CBR assisted context-aware surface realisation for data-to-text generation.

UPADHYAY, A. and MASSIE, S.

2023

CBR Assisted Context-Aware Surface Realisation for Data-to-Text Generation

Ashish Upadhyay[1,2]* and Stewart Massie[1]

[1] Robert Gordon University, Aberdeen, UK
[2] J.P. Morgan Chase & Co., Glasgow, UK
{a.upadhyay,s.massie}@rgu.ac.uk

Abstract. Current state-of-the-art neural systems for Data-to-Text Generation (D2T) struggle to generate content from past events with interesting insights. This is because these systems have limited access to historic data and can also hallucinate inaccurate facts in their generations. In this paper, we propose a CBR-assisted context-aware methodology for surface realisation in D2T that carefully selects important contextual data from past events and utilises a hybrid CBR and neural text generator to generate the final event summary. Through extensive experimentation on a sports domain dataset, we empirically demonstrate that our proposed method is able to accurately generate contextual content closer to human-authored summaries when compared to other state-of-the-art systems.

Keywords: Textual Case-Based Reasoning, Data-to-Text Generation, Content Selection, Surface Realisation

1 Introduction

Data-to-Text Generation (D2T) summarises complex insights extracted from non-linguistic structured data into textual format [10,3]. D2T systems address two main problems: content planning, to outline the summary plan; and surface realisation, using the plan to generate the final textual summary [17,7]. D2T problems consist of a series time-stamped events where a textual summary is written for each event. The summaries can be rich and may also contain contextual information derived from past events in the time-series [14]. For example, the excerpt of a basketball summary shown in Figure 1, shows the contextual content derived from past event's data (bold-faced).

Current state-of-the-art neural systems despite being able to generate fluent and human-looking texts often hallucinate with inaccurate generations. They struggle to generate contextual content from past events which is often included in human generated summaries. In this paper, we present a CBR-assisted methodology for including context-aware content in the surface realisation stage.

Context aware content is generated in a two stage process. First, machine learning is used to select potential context-aware content on selected themes

[*] Work done during time at Robert Gordon University, Aberdeen

TEAM	WIN	LOSS	PTS	FG_PCT	REB	AST	...
Pacers	4	6	99	42	40	17	···
Celtics	5	4	105	44	47	22	···

PLAYER	H/V	AST	REB	PTS	FG	CITY
Myles Turner	H	1	8	17	6	Indian
Thaddeus Young	H	3	8	10	5	Indian
Isaiah Thomas	V	5	0	23	4	Boston
Kelly Olynyk	V	4	6	16	6	Boston
...						

The Boston Celtics defeated the host Indiana Pacers 105-99 at Bankers Life Fieldhouse on Saturday. **It was the second victory over Pacers for the Celtics this season after emerging victorious in Boston 91-84 on Nov. 16**. ... Isaiah Thomas led the team in scoring, totaling 23 points and five assists on 4–of–13 shooting. Kelly Olynyk got a rare start and finished second on the team with his 16 points, six rebounds and four assists. ... Boston will return to action on Monday against the New Orleans Pelicans.

Fig. 1: Input table and output summary from a basketball game [12]

from previously occurring events. Then a case-based approach is used to select the specific content that gets merged with a textual summary generated by a state-of-the-art neural system.

In this work, we develop full summaries including both the content planning and surface realisation stages. For content planning, we employ a previously developed approach in [13]. But have developed novel approach for surface realisation. The key contributions are:

- developing machine learning approach for selecting potential context-aware content for the selected themes;
- employing a case-based approach to identify relevant templates that are used to select the specific content examples for an event summary; and
- a human-based evaluation of our approach to measure the accuracy of the generated summaries;

The rest of the paper is organised as follows. We first present some literature that considers different approaches to D2T in the related works, and then discuss some background information. We continue to discuss our methodology in two sections: first, content selection, to outline the process of selecting important and relevant historic content; and then surface realisation, the process of utilising the CBR method in generating the textual summary. The experiment setting and results are discussed next, before finally finishing with conclusions and future directions.

2 Related Works

Data-to-Text Generation is the process of summarising non-linguistic structured data into a textual summary as compared to Text-to-Text Generation that aims to generate textual summaries from linguistic input [3]. Traditional approaches to D2T have solved the task in modular fashion with multiple modules solving different sub-tasks [9]. Recent advancements in neural systems have approached

D2T as an end-to-end system as well as in a modular manner but with evidence backing in favour of latter [7,2,17].

