair of sentences to obtain multiple clusters, each representing an entity (e.g., *Johnny Kemp*) containing multiple mentions (e.g., *Johnny Kemp*; *he*; *the singer*) of that entity. More than one cluster can be detected by the coreference resolver, as additional entities may exist in the given sentence pair (e.g., *Johnny Kemp* and *the police*). Similarly, in Section §2, human annotators identified multiple PoC clusters, each representing a point of correspondence containing one mention from each sentence. We evaluate how well the resolver-detected clusters compare to the human-detected clusters (i.e., PoCs). If a resolver cluster overlaps both mentions for the gold-standard PoC, then this resolver cluster is classified as a hit. Any resolver cluster that does not overlap both PoC mentions is a miss. Using this metric, we can calculate precision, recall, and F1 scores based on correctly/incorrectly identified tokens from the outputs of each resolver.

The results are presented in Table 3. The three resolvers exhibit similar performance, but the scores on identifying points of correspondence are less than satisfying. The SpaCy resolver has the highest precision (59.2%) and Stanford CoreNLP achieves the highest F1-score (35.3%). We observe that existing coreference resolvers can sometimes struggle to use the high-level reasoning that humans use to determine what connects two sentences together. Next, we go deeper into understanding what PoC types these resolvers struggle with. We present the recall scores of these resolvers split by PoC correspondence type. Event coreference poses the most difficulty by far, which is understandable as coreference resolution only focuses on entities rather than events. More work into detecting event coreference can bring significant improvements in PoC identification. Common-noun coreference also poses a challenge, in part because names and pronouns give strong clues as to the relationships between mentions, while common-noun relationships are more difficult to identify since they lack these clues.

4 Sentence Fusion

Truly effective summarization will only be achievable when systems have the ability to fully recognize points of correspondence between sentences. It remains to be seen whether such knowledge can be acquired implicitly by neural abstractive systems through joint content selection and generation. We next conduct an initial study to assess neural abstractive summarizers on their ability to perform sentence fusion to merge two sentences into a summary sentence. The task represents an important, atomic unit of abstractive summarization, because a long summary is still generated one sentence at a time (Lebanoff et al., 2019b).

We compare two best-performing abstractive summarizers: *Pointer-Generator* uses an encoder-decoder architecture with attention and copy mechanism (See et al., 2017); *Transformer* adopts a decoder-only Transformer architecture similar to that of (Radford et al., 2019), where a summary is

System	R-1	R-2	R-L	%Fuse
Concat-Baseline	36.13	18.64	27.79	99.7
Pointer-Generator	33.74	16.32	29.27	38.7
Transformer	**38.81**	**20.03**	**33.79**	50.7

Table 4: ROUGE scores of neural abstractive summarizers on the sentence fusion dataset. We also report the percentage of output sentences that are indeed fusion sentences (%Fuse)

decoded one word at a time conditioned on source sentences and the previously-generated summary words. We use the same number of heads, layers, and units per layer as BERT-base (Devlin et al., 2018). In both cases, the summarizer was trained on about 100k instances derived from the train split of CNN/DailyMail, using the same heuristic as described in (§2) without PoC annotations. The summarizer is then tested on our dataset of 1,599 fusion instances and evaluated using standard metrics (Lin, 2004). We also report how often each summarizer actually draws content from both sentences (*%Fuse*), rather than taking content from only one sentence. A generated sentence counts as a fusion if it contains at least two non-stopword tokens from each sentence not already present in the other sentence. Additionally, we include a *Concat-Baseline* creating a fusion sentence by simply concatenating the two source sentences.

