Discourse-Based Sentence Splitting

Liam Cripwell, Joel Legrand, Claire Gardent

Discourse-Based Sentence Splitting

Liam Cripwell
Université de Lorraine
CNRS/LORIA
liam.cripwell@loria.fr

Joël Legrand
Université de Lorraine
Centrale Supélec
CNRS/LORIA
joel.legrand@inria.fr

Claire Gardent
CNRS/LORIA
Université de Lorraine
claire.gardent@loria.fr

Abstract

Sentence splitting involves the segmentation of a sentence into two or more shorter sentences. It is a key component of sentence simplification, has been shown to help human comprehension and is a useful preprocessing step for NLP tasks such as summarisation and relation extraction. While several methods and datasets have been proposed for developing sentence splitting models, little attention has been paid to how sentence splitting interacts with discourse structure. In this work, we focus on cases where the input text contains a discourse connective, which we refer to as discourse-based sentence splitting. We create synthetic and organic datasets for discourse-based splitting and explore different ways of combining these datasets using different model architectures. We show that pipeline models which use discourse structure to mediate sentence splitting outperform end-to-end models in learning the various ways of expressing a discourse relation but generate text that is less grammatical; that large scale synthetic data provides a better basis for learning than smaller scale organic data; and that training on discourse-focused, rather than on general sentence splitting data provides a better basis for discourse splitting.

1 Introduction

Sentence splitting segments a sentence into two or more shorter sentences. It is a key component of sentence simplification. It has also been shown to help human comprehension (Mason, 1978; Williams et al., 2003) and to be a useful preprocessing step for several NLP tasks, such as relation extraction (Niklaus et al., 2016) and machine translation (Chandrasekar et al., 1996; Mishra et al., 2014; Li and Nenkova, 2015; Mishra et al., 2014).

There is a large body of work on sentence splitting. It has been studied in the context of many text simplification systems (Siddharthan, 2006; Zhu et al., 2010; Woodsend and Lapata, 2011; Siddharthan and Mandya, 2014; Narayan et al., 2017; Narayan and Gardent, 2016, 2014) and is the focus of so-called, split-and-rephrase models (Narayan et al., 2017; Aharoni and Goldberg, 2018; Botha et al., 2018; Niklaus et al., 2019b,a,c).

So far however, little attention has been paid to how discourse splitting interacts with discourse structure. As illustrated in Table 1, two main types of splitting can be distinguished depending on whether the split is licensed by a syntactic construct or by a discourse connective. Whereas syntax-based splitting is licensed by syntactic constructs such as relative clauses, VP or sentence coordinations, gerund or appositive constructions, discourse-based splitting is licensed by the presence of a discourse relation between two discourse units.

Importantly, in the case of discourse-based splitting, the discourse relation which holds in the input must be preserved in the split output. This is illustrated in Table 1 where the temporal relation marked by ***and after this*** in the input (C1) is made explicit in the split output (S1) by the adverbial ***Afterwards***. In contrast, omitting this adverbial (S3) results in a semantic loss and makes the output more difficult to understand. As shown by the (S2) variant, a split can also use a discourse adverbial with an inverse meaning (***Before this***) which induces a corresponding inversion in the linear order of the text.

In this paper, we focus on discourse-based sentence splitting and make the following contributions:

1. We create synthetic and organic training data for discourse splitting and investigate various ways of leveraging this data for training discourse-based sentence splitting models.

2. We compare a discourse-agnostic, end-to-end approach with a pipeline model that uses dis-

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 261–273
November 7–11, 2021.

C1.	The Masovians were caught by surprise, since virtually without any defense the capital, Płock, fell **and after this** Mindaugas crossed the Vistula river and captured the fortress of Jazdów.
S1. ✓	The Masovians were caught by surprise, since virtually without any defense the capital, Płock, fell. **Afterwards**, Mindaugas crossed the Vistula river and captured the fortress of Jazdów.
S2. ✓	Mindaugas crossed the Vistula river and captured the fortress of Jazdów. **Before this**, the Masovians were caught by surprise, since virtually without any defense the capital, Płock, fell.
S3. ✗	Mindaugas crossed the Vistula river and captured the fortress of Jazdów. The Masovians were caught by surprise, since virtually without any defense the capital, Płock, fell.
T	<DR> TEMPORAL:ASYNCHRONOUS <ARG1> The Masovians were caught by surprise, since virtually without any defense the capital, Płock, fell <ARG2> Mindaugas crossed the Vistula river and captured the fortress of Jazdów <EOS>
C2.	He settled in London, devoting himself chiefly to practical teaching.
S4.	He settled in London. He devoted himself chiefly to practical teaching.
C3.	It was a time to go back to nature, and the plastic flamingo quickly became the prototype of bad taste and anti-nature.
S5.	It was a time to go back to nature. The plastic flamingo quickly became the prototype of bad taste and anti-nature.

Table 1: Discourse- (1) vs. Syntax-Based (2) Sentence Splitting

course structure to mediate the split.

3. We show that training on discourse-focused rather than general sentence splitting data helps to improve performance.

4. To help spur research on discourse-based sentence splitting, we make our dataset and code publicly available. [1]

2 Related Work

Together with deletion, reordering and substitution, sentence splitting is one of the main operations used in text simplification.

Early work on simplification used a rule based approach to splitting (Siddharthan, 2006; Siddharthan and Mandya, 2014). For instance, (Siddharthan, 2006) defines 26 handcrafted rules for simplifying apposition and/or relative clauses in dependency structures and 85 rules to handle subordination and coordination.