Traditional rule-based D2T systems use domain-specific engineered rules and templates in different modules while recent neural systems use data-driven learning based approach for text generation. Rule-based systems produce high quality texts in terms of accuracy but are often monotonous and lack diversity. In contrast, neural systems are able to generate fluent and human-like texts but hallucinate with inaccurate generations. On the other hand, CBR systems are able to complement both types of systems by employing a data-driven dynamic template approach that is able to generate accurate as well as fluent and diverse texts [15,13].

There has been some work that consider the historic aspects of time-stamped event summaries in D2T domains. Authors in [14] propose a typology of content type in human authored D2T summaries and empirically demonstrate the struggle of neural systems in generating content of historic type. Few earlier works, both in neural as well as traditional systems, have tried to include some form of historic content in final event summary with different methods [4,11]. However, these still struggle with the fundamental problems of accuracy vs diversity trade-off.

In our work, we propose a method of content selection for selecting important historic events that can be utilised by any neural system. We then propose a CBR-inspired surface realisation method that uses both neural and CBR systems in a collaborative manner to improve the accuracy of the generations without harming the fluency.

3 Background

The content of the event summaries generated from D2T problems can typically be broken down and classified into three categories: **Intra-Event Basic (B)**, facts directly copied from the current event's input data; **Intra-Event Complex (C)**, facts derived from the current event's input data; **Inter-Event (I)**, facts copied or derived from other events' data (see Figure 3) [14].

The process of generating an event summary consists of multiple stages: content planning, planning the layout of summary's content; content selection, selecting important content from the input data to display in the summary according to the plan; surface realisation, taking the selected important content in accordance with the content plan and generating textual summary.

The content plan is a list of placeholders denoting the organisation of the summary, while the content selection selects a subset of data (either verbatim or derived) from the input data. These steps have been usually performed separately, however recent neural models have also combined: either all three in a single step [6,17,8]; or content planning and selection into one step and surface realisation into another [5,7].

Authors in [13] proposed a CBR approach to content planning in D2T where the plan (the case solution) is a sequence of concepts represented by the sentence

Sentence	Entities	Content Types	Concept
Sixers came out in domination mode in the third and outscored Bulls, 37-18, to take a 102-76 lead heading into the fourth.	Team, Team	Complex	$T\&T - C$
Bulls put up a fight in the fourth but the Sixers were able to cruise to their first win of the season without a problem.	Team, Team	Complex, Inter	$T\&T - C\&I$
Joel Embiid led the Sixers with 30 points on 9-of-14 shooting, in 33 minutes of action.	Player, Team	Basic, Complex	$P\&T - B\&C$
⋮	⋮	⋮	⋮
Bobby Portis is averaging 20 points and 10 rebounds on the season.	Player	Inter	$P - I$

Content-Plan: $\{\ T\&T - C,\ T\&T - C\&I,\ P\&T - B\&C,\ \cdots,\ P - I\ \}$

Fig. 2: Content-plan of a summary taken from SportSett dataset

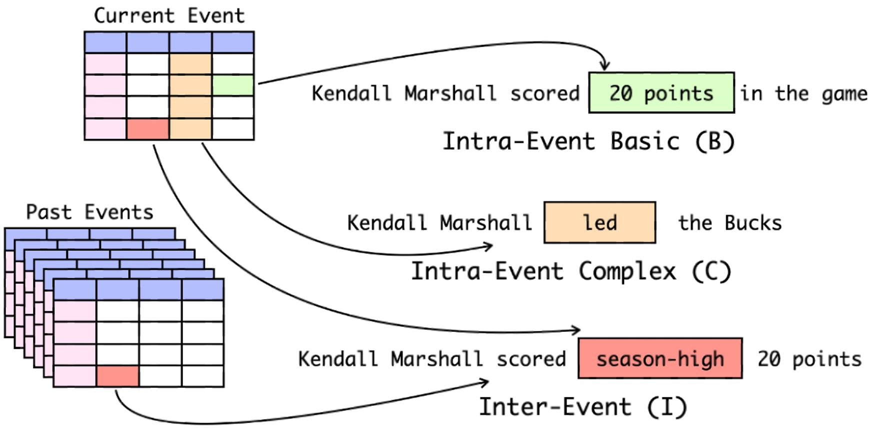
Current Event
Kendall Marshall scored 20 points in the game
Intra-Event Basic (B)
Past Events
Kendall Marshall led the Bucks
Intra-Event Complex (C)
Kendall Marshall scored season-high 20 points
Inter-Event (I)

Fig. 3: Content Types in a human written D2T summary

structure of the summary. Figure 2 shows the content plan extracted from a basketball summary using this approach. The content plan is a list of concepts, each denoting a sentence structure conveying the entity and content type to describe in that sentence. As denoted, the first sentence in the summary should describe two team entities with intra-event complex type content.

