UPADHYAY, A. and MASSIE, S. 2022. A case-based approach for content planning in data-to-text generation. In *Keane, M.T. and Wiratunga, N. (eds.) Case-based reasoning research and development: proceedings of the 30th International conference on case-based reasoning (ICCBR 2022), 12-15 September 2022, Nancy, France*. Lecture notes in computer science, 13405. Cham: Springer [online], pages 380-394. Available from: https://doi.org/10.1007/978-3-031-14923-8_25

A case-based approach for content planning in data-to-text generation.

UPADHYAY, A. and MASSIE, S.

2022

A Case-Based Approach for Content Planning in Data-to-Text Generation

Ashish Upadhyay and Stewart Massie

Robert Gordon University, Aberdeen, UK
{a.upadhyay,s.massie}@rgu.ac.uk

Abstract. The problem of Data-to-Text Generation (D2T) is usually solved using a modular approach by breaking the generation process into some variant of planning and realisation phases. Traditional methods have been very good at producing high quality texts but are difficult to build for complex domains and also lack diversity. On the other hand, current neural systems offer scalability and diversity but at the expense of being inaccurate. Case-Based approaches try to mitigate the accuracy and diversity trade-off by providing better accuracy than neural systems and better diversity than traditional systems. However, they still fare poorly against neural systems when measured on the dimensions of content selection and diversity. In this work, a Case-Based approach for content-planning in D2T, called CBR-Plan, is proposed which selects and organises the key components required for producing a summary, based on similar previous examples. Extensive experiments are performed to demonstrate the effectiveness of the proposed method against a variety of benchmark and baseline systems, ranging from template-based, to case-based and neural systems. The experimental results indicate that CBR-Plan is able to select more relevant and diverse content than other systems.

Keywords: Data-to-Text Generation, Case-Based Planning, Content Planning

1 Introduction

Data-to-Text Generation (D2T) is the process of summarising insights and information extracted from non-linguistic structured data in a textual format [15,3]. With business processes often generating a huge amount of domain-specific data, which is not easily understandable by humans, there is a growing need to synthesise this data by converting it into textual summaries that are more accessible. There are many real-world applications, from weather or financial reporting [7,5,16] to medical support and sports journalism [9,23]. There are two main problems that should be addressed in a D2T problem: **content planning**, selecting and ordering important content from the input data (implicit or explicit), as in *what to say?*; and **surface realisation**, conveying the selected content in a textual summary, as in *how to say?*. Content planning is the focus of this work.

There are different types of systems that can be used to solve the D2T problem. Traditional methods, that use a modular approach with hand-crafted rules and templates acquired from domain knowledge, are very good at producing high-quality textual summaries with accurate information [15,16,7]. However, they lack diversity and generate monotonous texts. Also, in complex domains, it is difficult to hand-craft the rules for every possible situation, making these systems difficult to scale. Current state-of-the-art neural systems usually take an end-to-end approach and are capable of producing diverse and fluent summaries while offering better scalability across domains [23,13]. However, they are prone to errors and often generate inaccurate texts not supported by the input. There have been some attempts to utilise neural systems in a modular way by breaking them into separate planning and realisation phases [12,2]. Whilst offering better performance than the end-to-end counterparts they tend to be more conservative and achieve better accuracy at the cost of diversity.

Case-Based approaches, also by taking a somewhat modular approach, try to mitigate the accuracy and diversity trade-off by retrieving similar problems from the case-base and reusing them to dynamically generate a custom template for the new problem [22,1]. They offer better accuracy than neural systems while better diversity than traditional systems. Nonetheless, the case-based approaches still perform below par with neural systems when evaluated for content selection and diversity in generations. Thus, missing out on much relevant information that may be important for the summary and would have been selected by a human author.