The results according to the ROUGE evaluation (Lin, 2004) are presented in Table 4. Sentence fusion appears to be a challenging task even for modern abstractive summarizers. Pointer-Generator has been shown to perform strongly on abstractive summarization, but it is less so on sentence fusion and in other highly abstractive settings (Narayan et al., 2018). Transformer significantly outperforms other methods, in line with previous findings (Liu et al., 2018). We qualitatively examine system outputs. Table 1 presents fusions generated by these models and exemplifies the need for infusing models with knowledge of points of correspondence. In the first example, Pointer-Generator incorrectly conflates *Robert Downey Jr.* with the *journalist* asking questions. Similarly, in the second example, Transformer states the *police officer* refused to leave when it was actually *Richards*. Had the models explicitly recognized the points of correspondence in the sentences—that *the journalist* is a separate entity from *Robert Downey Jr.* and that *Richards* is separate from *police officer*—then a more accurate summary could have been generated.

5 Related Work

Uncovering hidden correspondences between sentences is essential for producing proper summary sentences. A number of recent efforts select important words and sentences from a given document, then let the summarizer attend to selected content to generate a summary (Gehrmann et al., 2018; Hsu et al., 2018; Chen and Bansal, 2018; Putra et al., 2018; Lebanoff et al., 2018; Liu and Lapata, 2019). These systems are largely agnostic to sentence correspondences, which can have two undesirable consequences. If only a single sentence is selected, it can be impossible for the summarizer to produce a fusion sentence from it. Moreover, if *non-fusible* textual units are selected, the summarizer is forced to fuse them into a summary sentence, yielding output summaries that often fail to keep the original meaning intact. Therefore, in this paper we had investigated the correspondences between sentences to gain an understanding of sentence fusion.

Establishing correspondence between sentences goes beyond finding common words. Humans can fuse sentences sharing *few or no* common words if they can find other types of correspondence. Fusing such disparate sentences poses a serious challenge for automated fusion systems (Marsi and Krahmer, 2005; Filippova and Strube, 2008; McKeown et al., 2010; Elsner and Santhanam, 2011; Thadani and McKeown, 2013b; Mehdad et al., 2013; Nayeem et al., 2018). These systems rely on common words to derive a connected graph from input sentences or subject-verb-object triples (Moryossef et al., 2019). When there are no common words in sentences, systems tend to break apart.

There has been a lack of annotated datasets and guidelines for sentence fusion. Few studies have investigated the types of correspondence between sentences such as entity and event coreference. Evaluating sentence fusion systems requires not only novel metrics (Zhao et al., 2019; Zhang et al., 2020; Durmus et al., 2020; Wang et al., 2020) but also high-quality ground-truth annotations. It is therefore necessary to conduct a first study to look into cues humans use to establish correspondence between disparate sentences.

We envision sentence correspondence to be related to text *cohesion* and *coherence*, which help establish correspondences between two pieces of text. Halliday and Hasan (1976) describe text **cohesion** as cohesive devices that tie two textual elements together. They identify five categories of cohesion:

(McKeown et al., 2010)
[S1] Palin actually turned against the bridge project only after it became a national symbol of wasteful spending. **[S2]** Ms. Palin supported the bridge project while running for governor, and abandoned it after it became a national scandal. **[Fusion]** Palin turned against the bridge project after it became a national scandal.
DiscoFuse (Geva et al., 2019)
[S1] Melvyn Douglas originally was signed to play Sam Bailey. **[S2]** The role ultimately went to Walter Pidgeon. **[Fusion]** Melvyn Douglas originally was signed to play Sam Bailey, but the role ultimately went to Walter Pidgeon.
Points of Correspondence Dataset (Our Work)
[S1] The bodies showed signs of torture. **[S2]** They were left on the side of a highway in Chilpancingo, about an hour north of the tourist resort of Acapulco in the state of Guerrero. **[Fusion]** The bodies of the men, which showed signs of torture, were left on the side of a highway in Chilpancingo.

Table 5: Comparison of sentence fusion datasets.

reference, *lexical cohesion*, *ellipsis*, *substitution* and *conjunction*. In contrast, **coherence** is defined in terms of discourse relations between textual elements, such as *elaboration*, *cause* or *explanation*. Previous work studied discourse relations (Geva et al., 2019), this paper instead focuses on *text cohesion*, which plays a crucial role in generating proper fusion sentences. Our dataset contains pairs of source and fusion sentences collected from news editors in a natural environment. The work is particularly meaningful to text-to-text and data-to-text generation (Gatt and Krahmer, 2018) that demand robust modules to merge disparate content.