Further work focused on learning statistical simplification models from parallel datasets of complex-simplified sentences derived from English Wikipedia and Simple English Wikipedia. (Zhu et al., 2010) introduces a syntax-based machine translation model where splitting probabilities are learned from syntactic structure. (Woodsend and Lapata, 2011) induced a grammar from the parallel Wikipedia corpus annotated with syntactic trees and use an integer linear programming model for selecting the most appropriate simplification from the space of possible rewrites generated by the grammar. They report learning 438 rules for sentence splitting. Probabilistic models have also been proposed. (Narayan and Gardent, 2014) determine splitting points using a dedicated probabilistic module trained on the Parallel Wikipedia corpus annotated with semantic structures while (Narayan and Gardent, 2016) extends this approach to an unsupervised setting where splitting points are determined based on the maximum likelihood of sequences of thematic role sets present in the simplified version of English Wikipedia.

More recent work has directly adressed the sentence splitting task. (Narayan et al., 2017) introduce a dataset for training sentence splitting models called WebSplit and report results for various neural models trained on this data, comparing a vanilla sequence-to-sequence model with a multi-source and a semantically informed model. (Aharoni and Goldberg, 2018) present an alternative train/dev/test partition for WebSplit which better supports generalisation and show that adding a copy mechanism helps improve results. One limitation of the WebSplit corpus is that it uses a small vocabulary. To remedy this shortcoming, (Botha et al., 2018) create a new dataset called WikiSplit by mining Wikipedia's edit history. WikiSplit contains one million naturally occurring sentence splits. The authors show that incorporating WikiSplit as training data produces a model which outperforms prior results on the WebSplit test data by 32 BLEU points.

While these efforts are focused on syntax- or semantic-based sentence splitting, our work targets discourse-based sentence splitting.

Closest to our work, (Niklaus et al., 2019b,a,c) defines a set of 35 hand-crafted transformation rules to recursively decompose sentences into a hi-

[1] Our code and data is available at https://github.com/liamcripwell/disco_split.

erarchical structure relating core sentences linked via rhetorical relations. They do not generate a well-formed text and the proposed rule-based approach will fail to easily generalise to other languages. Furthermore, because they focus on producing sentences representing minimal semantic units, their system outputs contain a large number of very short sentences which poses some readability issues. In contrast, we present a dataset for training discourse splitting models and Transformer-based, encoder-decoder models for generating discourse splits. The included examples exhibit a single split per sentence and do not rely on a deep hierarchical representation of the discourse structure, thereby preserving readibility.

3 Tasks and Data

3.1 Tasks

We focus on cases of discourse-splitting such as illustrated in the top tier of Table 1, where the input text $C1$ includes a discourse connective ("after this") denoting a discourse relation between two discourse units and the split output includes a corresponding discourse adverbial ("Afterwards" in S2, "Before this" in S3)[2]. We refer to the discourse tree representing the discourse structure of both C and S as T.

We consider two approaches: an end-to-end approach where the model directly splits the input text C into two shorter sentences S; and a pipeline approach where we first map C to a discourse tree T and then map this tree to the split output S.

3.2 Data

We create (C, S) pairs using both synthetic and organic, parallel data. We then extend these pairs to (C, T, S) triples using rule-based and discourse parsing techniques to create the associated discourse tree T.

3.2.1 Creating C/S Pairs

Organic, Parallel Data. We create this data by extracting discourse-split instances from two existing datasets, WikiSplit and MUSS.

WikiSplit (Botha et al., 2018) is a sentence splitting dataset containing 1M single sentences alongside a two sentence variant which preserves their original meaning. This data is extracted from Wikipedia edit history, and therefore contains organic instances of C to S transformations.

The multilingual unsupervised sentence simplification dataset (MUSS) (Martin et al., 2020) contains 2.7M pairs of text sequences mined from Common Crawl web data which were estimated to be paraphrases of each other using L2 distance on LASER embeddings. Filtering out only those pairs that represent a splitting operation yields a subset of 157K examples. Like WikiSplit, this dataset is organically human-authored.

To create a discourse splitting dataset, we then extract from these two datasets all instances such that either the input contains a discourse connective or the output contains a discourse adverbial. We consider the discourse relations specified in the Penn Discourse Treebank (PDTB) and select a subset of these which we determined to be commonly represented via an adverbial connective between two sentences. We then compile a set of intra-sentential connective analogues for each. Table 2 shows the set of discourse connectives and adverbials used together with their corresponding discourse relations.[3] They cover 7 out of the 15 second order relations occurring in the PDTB.

Synthetic Data. The Common Crawl News corpus (CC-News) (Nagel, 2016) is a large collection of news articles that have been scraped from the internet. We use the news-please (Hamborg et al., 2017) python library to mine (i) a set of 1 million sentence pairs (D-CC-News-S) whose second sentence contains an adverbial and (ii) a set of 800K sentences (D-CC-News-C) which contain a discourse connective. We then create the corresponding input text (C) and discourse tree (T) for each sentence in D-CC-News-S, and the corresponding discourse tree for each sentence in D-CC-News-C using rules and a discourse parser, as explained in the following section.

3.3 Creating (C,T,S) Triplets

We use discourse trees (i) to derive (C, T, S) triplets from the parallel data and (ii) to create matching C texts for the S texts in D-CC-News-S.

Creating Discourse Trees. For a given S, we employ the following rule-based method to derive a linearized tree of the form shown in Table 1, T. The adverbial is removed from the sentence pair and

[2] We leave for future work cases where the input contains multiple or implicit discourse relations.

[3] We manually performed the mapping of discourse relations to a set of adverbials and equivalent connectives by studying the PDTB manual and examples from existing splitting datasets.

mapped to the corresponding PDTB discourse relation while the two sentences are used as the tree's arguments and are rearranged into the linearized tree for the relation instantiated by the adverbial. The ordering of the arguments is determined according to a defined schema for each relation, as stipulated in the PDTB manual. Table 1, T shows an example of this for the *Temporal.Asynchronous* relation, where the arguments are ordered chronologically.