In this work, we propose a Case-Based Reasoning (CBR) approach for content-planning in D2T problems that selects the main components for a summary and organises them to create a plan by reusing solutions of previous similar problems. In this process, first, several key components are identified that contribute to writing a D2T summary, and then a CBR method is used to create a content-plan by selecting and organising a subset of those components. The main contributions of this work are as follows:

1. develop a new CBR-based model for the content-planning task in D2T [1];
2. introduction of a new concept identification process to support evaluation of D2T approaches; and
3. demonstrating the performance of the proposed method at content selection effectiveness and diversity on a standard D2T evaluation data set.

The rest of the paper is organised as follows: in the next section we discuss relevant related works from the literature; then in Section 3, we provide an in-depth background of the problem and provide insight on where this work fits into a bigger picture; which is followed with the description of our proposed method in Section 4; and the experimental setup in Section 5. We then discuss the results obtained from the experiments in Section 6; and finally conclude the paper with some key takeaways and future directions in Section 7.

[1] code-base is available at https://github.com/ashishu007/data2text-cbr-plan

2 Related Works

Data-to-Text Generation (D2T) is a sub-field of Natural Language Generation (NLG) aiming to summarise structured non-linguistic data as opposed to Text-to-Text Generation (T2T) which aims to summarise linguistic data in textual summaries [3]. D2T has been studied for decades, one of the very first systems proposed in the 1980s, generated textual summaries of financial data [7]. There have been several other traditional systems in multiple domains ranging from weather forecasting to medical support documentation [16,5,9]. They have followed a modular approach by breaking the problem into several smaller ones and solving them separately with different modules designed with carefully crafted rules [14]. Recent advancements in neural techniques have given rise to learning based neural systems that initially tried to model the whole task into a single end-to-end process [23,13]. But recent trends have seen the resurgence of modular approaches even in neural systems demonstrating better performance than their end-to-end counterparts [2,10,12]. However, the planning-based neural systems tend to be more conservative by generating easier and less diverse summaries in order to become more accurate.

Traditional D2T systems are capable of producing high-quality texts but come with the challenge of scalability across domains and lesser diversity in the text generated. On the other hand, neural systems offer better scalability and diversity than traditional systems, but at the expense of accurate generations. Case-Based systems also take a modular approach and aim to mitigate this accuracy and diversity trade-off by generating a custom template for a new problem using solutions from similar past experiences [1,21,22]. Although the idea of CBR systems being more accurate than neural and more diverse than traditional counterparts appears sound, typical performance is poorer than neural systems in terms of content selection and diversity.

Case-Based Planning has also been studied for a long time with initial methods being applied in several domains ranging from holiday and logistics to story planning [20,17,4]. In this work, the focus is to build a CBR-based content-planning module for D2T problems that generates better content-plans with respect to content selection and diversity.

3 Problem Description and Representation

Each case in a D2T case-base is an event which consists of the event's data on the problem side and its textual summary on the solution side. The textual summary of each event aims to describe the important insights and information extracted from the event's data. In easier domains, the event summary contains information from only the single event but in more complex domains, the summary may contain information from its neighbouring cases as well [22].

The problem side of each case in the case-base is represented by multiple entities. Each of these entities is further represented with several features that aim to describe the entities. Each feature is assigned a value, which in most cases,

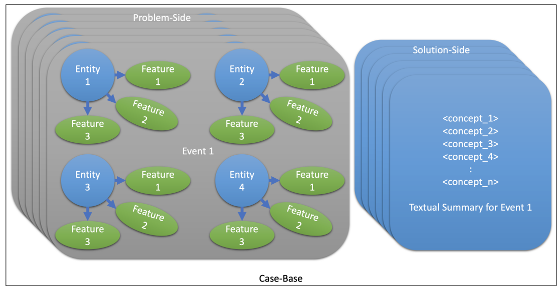


Fig. 1: Case-Base in a Data-to-Text Generation problem

is numerical but may also be either categorical or textual. Solutions-side of the event summaries is a combination of multiple concepts each describing single or multiple entities with some type of information. Each concept is essentially a sentence from the textual summary. An example case-base with its problem and solution sides is shown in Figure 1.