In this paper, we take the content plan generated from this approach and propose a method to generate the final event summary according to the plan. The method works in two stages: first, content selection, where we use a novel technique to select important historical data in order to generate inter-event type content; and second, surface realisation, where we use a hybrid method of neural and CBR systems for text generation.

4 Content Selection

Content Selection is an important stage in the data-to-text generation process. It is the process of selecting a subset of input data, either verbatim or derived, to include in the final summary. To generate any inter-event content in the final summary, a D2T system needs to process data from all previous events in the event time-series where any entity from the current event was involved. This massively increases the amount of data that needs to be processed by the system during run-time.

In this section, we describe our methodology to organise and generate all possible inter-event (historic) content from the historic events and then select the important ones to be included in the summary. This method of content-selection involves the following steps:

- **Finding Possible Inter-Event Themes**: The first step is to identify the possible themes that convey inter-event information about entities present in the event.
- **Building Resources for these Themes**: The next step is to develop a parallel resource for these inter-event themes that can be queried to get information during the run-time processing to generate an event summary.
- **Select the Inter-Event Themes to include in the summary**: Finally, for each summary, during run-time processing, select the important inter-event features that should be included in the final textual summary.

Each of these content selection steps are now discussed in more detail.

4.1 Finding Possible Inter-Event Themes

The first step in selecting inter-event content is to identify some popular themes that are commonly discussed in the event summaries. To find these we perform some analysis on the data by applying the following steps:

- break the summaries into sentences and then classify the sentences into their content-types (as in Figure 3);
- take the sentences classified as containing 'inter-event' and divide them into different entity types (in sports domains: players and teams)
- apply topic modelling on sentences from each entity type and select the top topics;

By this process, we select a dominant topic from each of the entity type (player and team), which are:

1. **Players' Average Stats**: player A is averaging X points in last Y games;
2. **Teams' Win/Loss Streak**: this was team B's j^{th} straight loss/win;

In our topic modelling, we also found some other common themes such as: player's total double-double scores [3] of the season; or, team's standing in the

[3] https://en.wikipedia.org/wiki/Double-double

league/conference. However, we decide to experiment with only two themes selected above to keep the problem complexity simple and evaluate the idea properly.

4.2 Building Resources for these Themes

Once the inter-event themes have been identified, the next step is to build some parallel resources that can be used during run-time to query and get the information about a theme for an entity in an event. This parallel resource will store the inter-event information relating to the theme for each entity in the event.

We first identify a few inter-event features that will be used to represent the entities along with their existing intra-event features. For the player average theme, the features chosen are: *average/total X in last Y games*, where X $\in$ (points, rebounds, assists, blocks, steals) and Y $\in$ (2, 10). For the team streak theme, the features selected are: streak count, and streak type, where streak count $\in$ (0, 82) and streak type can be win or loss.

After identifying these features, for each entity from every event in the dataset, we generate the values for these identified inter-event features and store them into a separate parallel resource for each theme (currently json, but a better choice could be a relational database). The process of generating the values for these features is as follows:

- **Filter**: filter all the events from time-series containing a given entity and happening before the current event;
- **Sort**: sort these events based on the timestamp in ascending order of time delta, where the most recent event is the closest; and
- **Aggregate**: aggregate all the relevant values of the entity feature into the identified inter-event feature;

As an example, consider an event which is the 25th match for player Kevin Durant. To calculate his average points in last 5 games: we first filter all the matches from this season in which Durant played and the match happened before this one; we then sort these matches based on their date and then average the number of points made by Durant in the most recent 5 games.

4.3 Selecting Important Attributes

After building the parallel resource, the next step is to select the inter-event features from each theme that could be included in the final summary. This is done by training a binary classifier for each theme whose task is: given an inter-event feature for an inter-event theme, classify if it should be added to the final summary or not.