To create a discourse tree for a complex sentence C occurring in D-CC-News-C, we use the (Lin et al., 2014) end-to-end PDTB discourse parser. Although not the most recent discourse parser, we chose it because it is publicly available as a simple to use end-to-end system and specifically uses the PDTB schema. However, we noticed that the parser often fails to extract the arguments of the relation, so we also fall back to using a naive extraction strategy in such cases. This naive approach works by selecting the content on either side of the connective as the relation arguments.

In both cases (deriving a discourse tree from a complex sentence C or from a pair of sentences S), the created discourse tree is similar to a PDTB discourse tree in that it uses the PDTB inventory of discourse relations and order their arguments according to the PDTB annotation guidelines.

Deriving Complex Sentences from pairs of Simple Sentences. We derive a single sentence variant C from a sentence pair S in D-CC-News-S using a simple rule-based method which fuses the pair while maintaining the appropriate discourse relation and instantiating different possible argument orderings and connective alternatives. This process works by first randomly selecting a connective from the set of possibilities, given the adverbial in S, and then combining it along with the two arguments to form a single sentence. These combinations are of the form "arg1 connective arg2" or "connective arg1, arg2", depending on the selected connective.

This method only partially captures possible variations between C and S due to C being constructed from S using simple rules that do not take into account lexical variability (paraphrasings, etc.) that can exist for organic examples. However, as shall be shown in Section 6, because it permits creating multiple discourse variants of the same discourse split S using different connectives and orderings, this synthetic data helps to train discourse splitting models that are better able to generalise, such that they can generate different constructions for the same relation.

We do not attempt to automatically derive S from C for D-CC-News-C as this is a more complex task requiring many more alterations to reliably produce coherent samples. For instance, when there is a connective at the beginning of the sentence, it is difficult to identify which parts of the remaining sentence constitute the individual relation arguments. Additionally, rewriting and coreference resolution regularly need to be performed.

3.4 Training and Test Data

Table 3 summarises the data used for training and development. For evaluation, we extracted a set of 352 (C, T, S) triples from the organic datasets (184 triples from WikiSplit and 168 from MUSS), making sure to maintain an approximately even distribution over the supported connectives. To ensure a high level of quality, we then manually corrected the contents of T, C, and S, where necessary i.e., when C and S connectives did not match or when the wrong parts of the text have been flagged as relation arguments in T.

4 Models

Given a complex sentence C with discourse tree T and split output S, we consider and compare two approaches: an end-to-end approach $C2S$ where the split output S is directly generated from C; and a pipeline approach PL which uses C's discourse tree to mediate the split i.e., first mapping C to its discourse tree T and second, mapping this tree to the split output S. We try both of these approaches in order to investigate how difficult it is for an end-to-end model to incorporate the discourse structure on its own and to what extent, if any, explicit mediation of this information aids the performance of discourse-based splitting.

For each of these two approaches, we explore different ways of combining the training data: using only the synthetic data (Synth), only the organic data (Organic) or both (Synth+Organic). We also investigate a pre-training and fine-tuning approach where we pre-train on the synthetic data and fine-tune on the organic data; and a multi-task learning approach where we multi-task on the intermediate mapping tasks (mapping C to T and mapping T to S) and on the end-to-end task (mapping C to S).

D-Reln	D-Con	D-Adv
TEMPORAL:Asynchronous	"and afterwards", "but afterwards", "after which", "then", "after that", "after this", "but, after that", "and after this", "after which", "eventually", "and eventually", "and in turn", "in turn", "which, in turn", "and then", "and so", "later", "and later", "but later", "next", "before", "followed by", "when", "thereafter", "and thereafter", "after which", "before that", "but before that", "although before that", "prior to this", "earlier", "and earlier", "formerly", "previously", "after", "and previously", "recently"	"afterward(s)", "after that", "eventually", "in turn", "later", "next", "thereafter", "before that", "earlier", "previously"
TEMPORAL:Synchrony	"in the meantime", "but in the meantime", "whilst", "meanwhile", "while in the meantime", "while", "simultaneously", "and simultaneously"	"in the meantime", "meanwhile", "simultaneously"
CONTINGENCY:Cause	"accordingly", "so", "as such", "and as such", "as a result", "and as a result", "however", "so that", "resulting in", "consequently", "and therefore", "and so", "with", "therefore", "which means", "which means that", "thus", "and thus", "thusly"	"accordingly", "as a result", "consequently", "therefore", "thus"
COMPARISON:Contrast	"by comparison", "in comparison", "while", "compared to", "whilst", "by contrast", "in contrast", "and in contrast", "while", "although", "conversely", "and conversely", "nevertheless", "but", "none the less", "yet", "however", "on the other hand", "and on the other hand", "but on the other hand", "but", "whereas"	"by/in comparison", "by/in contrast", "conversely", "nevertheless", "on the other hand"
TEXPANSION:Conjunction	"additionally", "and additionally", "and also", "and is also", "besides", "besides this", "aside from", "further", "furthermore", "and furthermore", "and further", "in addition to", "likewise", "and likewise", "moreover", "indeed", "similarly", "and similarly", "while"	"additionally", "also", "besides", "furthermore", "in addition", "likewise", "moreover", "similarly"
EXPANSION:Instantiation	"for example", "for instance", "such as", "in particular"	"for example", "for instance", "in particular"
EXPANSION:Alternative	"instead", "but instead", "though", "but rather", "rather"	"instead", "rather"