Taking the sports domain datasets [23,19] as an example, an event can be seen as a match between two teams for which a summary is written. Each event is represented by its entities which are the players and teams that play in the game. Furthermore, each entity is represented with several features which are the stat categories that are recorded for each player or team. The textual summary is a set of concepts each describing an entity (or combination of entities) from the event.

To build a D2T system, the following points should be taken in account:

– first, identify all the relevant concepts that may be interesting for any event and can be included in the summaries;
– second, select important concepts with respect to a given event that is interesting and should be included in the event's summary;
– third, decide which entities (or a combination of entities) should be described using each selected concept; and
– finally, generates a semantically accurate sentence for each concept describing an entity and pragmatically orders them to generate the final summary.

In this work, we will focus on the first three steps of this process. We will use information gathered from some corpus analysis to identify several possible concepts. Then we will use a CBR method to select the relevant concepts for a target case, and finally, another data-driven method will be used to identify the entities that will be described by each selected concept. This is analogous to the **Content Planning** phase of a standard D2T system [2,12].

4 Content Planning Methodology

In this work, we focus on building a method for generating a content-plan that will be used by a text-generator model to produce an event's summary. As dis-

S01	The Philadelphia 76ers (16-52) defeated the Detroit Pistons (24-44) 94-83 on Wednesday in Philadelphia .
S02	The 76ers were able to pull off the win despite Nerlens Noel leaving the game with a right foot contusion after playing just 22 minutes .
S03	He had 11 points on 5-of-10 shooting , four rebounds and three blocks in that time and did not return .
S04	It was Ish Smith who led the way for Philadelphia , as he was moved into the starting point guard role while Isaiah Canaan was moved to the bench .
S05	Smith thrived in the role , recording 15 points on 6-of-12 shooting , eight assists and three steals in 26 minutes .
S06	Canaan struggled coming off the bench , putting up just nine points on 2-of-10 shooting and four assists in 22 minutes .
S07	Jason Richardson shot his way out of a slump , scoring 15 points on 4-of-7 shooting in 25 minutes , as it was just the first time in five games that Richardson scored in double figures .
S08	Conversely , Robert Covington struggled , as he shot just 1-of-6 from the field on his way to three points in 22 minutes off the bench .
S09	It was a quick fall back to reality for the Pistons , as just a day after upsetting the Grizzlies and ending a 10 - game losing streak , they lost to one of the NBA 's worst teams .
S10	They were without Greg Monroe , who sat out his second consecutive game with a strained knee .
S11	Despite the loss , Detroit did have some strong performances .
S12	Reggie Jackson followed up Tuesday night 's big game with a triple-double , putting up 11 points on just 4-of-17 shooting , 11 rebounds and 10 assists in 32 minutes .
S13	Kentavious Caldwell-Pope also followed up his 24-point performance on Tuesday in a strong way , scoring 20 points on 7-of-16 shooting and grabbing eight rebounds in 36 minutes .
S14	Up next , the 76ers will take on the Knicks at home Friday , while the Pistons head home Saturday to take on the Bulls .

Fig. 2: An example summary from SportSett dataset

cussed earlier, a summary is a set of concepts each describing an entity (or a combination of entities) from the event. Thus a content-plan is a set of concepts, ordered in a sequence, each aligned with one or more entities. To build the content-plan, the first step is to identify all the possible concepts that may be relevant to any event's summary. Then, for a given target event (new problem), a subset of identified concepts and associated entities corresponding to each concept should be selected and ordered. The final stage would be to select a suitable template for each selected concept and populate it with the corresponding entity's values. But this final step of generating the text by selecting and populating templates is out of the scope of this work.