To build a theme classifier, an important step is to identify attributes needed to train these classifiers. Through our domain knowledge, we identify the following attributes for the two themes:

- Player average theme

  - **player name** - converted into a number using label encoding;
  - **player popularity** - calculated as the ratio of number of game summaries mentioning the player to the number of games the player has played;
  - **record type** - label encoded value for point, rebound, assist, steal or block;
  - **last Y games** - number of games totalling or averaging for (2 to 10)
  - **value** - actual value of the inter-event feature
  - **average or total** - a binary attribute denoting if this average score or total score over last Y games
- Team streak theme
  - **team name** - converted into a number using label encoding;
  - **team popularity** - ratio of the number of sentences mentioning the team to the number of sentences in the summary averaged over toal number of games in the season;
  - **streak count** - count of the streak;
  - **streak type** - a binary value denoting if this is a win or loss;
  - **broken streak count** - denoting if the team has been on a different streak than current result (if there has been a winning streak before if the current one was the lost game)
  - **broken streak type** - type of the broken streak

We build the train and test set for these theme classifiers using the train and validation set of D2T dataset respectively.

5 Surface Realisation

Now with the important inter-event content selected, we move on to using this content in accordance to the plan derived from [13] to generate the final summary. This stage of text generation in D2T is known as surface realisation. Earlier studies have shown that neural networks are capable of producing good content for intra-event types (both basic and complex), however struggle in producing content of inter-event type [14]. Thus in this work, we propose two alternative methods to improve the inter-event content of summaries generated by neural systems.

- **Input Augmentation**: the first approach is to augment the input of neural system by adding the content plan and selected inter-event content to its input and train the model to generate summaries with better coverage; and
- **Post-Editing**: the second approach is to further post-edit the output of neural system by identifying the sentences with inter-event content and replacing those with sentences generated using the CBR-D2T dynamic template method from [15];

These two approaches to providing inter-event content are now discussed in more detail.

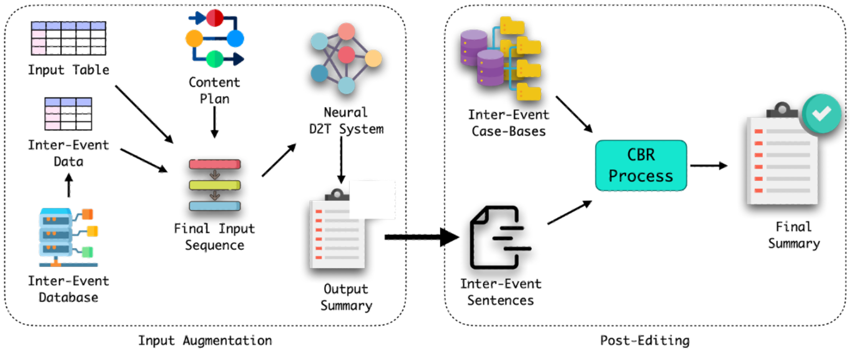

Fig. 4: Surface Realisation process with post-editing

5.1 Problem Representation Augmentation

In the first stage, we augment the problem representation from the current event with the data associated with the machine learning identified potential context-aware content for the selected themes from previously occurring events. This augmented problem representation is the input to the neural D2T system employed in our approach.

The current state-of-the-art in neural D2T uses a pipeline approach with separate planning and realisation phases [7]. The planning module outputs a sequence of paragraph plans, known as a macro plan, which is similar to the concepts described in Section 3 except they only contain the entity information and not the content type information. These paragraph plans also only contain the intra-event data for the entities. This macro plan is then fed to the surface realiser, which is a sequence-to-sequence model, to produce the final neural summary.

In our approach, we generate a similar macro-plan but which follows the content plan created by [13]. For each concept in the plan with intra-event type content, we keep the paragraph plan the same as before, i.e., only contain the current event data of the entity. However, for each inter-event concept in the plan, we append the inter-event features with their values in the paragraph sequence which were classified as 'yes' in the content selection process described in Section 4. Finally, this input sequence of paragraph plans is fed into the neural surface realiser to produce the textual summary.

The intuition here is that this process of augmenting the problem representation will provide the neural D2T system with the opportunity to generate context-aware content derived from previously occurring events. A pictorial representation of the process is shown in Figure 4.

5.2 Post-Editing

Even with the input augmentation, it is difficult to control the learning of neural systems and it is possible to still have inaccuracies in the generated summary.