Table 2: Discourse Relations, Connectives and Adverbials

Dataset	**# Instances**	**Discourse Relation**						
		Temporal		**Contingency**	**Comparison**	**Expansion**		
		Async	Sync	Cause	Contrast	Conj	Inst	Alt
D-MUSS	31,417	10,382	3,744	4,294	7,468	3,534	1,526	236
D-WikiSplit	371,117	192,798	21,076	36,086	59,729	44,739	10,346	6,343
Total Organic	402,534	203,183	24,820	40,380	67,197	48,273	11,872	6,579
D-CCNews-C	817,316	262,466	55,270	116,341	288,123	63,599	25,349	6,168
D-CCNews-S	999,437	113,298	150,105	102,956	69,864	345,189	137,178	80,847
Total	**2,219,287**	**578,947**	**230,195**	**259,677**	**425,184**	**457,061**	**174,399**	**93,594**

Table 3: Discourse Split Training Data (# Instances: Number of (C, T) pairs for D-CCNews-C, number of (C, T, S) triples for all other datasets, Conj:Conjunction, Inst:Instantiation, Alt:Alternative). The top tier describes the organic discourse data extracted from MUSS and WikiSplit, the second tier the synthetic data derived from CC-News

5 Experimental Setup

All of our generative models use the BART architecture (Lewis et al., 2020) and were trained on a computing grid using 4 Nvidia RTX 2080 Ti GPUs. Each experiment starts by fine-tuning the *facebook/bart-base* model hosted by Hugging-

Face [4], which has 6 layers in each of the encoder and decoder, a hidden size of 768, and was pre-trained to perform reconstruction of corrupted documents on a combination of books and Wikipedia data.

During training, we used a learning rate of $3e^{-5}$, a batch size of 16, and performed dropout with a rate of 0.1 and early stopping as regularisation measures. For each experiment we set aside 5% of the training set for validation. During generation, we perform beam search with a beam size of 4.

We compare the following models:

Split Baseline (BL_{Split}) Pre-trained BART fine-tuned on a 1M example dataset of both syntax- and discourse-based splittings (WikiSplit). This baseline allows us to compare training with very large heterogeneous training data (BL_{Split}) vs. learning from smaller, discourse-split data (BL_{DSplit}).

Discourse-Split Baseline (BL_{DSplit}) Pre-trained BART fine-tuned on a discourse-focused subset of WikiSplit (D-WikiSplit). This baseline is to be directly compared with BL_{Split}.

Parser Pipeline Baseline (PL_{Parse}) A pipeline of two models. The first uses the discourse parser process used to generate Ts from Cs in Section 3 (C2T) and the second is a pre-trained BART fine-tuned on (T, S) data (T2S). We experimented training the T2S component on various datasets and found the best to be that trained purely on synthetic data. Thus, any pipeline mentioned in the remainder of this paper refers to a specific C2T component connected to this same T2S component. This baseline allows us to compare pipeline models whose C2T component is learned on the split data vs. one where the C2T component uses an existing discourse parser.

End-to-End Model (E2E) Pre-trained BART fine-tuned on discourse-split data. We report results for variants trained on D-CC-News-S ($E2E_{Synth}$), D-Wikisplit and D-MUSS ($E2E_{Organic}$), and all three combined ($E2E_{Both}$).

Pipeline Model (PL) A pipeline of two models. The first model is pre-trained BART fine-tuned on (C, T) data and the second is a pre-trained BART fine-tuned on (T, S) data from D-CCNews-S. We report results for pipelines with a C2T component trained on all D-CCNews data (PL_{Synth}), D-MUSS data ($PL_{Organic}$), and D-CCNews combined with D-Wikisplit and D-MUSS data (PL_{Both}).

Pre-training and Fine-tuning (PT+FT) Pre-trained BART fine-tuned on one data set before being further fine-tuned on another. We try training first on either synthetic or standard WikiSplit data and then fine-tuning on D-WikiSplit and D-MUSS data. Using WikiSplit for the first step was found to be the best performing configuration for the end-to-end system ($E2E_{ptft}$), while using D-CCNews proved better for the pipeline (PL_{ptft}).

Multi-Tasking (MTL) We prefix the training data with a control token indicating whether a training instance maps a complex input to a discourse tree (c2t), a discourse tree to a split text (t2s) or a complex input to a split output (c2s) and train pre-trained BART on this data. We use training examples from D-CCNews, D-WikiSplit and D-MUSS. At inference time, we prefix the input with the c2s control token for the end-to-end model; and with the c2t and t2s control tokens for the two components of the pipeline model.

5.1 Evaluation Metrics

As illustrated in Table 4, variants of a discourse split may differ in terms of sentence order, discourse connnective and rephrasing. To account for such variants while automatically assessing meaning preservation and discourse structure in the generated output, we use a combination of metrics.

Meaning Preservation. We measure meaning preservation using BLEU-4 and SAMSA. We calculate BLEU scores (Papineni et al., 2002) between the ground-truth reference and the generated text using the SacreBLEU library (Post, 2018). We use the EASSE python library (Alva-Manchego et al., 2019) to compute SAMSA scores. SAMSA (Sulem et al., 2018) aims to put more focus on the structural aspects of the text, by leveraging a semantic parser. It observes changes made to predicate-argument structures, and thus for the sentence *"John got home and gave Mary a call."*, a higher score will be given to *"John got home. John gave Mary a call."* than for *"John got home and gave. Mary called."*. This indicates whether a model actually produces semantically coherent splits irrespective of whether a valid discourse connective and order is used. [5]

[4] https://huggingface.co/facebook/bart-base

[5] Despite SAMSA specifically targeting minimal units, while our systems aim to only perform a single split, we

Discourse Structure. To evaluate discourse structure we compute connective-, relation- and discourse-structure accuracy. Connective-accuracy (Conn-ACC) is the proportion of cases in which the generated text contains the same adverbial as the reference and relation-accuracy (Rel-ACC) the proportion of cases which maintain the discourse relation. The difference between Rel-ACC and Conn-ACC indicates how well the model is able to generalise amongst equivalent connectives of the same relation.