4.1 Concept Identification

The summaries in a typical D2T domain contain information of different complexities. A summary is written with multiple sentences, where each sentence can have several information elements of different types. For example, in Figure 2, S12 identifies that player 'Reggie Jackson' scored a triple-double [2] with 11 points, rebounds and 10 assists in the game, continuing his good performance from the previous game. Firstly, the straightforward information, such as he scored 11 points and rebounds, can be directly copied from the input data (problem side representation) into the output summary. Secondly, the information that the player scored a triple-double is not explicitly stated in the input data, rather it needs to be derived from several features from the single event, which in this case would be from the player's stats, such as points, rebounds, etc. This kind of information will be more difficult for a system to generate, as it requires the system to be capable of performing inference and arithmetic operations. Lastly, the fact that it was the player's continuation from the previous game can only

[2] https://en.wikipedia.org/wiki/Double-double#Triple-double

be derived by taking the data from several events into account. This will be even harder as the amount of data that needs to be considered for inference will grow massively. Thus, an event's summary can have three types of information:

- **Intra-Event Basic (Basic; B)** information that can be directly copied from an event's input data into the output summary;
- **Intra-Event Complex (Within; W)** information that needs to be implicitly derived from given information in an event's data; and
- **Inter-Event (Across; A)** the information that can be only derived from taking the data of multiple events into account.

Sentences in a summary can be classified based on the type of information it contains into one of the seven categories: just Basic, **B**; just Within, **W**; just Across, **A**; both Basic and Within, **B&W**; both Basic and Across, **B&A**; both Within and Across **W&A**; and finally, all Basic, Within and Across, **B&W&A**. In addition to different types of information in the summary, each sentence can describe different types of entities: a **Player (P)**; and a **Team (T)**. Taking another example from Figure 2, S01 describes two teams' information whereas sentence S02 describes a team's and a player's information, while S03 describes a player's information. Thus, a sentence, based on the entity it describes, can be classified into the following five categories: just one Player, **P**; just one Team, **T**; more than one Players, **P&P**; more than one Teams, **T&T**; and finally, both Players and Teams, **P&T**.

An event summary from the SportSett dataset, based on the information and entities a sentence describes, can be classified into a total of 35 categories (7 types of information times 5 types of entities). We term each of these 35 categories as different concepts that can be used to write a summary of an event. We show the proportion of these concepts in our case-base in Figure 3. On $x-axis$, we see all possible concepts, and on $y-axis$, we show the number of sentences categorised as that concept. These statistics are calculated using an automated system that extracts the entities mentioned in a sentence and classifies the sentence into its information-type category. This system consists of two modules: first, an entity extraction module, the same as the method used in building train data for IE models in [23]; and second, an information-type classifier, which is a Roberta model [8] fine-tuned with a multi-label classifier head trained on 600 samples and tested on 250 manually annotated samples. The classifier achieves 91% of the Macro-F1 score.

It is noted that although the examples used here for demonstration are specific to a dataset, the same approach can be applied to other datasets or domains with similar settings such as MLB D2T [11].

4.2 Concept Selection

After we have identified all possible concepts, the next step in writing a D2T summary is to select a subset of concepts that may be important and interesting for the target event's summary. Since the event summaries in D2T domains

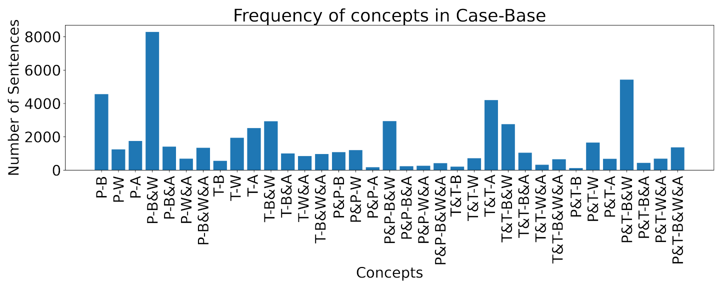


Fig. 3: Frequency of concepts in the case-base

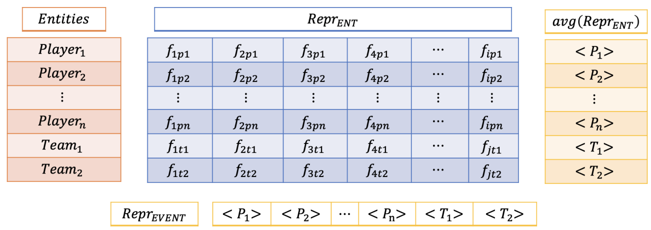
follow the principle of '*similar problems have similar solutions*', we can employ a standard CBR approach to select a subset of concepts by retrieving similar events from previous examples and then reusing their solution to propose the solution for the new problem. To build a CBR model for this concept selection stage, we first need to build our case-base which will consist of events with their entity-based representation on the problem-side and their list of concepts used in the summary on the solution-side.