Thus, to this end, we propose a method of post-editing the neural system's summary using a CBR-D2T method of dynamic templating proposed in [15].

In this post-editing method, the neural network summary is broken into sentences and then sentences identified in content plan as inter-event are replaced with a new sentence generated using the dynamic template CBR-D2T method. We build separate case-bases for player inter-event and team inter-event concepts. Cases in the case-base contain inter-event features on the problem-side and an associated inter-event content template as a solution. The process of building a case-base and generating a new sentence for a target problem is same as in [15]. The post-editing process is described pictorially in Figure 4. The idea here is that the output summaries should have similar distribution of content types as found in human written summaries.

6 Experiment Setup

We now define the experiment setup used to evaluate our proposed method.

6.1 Dataset

The SportSett dataset [12] of NBA matches is used to evaluate the proposed content selection and surface realisation algorithms [4]. Each match from the dataset contains a textual summary as the output and the associated match statistics, with the box- and line-scores, as the problem input. There is a temporal aspect involved here, as future matches should not be available to the learner. Hence the training set contains the earlier matches from the 2014, 2015 and 2016 seasons (total of 4775, some matches from the 2016 season have more than one summary) while the validation and test sets contain matches from the 2017 and 2018 seasons (1230 matches each) respectively.

The data for training the theme classifier for content selection, is build using the train and validation sets of SportSett. For each theme, if its inter-event features are included in a summary for an entity of the event, then its label is given as 1 otherwise 0. We use samples from the training set of SportSett for building the train set of the theme classifier while the validation set is used for building the test set for the classifiers.

6.2 Content Selection Models

We experiment with several binary classifiers for building the theme classifiers for content selection: Logisitic Regression (**LR**), k-Nearest Neighbours (**kNN**), Support Vector Machines (**SVM**), Multi-Layer Perceptron (**MLP**), and Random Forest (**RF**) [5].

[4] we use the GEM version of the dataset from https://huggingface.co/datasets/GEM/sportsett_basketball

[5] these models are trained with https://scikit-learn.org/stable/

Table 1: Dataset stats for building theme classifiers

	Player Average		Team Streak	
Label	Train Size	Test Size	Train Size	Test Size
Positive	3790	65	1707	488
Negative	73850	1470	5673	1972
Total	77640	1535	7380	2460

6.3 Surface Realisation Systems

For surface realisation, we select the current state-of-the-art macro-plan model (MP) [7] as the benchmark to compare our methods. We use the same macro-plan model for input augmentation and the post-editing methodologies. This is a pipeline-based neural network model with two components: a content planner, which combines planning and selection and is based on [16] that takes the event input data and generates a content plan (also referred to as a macro-plan); and a surface realiser, which is a sequence-to-sequence neural model with a Bi-LSTM encoder and an LSTM decoder that takes the macro-plan as input and generates the textual summary as output.

Thus, in our experiments we have three model's outputs to compare against each other:

– $\mathbf{MP_{base}}$: the base MP model of [7] with the authors input and training configuration. We also use the original content planning method proposed by their authors;
– $\mathbf{MP_{aug}}$: this model is the surface realiser from $\mathrm{MP_{base}}$ that takes the augmented input as described in Section 5. The augmented input is derived from taking the content plan from [13] and adding the inter-event content selected (using Section 4 method) to other intra-event content generated from the current event's data; and
– $\mathbf{MP_{pe}}$: this is the post-editing model which utilises the CBR-D2T method from [15] to post edit the output of the $\mathrm{MP_{aug}}$ model.

6.4 Evaluation Metrics

For content selection, basic classification metrics such as: Precision, Recall, and F1 score are used. Since the dataset is imbalanced, we report the marco-average of these metrics and use them for model selection.

For surface realisation, the following automated metrics are used:

– **Extractive Evaluation**: Inspired by information extraction evaluations from [17], we use a set of regular expressions to extract inter-event tuples from the system generations. These extracted tuples are then matched with the input data to evaluate the performance of text generation model;

Table 2: Performance of Theme Classifiers for inter-event content selection

	Player Average				Team Streak			
Model	Accuracy	F1	Precision	Recall	Accuracy	F1	Precision	Recall
LR	**95.77**	48.92	47.85	50	80.69	54.19	69.08	54.72
kNN	94.79	53.20	56.68	52.43	75.41	55.07	56.72	54.74
SVM	**95.77**	48.92	47.88	50	80.20	44.71	90.1	50.1
MLP	94.20	**60.49**	**61.74**	**59.48**	**80.85**	51.72	**72.61**	53.44
RF	94.85	57.09	61.44	55.41	79.07	**58.06**	63.42	**57.18**

– **Content Type Distribution and Concept Selection**: we also check the content type distribution of summaries generated from these models using the method proposed in [14]. The concept selection abilities of these models is also evaluated in accordance with the content selection process described in [13]. Here we check the precision, recall, f1, and DLD [1] scores of concepts selected in each summary against the human written gold summaries. A concept denotes the sentences structure identifying the type of entity and content described in the sentence.