We also introduce a custom binary metric (D-ACC) which classifies an output as positive if (i) the correct discourse relation is maintained, (ii) the sentences are correctly ordered, and (iii) there is sufficient semantic similarity between the generated text and the ground-truth. A text will have a D-ACC score of 1 if it has a high BLEU (BLEU > 0.5) and either a low sentence BLEU (S-BLEU < 0.1) with a discourse adverbial which reverses the order of the argument (Table 4, Ex. 2) or a high sentence BLEU and a discourse adverbial which preserves the input discourse relation (Table 4, Ex. 2 and 3). Conversely, outputs with low BLEU and outputs with high BLEU, low S-BLEU and the same discourse connective as the reference (Table 4, Ex. 4) will be assigned a score of 0.[6]

We treat SAMSA and D-ACC as our primary metrics for comparing performance between models as, together, they provide an evaluation of both the meaning preservation and coherence of the split as well as the preservation of the discourse structure. Table 4 shows several example outputs and their corresponding scores.

5.2 Human Evaluation

In addition to using automated metrics, we performed human evaluation to compare our highest performing models and baseline systems using the MTurk platform. We considered a subset of 96 randomly selected examples from our test set (12 from each discourse relation type) and presented human annotators with the generated text for that example from our best performing pipeline system (PL_{Synth}) and asked them to compare it with (a) the result from BL_{Split} (trained on generic split data), (b) the ground-truth result with adverbial removed, and (c) the result from our best performing end-to-end model ($E2E_{Both}$). Each combination was presented to 10 different annotators who were asked to compare the two texts in terms of their grammaticality, as well as how similar in meaning they are to the C input. In total we collected $5,760$ judgments: 960 judgments for each pair of models compared and each criteria (grammaticality vs. meaning preservation). Further details of this process are outlined in Appendix A.2.

We do not compute inter-annotator agreement scores due to some of the complexities in using the crowd-sourcing platform. Specifically, it would require having every annotator complete every comparison task, which is hard to manage at scale when posing each comparison as an individual task. To mitigate this issue, we opted to have a larger number of annotators complete each task, coupled with a larger number of unique tasks, in an attempt to smooth out individual differences.

6 Results and Discussion

Table 5 summarises the results.

Pipeline vs. End-to-End. While no single configuration outperforms all others, PL_{Synth} ranks high for meaning preservation (SAMSA and BLEU) and for discourse structure (D-ACC and D-Rel).

More generally, we see that PL models universally outperform their $E2E$ variant in terms of discourse structure (Rel- and D-ACC). Conversely, the $E2E$ models tend to show better results in terms of meaning preservation (SAMSA and BLEU). This suggests that while the PL models are good at producing valid connectives and the correct sentence order (high D-ACC), their generative capacity needs improvement.

Synthetic vs Organic Data. Another clear trend is that models trained with synthetic data have significantly higher D-ACC than those trained with organic data. This confirms our hypothesis that, because it includes multiple variants of the same discourse split using different connectives and orderings, the synthetic data helps to train discourse splitting models that are better able to generalise i.e., are able to generate with different connectives for the same relation.

believe it is sufficient here as all outputs should contain the same number of sentences and therefore would recieve the same non-split penalty.

[6] We determined the above thresholds of 0.5 and 0.1 empirically via the manual examination of a number of test examples. The S-BLEU threshold is much lower because when the argument ordering is reversed we expect there to be little to no n-gram overlap with the ground-truths.

	Text	BLEU	S-BLEU	SAMSA	D-ACC
Ref.	The Masovians were caught by surprise, since virtually without any defense the capital, Płock, fell. **After this**, Mindaugas crossed the Vistula river and captured the fortress of Jazdów.				
✓1:C	The Masovians were caught by surprise, since virtually without any defense the capital, Płock, fell. **Afterwards**, Mindaugas crossed the Vistula river and captured the fortress of Jazdów.	89.11	92.80	66.66	1
✓2:O,C	Mindaugas crossed the Vistula river and captured the fortress of Jazdów. **Before this**, the Masovians were caught by surprise, since virtually without any defense the capital, Płock, fell.	84.93	3.30	66.66	1
✓3:T	The Masovians were caught by surprise, since the capital, Płock, fell. **After this** had happened, Mindaugas then crossed the Vistula river and captured the fortress of Jazdów.	73.64	65.96	66.66	1
✗4:O	Mindaugas crossed the Vistula river and captured the fortress of Jazdów. **After this**, the Masovians were caught by surprise, since virtually without any defense the capital, Płock, fell.	89.56	4.95	66.66	0

Table 4: Example illustrating how correct and incorrect variants of the reference impact the scores. O indicates that the order of the sentences has been reversed, C that the discourse adverbial differs from that used in the reference, and T that the text has changed. Only D-ACC distinguishes good from bad variants.

Model	Data	**SAMSA**		**BLEU**		Discourse Structure					
						Rel		**Conn**		**D-ACC**	
		$E2E$	PL	$E2E$	PL	$E2E$	PL	$E2E$	PL	$E2E$	PL
PL_{Parse}			46.37		67.25		0.73		0.31		0.65
BL_{DSplit}	D-WikiSplit	53.91	50.27	80.16	71.65	0.46	0.59	0.43	0.26	0.45	0.51
BL_{Split}	WikiSplit	**54.15**		80.09		0.45		0.44		0.45	
	Synth	47.82	49.96	80.90	72.98	**0.57**	**0.69**	0.45	**0.31**	**0.55**	**0.64**
	Organic	53.26	48.15	80.00	68.90	0.47	0.61	0.43	0.27	0.46	0.55
	Both	52.96	50.40	81.31	72.87	0.55	0.63	0.44	0.27	0.54	0.60
PT+FT	Synth/Org	53.97	49.99	**81.64**	73.59	0.50	0.60	**0.47**	0.24	0.50	0.57
MTL	C/T,T/S,C/S	44.97	**52.93**	74.67	**75.55**	0.45	0.52	0.39	**0.31**	0.44	0.51

Table 5: A summary of results. Each row represents the results of the best E2E and PL model for the specified data category.