We contrast our dataset with previous sentence fusion datasets. McKeown et al. (2010) compile a corpus of 300 sentence fusions as a first step toward a supervised fusion system. However, the input sentences have very similar meaning, though they often present lexical variations and different details. In contrast, our proposed dataset seeks to fuse significantly different meanings together into a single sentence. A large-scale dataset of sentence fusions has been recently collected (Geva et al., 2019), where each sentence has disparate content and are connected by various discourse connectives. This paper instead focuses on *text cohesion* and on fusing only the salient information, which are both vital for abstractive summarization. Examples are presented in Table 5.

6 Conclusion

In this paper, we describe a first effort at annotating points of correspondence between disparate sentences. We present a benchmark dataset comprised of the documents, source and fusion sentences, and human annotations of points of correspondence between sentences. The dataset fills a notable gap of coreference resolution and summarization research. Our findings shed light on the importance of modeling points of correspondence, suggesting important future directions for sentence fusion.

Acknowledgments

We are grateful to the anonymous reviewers for their helpful comments and suggestions. This research was supported in part by the National Science Foundation grant IIS-1909603.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Granada.

Regina Barzilay and Kathleen R. McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3).

Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*.

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)*.

Micha Elsner and Deepak Santhanam. 2011. Learning to fuse disparate sentences. In *Proceedings of ACL Workshop on Monolingual Text-To-Text Generation*.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019.

Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Katja Filippova and Michael Strube. 2008. Sentence fusion via dependency graph compression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

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)*.

Mor Geva, Eric Malmi, Idan Szpektor, and Jonathan Berant. 2019. DISCOFUSE: A large-scale dataset for discourse-based sentence fusion. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

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

Hardy Hardy, Shashi Narayan, and Andreas Vlachos. 2019. HighRES: Highlight-based reference-less evaluation of summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3381–3392, Florence, Italy. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. A unified model for extractive and abstractive summarization using inconsistency loss. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Wojciech Kryściński, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Logan Lebanoff, John Muchovej, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019a. Analyzing sentence fusion in abstractive summarization. In *Proceedings fo the EMNLP 2019 Workshop on New Frontiers in Summarization*.

Logan Lebanoff, Kaiqiang Song, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019b. Scoring sentence singletons and pairs for abstractive summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Logan Lebanoff, Kaiqiang Song, and Fei Liu. 2018. Adapting the neural encoder-decoder framework from single to multi-document summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Łukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. In *Sixth International Conference on Learning Representations (ICLR)*.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jing Lu and Vincent Ng. 2018. Event coreference resolution: A survey of two decades of research. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*.

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

Erwin Marsi and Emiel Krahmer. 2005. Explorations in sentence fusion. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*.

Kathleen McKeown, Sara Rosenthal, Kapil Thadani, and Coleman Moore. 2010. Time-efficient creation of an accurate sentence fusion corpus. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Yashar Mehdad, Giuseppe Carenini, Frank W. Tompa, and Raymond T. NG. 2013. Abstractive meeting summarization with entailment and fusion. In *Proceedings of the 14th European Workshop on Natural Language Generation*.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-Step: Separating planning from realization in neural data-to-text generation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

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 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Mir Tafseer Nayeem, Tanvir Ahmed Fuad, and Yllias Chali. 2018. Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Vincent Ng. 2017. Machine learning for entity coreference resolution: A retrospective look at two decades of research. In *Proceedngs of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*.

Jan Wira Gotama Putra, Hayato Kobayashi, and Nobuyuki Shimizu. 2018. Incorporating topic sentence on neural news headline generation.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Kapil Thadani and Kathleen McKeown. 2013a. Sentence compression with joint structural inference. In *Proceedings of CoNLL*.

Kapil Thadani and Kathleen McKeown. 2013b. Supervised sentence fusion with single-stage inference. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*.

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)*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.