We also used human evaluation to measure the accuracy of inter-event content in the system generated summaries. We utilise the human evaluation method used in previous research in the field [17,5,7,8]. In this evaluation, the human annotators are given some sentences from a summary along with the input data given to the system and asked to report the number of supporting and contradicting claims made in those sentences.

7 Results

The results are discussed in two parts: first, we briefly discuss the results of content selection, where we identify the best learning algorithm for building a theme classifier; and second, we discuss the results of surface realisation experiments, where we compare the effectiveness of the different methods proposed for adding inter-event content to the summaries.

7.1 Content Selection

The performance of theme classifiers built using different learning algorithms is shown in Table 2. We report the macro averaged scores of precision, recall and F1 metrics along-with accuracy of the classifiers. It can be observed that despite a higher accuracy, the other metrics have lower scores. This is expected as the dataset for these theme classifiers is imbalanced towards the negative class. Still we can see learners, such as MLP and RF, achieve around 60% for F1 scores. Since there will be another training with the neural network to generate the final summary by using the human written summaries, these results can be accepted we select MLP as the Player Average Theme classifier and RF as the Team Streak Theme classifier.

Table 3: RegEx evaluation results

	Player Average			Team Streak		
Systems	%Correct	#Supp.	#Contr.	%Correct	#Supp.	#Contr.
MP_{base}	20	4	16	0	0	0
MP_{aug}	42.65	29	39	0	0	1
MP_{pe}	63.41	1813	1046	38.04	35	57

7.2 Surface Realisation

Extractive Evaluations: We start with discussing the results from regular expression evaluation of the surface realisation outputs. The evaluation consists of a few regular expressions per theme that count the mention of inter-event content in the generated summaries. These expressions extract a tuple of information in the form of ($entity\_name$, $value$, $inter\_event\_feature\_name$) and match these with the input to count the number of supporting and contradicting claims. For example, for the given sentence - "*Kevin Durant is averaging 14 points over his last 5 outings*"; the extracted tuple would be - ($Kevin\_Durant$, 14, $AVG\_PTS\_LAST\_5\_GAMES$). This would then be matched to the input data to identify if this is supporting or contradicting claim. The results from this experiment for both themes is shown in Table 3. The column name '#Supp.' shows the number of supporting claims, '#Contr.' shows the number of contradicting claims, while '%Correct' is the percentage of correct/supporting claims out of total extracted claims.

The results clearly demonstrate the benefit of including inter-event content to the input data in order to include better inter-event content in the summaries. We see that MP_{base} only generates 20 inter-event examples for the player average theme with only 4 of those being correct. It also doesn't generate any inter-event content for team streak theme at all. Next, we see a good performance gain with MP_{aug} when the input of model is augmented with the selected inter-event content. This model generates 68 player average theme claims out of which 42% are correct. However, it still doesn't generate any supporting team streak theme claims. This suggests that it is difficult to make neural models generate a specific type of content if there aren't sufficient examples of it in the training set. Finally, we observe the MP_{pe} model's performance and immediately notice massive improvements across both themes. This model is able to generate around 2.9k player average theme claims, out of which 63% are also correct. For team streak theme as well, the model is generating 90+ claims with 38% of them being correct.