Models	**Grammaticality**			**Meaning Pres.**		
	>	=	<	>	=	<
PL_{Synth} vs. $E2E_{Both}$	0.21	0.40	0.39	0.15	0.47	0.38
PL_{Synth} vs. BL_{Split}	0.24	0.34	0.42	0.18	0.44	0.39
PL_{Synth} vs. no adv.	0.36	0.29	0.36	0.35	0.35	0.30
PL_{Synth} vs. $E2E_{Both}$				0.06	0.81	0.14
PL_{Synth} vs. BL_{Split}		=		0.09	0.77	0.14
PL_{Synth} vs. no adv.				0.25	0.64	0.11

Table 6: Results for human evaluation. Cells show the proportion of cases where the pipeline was deemed better, equal or worse than a particular baseline.

For both $E2E$ and PL models, combining organic and synthetic data ($E2E_{Both}$ and PL_{Both}) appears to reduce the performance trade-off of using one data type in isolation.

Alternative ways of combining organic and synthethic data using either fine-tuning and pre-training or multi-tasking did not yield improvements. For both regimes, we experimented with multiple hyper-parameters and data combinations. The details of these experiments are given in Appendix A.3.

Generic- vs. Discourse-Split Data In terms of meaning preservation (BLEU, SAMSA), BL_{DSplit} (trained on 371K instances) performs on par with BL_{Split} (1M instances), showing that discourse-focused models can compete with standard splitting models when trained on much smaller, ded-

icated datasets. Moreover, in terms of discourse structure (D-ACC) and generalisation (Rel-ACC, Conn-ACC), $E2E_{Organic}$ has significantly higher generalisation capacity than BL_{Split} ($p = 0.046$). This improvement becomes more dramatic when also including the synthetic data ($E2E_{Both}$) ($p = 8.72e^{-7}$).

Human Evaluation The results from the human evaluation (Table 6) confirm those of the automatic evaluation.

Human annotators find the output of PL_{Synth} less grammatical and meaning preserving than either of the end-to-end models ($E2E_{Both}$ and BL_{Split}). This corroborates the divergence seen between $E2E$ and PL models for SAMSA and BLEU scores.

For meaning preservation, annotators more often selected PL_{Synth} over BL_{Split} than they did PL_{Synth} over $E2E_{Both}$ ($p = 0.138$), strengthening the observation that discourse-focused models perform this task better than generic splitting models.

PL_{Synth} produces texts that are equally grammatical yet significantly more meaning preservative ($p = 0.017$) than the adverbial-stripped ground-truths. This reinforces the importance of maintaining discourse coherence when performing sentence splitting.

Upon examination of human evaluations, we found that annotators often marked the less grammatical text as being less meaning preservative by default. When controlling for this and only considering cases where both texts were labelled as equally grammatical (bottom tier of Table 6), we see improved results for PL_{Synth} such that, in terms of meaning preservation, there is less difference between PL_{Synth} and the end-to-end models and an increased difference between PL_{Synth} and the ground-truth with adverbial removed.

Qualitative Analysis In addition to the automatic and human evaluations, we perform a qualitive analysis of common mistakes seen in system outputs. Table 7 in Appendix A.4 shows some examples of common errors for PL_{Synth}, $E2E_{Both}$ and BL_{Split}.

We can group these mistakes into 4 broad categories: *connective*, *content*, *splitting*, and *hallucinations*. Connective errors are those that use an incorrect connective or lack one entirely. Content errors are cases where the semantic content of the input is not maintained in the output. Splitting errors are cases where splitting has not been performed or has been done in the wrong place. We also occasionally see hallucinations where the output has included out-of-context information.

The BL_{Split} model will often fail to use a valid adverbial, instead merely splitting the sentence at the position of the connective. We believe this is due to it not fully learning to maintain the discourse relation. It has also been observed to include hallucinated terms in the output.

We commonly see splitting errors for both PL and $E2E$ models. The PL often splits at a position containing a known connective term, but where it is not acting as a connective given the context. This is due to the intermediary task incorrectly segmenting the input, possibly as a result of parser mistakes in the training data. On the other hand, the $E2E$ will sometimes not perform any split, particularly where certain grammatical markers (e.g. semicolons) are present.

7 Conclusion

In this paper we introduced the task of Discourse-based Sentence Splitting together with a large-scale dataset of both organic and synthetic discourse splits. Experimental evaluation revealed that discourse-based, pipeline models have better discourse relation preservation capabilities than end-to-end models, and that synthetic data is critical for learning models that can generalise i.e. that can generate mutliple variants of the same discourse relation. In future work, we would like to create more document-aware models incorporating both syntax- and discourse-based sentence splitting at the document level.

Acknowledgements

Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

We thank the anonymous reviewers for their feedback. We thank the authors of (Martin et al., 2020) who kindly provided us with their data for this study. We gratefully acknowledge the support of the French National Research Agency (Gardent; award ANR-20-CHIA-0003, XNLG "Multilingual, Multi-Source Text Generation").

References

Roee Aharoni and Yoav Goldberg. 2018. Split and rephrase: Better evaluation and stronger baselines. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 719–724, Melbourne, Australia. Association for Computational Linguistics.

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. EASSE: Easier automatic sentence simplification evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 49–54, Hong Kong, China. Association for Computational Linguistics.

Jan A. Botha, Manaal Faruqui, John Alex, Jason Baldridge, and Dipanjan Das. 2018. Learning to split and rephrase from Wikipedia edit history. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 732–737, Brussels, Belgium. Association for Computational Linguistics.