Content Type Distribution and Concept Selection: Next we investigate the content type distribution of summaries generated from these different systems. Figure 5 shows the percentage of sentences with different content types in summaries generated from the three systems and the human written gold sum-

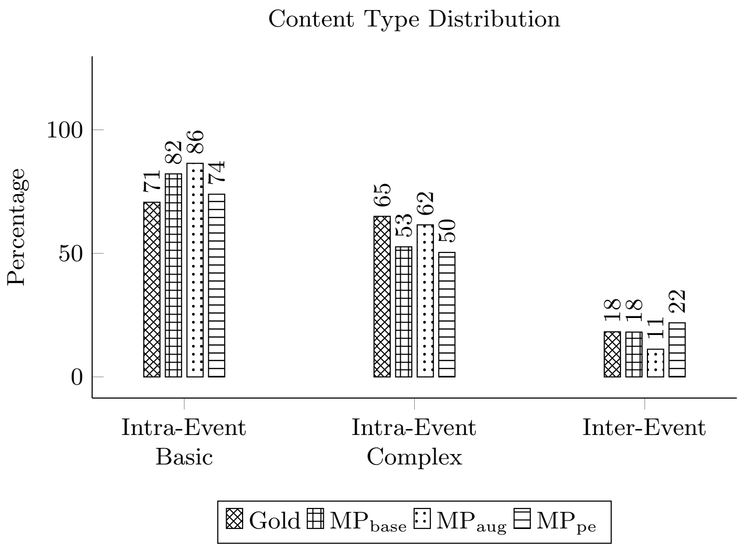

Fig. 5: Content-Type distribution of summaries from different systems

Table 4: Concept selection ability of different systems

		CS		CO	Length
CBR-Plan$_{euc}$	F1	Precision	Recall	DLD	Avg
Gold	-	-	-	-	12.76
MP$_{base}$	28.63	33.13	25.2	8.8	9.71
MP$_{aug}$	46.18	33.82	39.04	13.03	9.35
MP$_{pe}$	40.71	29.88	34.47	9.49	9.37

maries [6]. MP$_{pe}$ produces most amount of inter-event content, even higher than Gold. However, MP$_{base}$ is also able to generate equal amount of inter-event content as Gold but most of which is incorrect as identified in extractive evaluation results. We see that MP$_{aug}$ has the lowest amount of inter-event sentences despite having more content relating to inter-event themes as described in Table 3. This suggests that the MP$_{base}$ also generates some sentences with inter-event content that are not identified by regex evaluations. A quick look to the generated summaries will show that sentences such as: 'this was player X's first game after missing Y games due to injury'. These sentences, even though classified as inter-event, do not contain any information to be easily verified via automated metrics or even quick human evaluation.

We also investigate the content planning ability of the three systems using their generated summaries as evaluation method described in [13]. This method

[6] It is to note that a sentence can have multiple types of content, thus adding the percentage of different content types will not be equal to 100.

Table 5: Human evaluation results along with BLEU scores of different systems

Systems	**#Support** (↑)	**#Contra** (↓)	**BLEU** (↑)
$\mathbf{MP_{base}}$	0	3.75	17.6
$\mathbf{MP_{aug}}$	0.42	1.57	15.76
$\mathbf{MP_{pe}}$	1.22	0	15.08

extracts a concept list (as shown in Figure 2) from a system generated summary and then compares it with the concept list from the gold summary. In Table 4, we show the F1, Precision, and Recall scores to compare the concepts selected in system generations, while DLD (edit-distance) scores to compare the ordering of these concept lists. We can see that both the $\mathrm{MP_{aug}}$ and $\mathrm{MP_{pe}}$ systems are able to improve all four scores when compared to $\mathrm{MP_{base}}$. This suggests that adding inter-event content to the input data helps in improving the organisation of a generated summary that is more similar to the human written one.

7.3 Human Evaluation of Surface Realisation

Although automated evaluations are quick and easy to obtain, they may fail sometimes, particularly on new or edge-cases. Due to the richness of vocabulary of sports domain summaries, it is helpful to have some human judgement to support the automated evaluations. We randomly select 20 summaries generated from each system and then select at-most three sentences classified as across event.

We ask the annotators to count the number of supporting vs contradicting predictions, for which the results are shown in Table 5. We observe that MP_{pe} has the highest number of supporting facts, 1.22, with no contradicting facts in its generated summaries. Next we see that MP_{aug} has higher number of supporting facts as compared to MP_{base}, 0.42 against 0, while also having lower number of contradicting facts, 1.57 against 3.75, respectively. This can be expected as the MP_{pe} is using a CBR based dynamic template system to produce accurate texts. On the other hand, MP_{base} and MP_{aug} are relying on the generation process of neural systems which can be prone to hallucinations. The systems with contextual information also maintain similar fluency in their generations compared to their counterparts, as demonstrated in the BLEU scores.

These results prove that providing contextual information from past events to neural systems improve the quality of their generated summaries. The generations are much closer to the human written summaries in terms of content type and content plan, and are also more accurate without sacrificing fluency.

8 Conclusion

Current state-of-the-art D2T systems, despite achieving good performance, struggle to generate accurate context-aware content derived from past events. In this paper, we propose a CBR-assisted methodology for editing the summaries produced by neural D2T systems in order to produce summaries with both accurate inter-event content and content distributions similar to that found in human generated solutions.

A two-staged approach requires first content selection and then surface realisation. For content selection, machine learning is used to identify potential inter-event content whose associated data augments the current event's problem representation. For surface realisation, the output summary of a neural D2T system is edited with inter-event content identified using a CBR-D2T approach to produce the final event summary.

Extensive experimentation with both automated and human evaluation is performed on a sports domain dataset. Results demonstrate that our method is able to produce summaries that are more accurate than other neural systems. On average more than twice as many supporting facts and no contradicting errors in an inter-event sentence. The summaries generated from our system are also closer to human written summaries in terms of their content plan and content type distribution.

References

1. Brill, E., Moore, R.C.: An improved error model for noisy channel spelling correction. In: Proceedings of the 38th annual meeting of the association for computational linguistics. pp. 286–293 (2000)
2. Castro Ferreira, T., van der Lee, C., van Miltenburg, E., Krahmer, E.: Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. In: Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. on Natural Language Processing (EMNLP-IJCNLP). pp. 552–562 (2019)
3. Gatt, A., Krahmer, E.: Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. Journal of Artificial Intelligence Research 61, 65–170 (2018)
4. Gong, H., Feng, X., Qin, B., Liu, T.: Table-to-text generation with effective hierarchical encoder on three dimensions (row, column and time). 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). pp. 3143–3152. Association for Computational Linguistics, Hong Kong, China (2019), https://aclanthology.org/D19-1310
5. Puduppully, R., Dong, L., Lapata, M.: Data-to-text generation with content selection and planning. In: The Thirty-Third AAAI Conf. on Artificial Intelligence, AAAI 2019. pp. 6908–6915 (2019)
6. Puduppully, R., Dong, L., Lapata, M.: Data-to-text generation with entity modeling. In: Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 2023–2035 (2019)
7. Puduppully, R., Lapata, M.: Data-to-text generation with macro planning. Transactions of the Association for Computational Linguistics 9, 510–527 (2021)
8. Rebuffel, C., Soulier, L., Scoutheeten, G., Gallinari, P.: A hierarchical model for data-to-text generation. In: European Conf. on Information Retrieval. pp. 65–80. Springer (2020)

9. Reiter, E.: An architecture for data-to-text systems. In: Proc. of the 11th European Workshop on Natural Language Generation. p. 97–104 (2007)
10. Reiter, E., Dale, R.: Building Natural Language Generation Systems. Cambridge University Press (2000)
11. Robin, J., McKeown, K.: Empirically designing and evaluating a new revision-based model for summary generation. Artificial Intelligence 85(1), 135–179 (1996), https://www.sciencedirect.com/science/article/pii/0004370295001255
12. Thomson, C., Reiter, E., Sripada, S.: SportSett:basketball - a robust and maintainable data-set for natural language generation. In: Proc. of the Workshop on Intelligent Information Processing and Natural Language Generation (2020)
13. Upadhyay, A., Massie, S.: A case-based approach for content planning in data-to-text generation. In: Case-Based Reasoning Research and Development: 30th International Conference, ICCBR 2022, Nancy, France, September 12–15, 2022, Proceedings. pp. 380–394. Springer (2022)
14. Upadhyay, A., Massie, S.: Content type profiling of data-to-text generation datasets. In: Proceedings of the 29th International Conference on Computational Linguistics. pp. 5770–5782. International Committee on Computational Linguistics, Gyeongju, Republic of Korea (Oct 2022), https://aclanthology.org/2022.coling-1.507
15. Upadhyay, A., Massie, S., Singh, R.K., Gupta, G., Ojha, M.: A case-based approach to data-to-text generation. In: Int. Conf. on Case-Based Reasoning. pp. 232–247. Springer (2021)
16. Vinyals, O., Fortunato, M., Jaitly, N.: Pointer networks. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 28 (2015)
17. Wiseman, S., Shieber, S., Rush, A.: Challenges in data-to-document generation. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 2253–2263 (2017)