R. Chandrasekar, Christine Doran, and B. Srinivas. 1996. Motivations and methods for text simplification. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please: A generic news crawler and extractor. In *Proceedings of the 15th International Symposium of Information Science*, pages 218–223.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Junyi Jessy Li and Ani Nenkova. 2015. Detecting content-heavy sentences: A cross-language case study. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1271–1281, Lisbon, Portugal. Association for Computational Linguistics.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2014. A pdtb-styled end-to-end discourse parser. *Natural Language Engineering*, 20(2):151–184.

Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. 2020. Multilingual unsupervised sentence simplification.

Jana M Mason. 1978. Facilitating reading comprehension through text structure manipulation. *Center for the Study of Reading Technical Report; no. 092*.

Kshitij Mishra, Ankush Soni, Rahul Sharma, and Dipti Sharma. 2014. Exploring the effects of sentence simplification on Hindi to English machine translation system. In *Proceedings of the Workshop on Automatic Text Simplification - Methods and Applications in the Multilingual Society (ATS-MA 2014)*, pages 21–29, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Sebastian Nagel. 2016. Cc-news.

Shashi Narayan and Claire Gardent. 2014. Hybrid simplification using deep semantics and machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 435–445, Baltimore, Maryland. Association for Computational Linguistics.

Shashi Narayan and Claire Gardent. 2016. Unsupervised sentence simplification using deep semantics. In *Proceedings of the 9th International Natural Language Generation conference*, pages 111–120, Edinburgh, UK. Association for Computational Linguistics.

Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. 2017. Split and rephrase. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 606–616, Copenhagen, Denmark. Association for Computational Linguistics.

Christina Niklaus, Bernhard Bermeitinger, Siegfried Handschuh, and André Freitas. 2016. A sentence simplification system for improving relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 170–174, Osaka, Japan. The COLING 2016 Organizing Committee.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2019a. DisSim: A discourse-aware syntactic text simplification framework for English and German. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 504–507, Tokyo, Japan. Association for Computational Linguistics.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2019b. Transforming complex sentences into a semantic hierarchy. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3415–3427, Florence, Italy. Association for Computational Linguistics.

Christina Niklaus, André Freitas, and Siegfried Handschuh. 2019c. MinWikiSplit: A sentence splitting corpus with minimal propositions. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 118–123, Tokyo, Japan. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Advaith Siddharthan. 2006. Syntactic simplification and text cohesion. *Research on Language and Computation*, 4(1):77–109.

Advaith Siddharthan and Angrosh Mandya. 2014. Hybrid text simplification using synchronous dependency grammars with hand-written and automatically harvested rules. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 722–731, Gothenburg, Sweden. Association for Computational Linguistics.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. Semantic structural evaluation for text simplification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 685–696, New Orleans, Louisiana. Association for Computational Linguistics.

Sandra Williams, Ehud Reiter, and Liesl Osman. 2003. Experiments with discourse-level choices and readability. In *Proceedings of the 9th European Workshop on Natural Language Generation (ENLG-2003) at EACL 2003*.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 409–420, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361, Beijing, China. Coling 2010 Organizing Committee.

A Appendices

A.1 Training Details

During this work we ran a range of experiments using different data set combinations. For each of our primary model types ($E2E$, PL) we ran training experiments with solely organic data, solely synthetic data, both together, and various combinations therein. For instance, in the case of solely organic data, we ran separate experiments using D-WikiSplit, D-MUSS, and the two in combination, for both $E2E$ and PL. The highest performing of these was then selected as $E2E_{organic}$ and $PL_{Organic}$.

All of these used the BART architecture with the same fixed hyperparameters, as outlined in the paper. Training and convergence times were quite varied depending on the task and data used, but $E2E$ models were trained on average for $\sim$48 hours, and PL models for $\sim$24 hours.

A.2 Human Evaluation

We perform our human evaluation via the crowdsourcing platform, Amazon Mechanical Turk. We present a web form to evaluators, which includes some example texts and questions they must answer. These forms are referred to as *hits*. In our case, each hit contains three pieces of text (A, B, and X). These are the output from PL_{Synth} for a given test example, the output from one of our three comparators (BL_{Split}, $E2E_{Both}$, and the ground-truth with no adverbial) for the same example, and the input C, respectively.

Evaluators are then asked to answer the following questions:

- Which text (A or B) has more grammatical/fluent/well-formed English?
- Which text (A or B) is most similar in meaning to X?

For each of the two questions they must answer with either *A*, *B*, or *Equal*. For each of the 96 selected test examples, we performed 3 model comparisons and sourced 10 separate evaluators for each, meaning we had 2,880 hits completed. Each of these hits gives us 2 judgements (one for grammaticality and one for meaning preservation), thus we recieved 5,760 individual judgements. We paid \$0.06 USD for each hit, meaning we spent \$172.8 USD in total.

An example of how one of these hits looks to evaluators can be seen in Figure 1.

A.3 Fine-tuning and Multi-Task Learning Experiments

In this work, we experimented with various fine-tuning and multi-task learning regimes in order to see if further performance gains could be met.

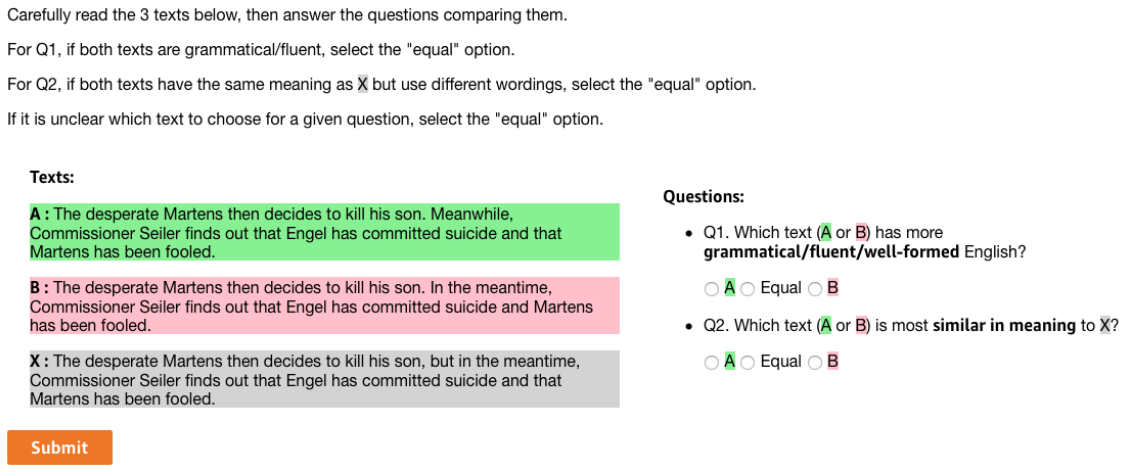

Figure 1: An example human evaluation hit for a single test example.

In the case of multi-task learning, we trained a sequence-to-sequence model to simultaneously learn to perform the C2T, T2S and C2S tasks. The motivation behind this was that there could potentially be useful shared features between the tasks that would help overall learning performance. We used C2T data from D-CCNews, D-MUSS, and D-WikiSplit; and both T2S and C2S data from D-CCNews-S, D-MUSS, and D-Wikisplit.

For our pre-training and fine-tuning experiments we experimented with different dataset combinations and training strategies for both our end-to-end model and our pipeline system. Initially, we tried pre-training first on synthetic data and then finetuning on organic data; either as D-MUSS and D-WikiSplit in combination, or one after the other. We also went on to experiment with pre-training on the standard WikiSplit dataset in an attempt to see whether useful features could be learned from training on a generic splitting task. For each of these, we also experimented with freezing/unfreezing different layers in the network (embedding, encoder, and decoder).

As mentioned in the paper, we were unable to observe any improvements over our standard models for any of these experiments. In the case of our pre-training and fine-tuning experiments, the best strategy we found for the end-to-end model was to simply train the BART architecture on standard WikiSplit and further finetune on D-MUSS and D-WikiSplit in combination. For the pipeline system, this was to train on the D-CCNews data, fine-tune on D-MUSS, then further fine-tune on D-WikiSplit. The performance of these models are reported in the paper, but, as can be seen, they failed to outperform other experiments in their respective categories.

A.4 Generation Examples

Table 7 shows example outputs from several models (PL_{Synth}, $E2E_{Both}$, BL_{Split}) for a range of different example inputs. We try to showcase various ways each of the models can fail.

Model	Text	Good?
	Girls raised by working moms are more likely to be successful in life, while sons raised in homes with working mothers spend their adulthood caring for family members.	
PL_{Synth}	Girls raised by working moms are more likely to be successful in life. **In contrast,** son sons raised in homes with working mothers spend their adulthood caring for family members.	✓
$E2E_{Both}$	Girls raised by working moms are more likely to be successful in life. **By contrast,** sons raised in homes with working mothers spend their adulthood caring for family members.	✓
BL_{Split} (*connective*)	Girls raised by working moms are more likely to be successful in life**. While** sons raised in homes with working mothers spend their adulthood caring for family members.	✗
	These guidelines have been developed with the recognition that Internet technologies are rapidly evolving, and accordingly, guidelines are subject to change.	
PL_{Synth}	Internet technologies are rapidly evolving. **Therefore,** guidelines are subject to change.	✓
$E2E_{Both}$	These guidelines have been developed with the recognition that Internet technologies are rapidly evolving. **Accordingly,** guidelines are subject to change.	✓
BL_{Split}	These guidelines have been developed with the recognition that Internet technologies are rapidly evolving. **Accordingly,** guidelines are subject to change.	✓
	A thing which does not exist in another thing by the self of the latter is not produced from that other thing; for instance, oil is not produced from sand.	
PL_{Synth} (*content*)	A thing which does not exist in another thing by the self of the latter is not produced from that other thing. For example, **oil is produced** from sand.	✗
$E2E_{Both}$ (*splitting*)	A thing which does not exist in another thing by the self of the latter is not produced from that other thing**; for instance,** oil is not generated from sand.	✗
BL_{Split}	A thing which does not exist in another thing by the self of the latter is not produced from that other thing. **For instance,** oil isn't produced from sand.	✓
	He was a mighty hunter before the Lord; therefore it is said, "Like Nimrod a mighty hunter before the Lord."	
PL_{Synth} (*splitting*)	He was a mighty hunter**. Next,** the Lord; therefore it is said, "Like Nimrod a mightyHunter before the Lord.	✗
$E2E_{Both}$	He was a mighty hunter before the Lord. **Therefore,** it is said, "Like Nimrod a mighty Hunter before theLord."	✓
BL_{Split} (*hallucination*)	He was a mighty hunter before the Lord **Anthem**. Therefore it is said, like Nimrod, a mighty hunters before the lord Anthem.	✗
	However, Colorado voters denied funding by a 3:2 margin in November 1972 and, three months later, the games were awarded instead to Innsbruck in Austria.	
PL_{Synth} (*splitting*)	However, Colorado voters denied funding by a 3:2 margin in November 1972 and, **three months. Eventually,** the games were awarded instead to Innsbruck in Austria.	✗
$E2E_{Both}$	However, Colorado voters denied funding by a 3:2 margin in November 1972. **Three months later,** the games were awarded instead to Innsbruck in Austria.	✓
BL_{Split}	However, Colorado voters denied funding by a 3:2 margin in November 1972. **Three months later,** the games were awarded instead to Innsbruck in Austria.	✓

Table 7: Example generated texts illustrating the performance of each model in various contexts.