The problem-side representation of an event can be built by combining the representation of the different entities an event contains. Each entity in an event is represented with several features, all of which are assigned a value, effectively representing an entity with a vector of length the same as the number of features. An event can have multiple entities, making the initial representation of an event two-dimensional. In this work, the entity representation is simplified by taking its arithmetic mean to build a one-dimensional representation of an event. The solution-side's concept list of an event can be extracted using the same technique used for calculating the proportion of concepts described in the previous section. An example problem-side representation and solution-side concept list of an event from the SportSett dataset is shown in Figure 4.

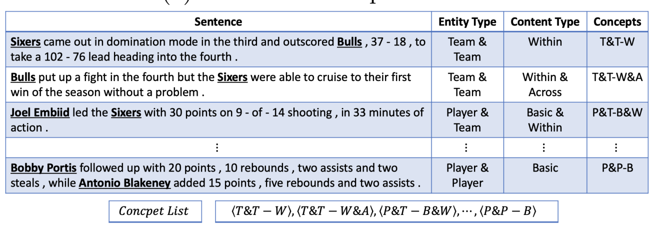
With the case-base developed, when a new problem arrives, its problem representation is built using its entities and then the most similar problem is retrieved from the case-base using Euclidean distance. The retrieved case is reused as the solution to the new problem. It is noted that a more sophisticated approach for retrieval can be developed by exploring alternative similarity measures and considering the top-k most similar cases when proposing the new solution. But these are left to the future work and here the focus is on utilising a simple approach for building the content-planning module of a data-to-text generation pipeline.

4.3 Entity Selection

The next step in content selection and planning is to select the entities (or combination of entities) that should be described in each of the selected concepts in the previous stage. This is achieved by ranking all the different types of



(a) Problem-side representation



(b) Solution-side concept-list

Fig. 4: (a) Problem-side and (b) Solution-side of an event

entities in a stack where the highest-ranked entity will be described using the first concept of its type, the second-highest ranked entity will be described with the second concept of its type, and so on. Thus, an algorithm is needed to rank the entities of an event. This can be achieved by learning the feature weights of the entity's representation, which can be used to score the entities and rank them based on the scores. To formalise:

$$Repr_{ENT} = [f_1, f_2, f_3, \cdots, f_n]$$
$$W = [w_1, w_2, w_3, \cdots, w_n], \forall w \in (-1, 1)$$
$$Score_{ENT} = \sum_{i=1}^{n} f_i \cdot w_i$$

The feature weights W are calculated using a PSO algorithm [6] optimised on a classification dataset for both the entity types (players and teams). For team entities, the classification data is prepared by subtracting the losing team's representation from winning team's representation and assigning it the label 1 (or win), and vice-versa for label 0 (or lost).

$$(Rep_{clf})^i = [(f_{1W} - f_{1L}), (f_{2W} - f_{2L}), \cdots, (f_{nW} - f_{nL})]$$
$$(Rep_{clf})^j = [(f_{1L} - f_{1W}), (f_{2L} - f_{2W}), \cdots, (f_{nL} - f_{nW})]$$
$$Lab^i = 1 \& Lab^j = 0$$

Similarly, win-loss data can be created for player entities as well where a player mentioned in the respective event summary will be considered a winner compared to a player from the event not mentioned in the summary.

5 Experimental Setup

This experiment aims to evaluate our new CBR planning-based algorithm, which we call CBR-Plan. At this stage, we are not interested in the text itself but rather the plan for the text solution. Hence, we measure the effectiveness of CBR-Plan by measuring its ability to select the same concepts and associated entities as chosen by a journalist who has already written solutions for the problems. A basketball dataset forms the case base and CBR-Plan is compared to both benchmark and state-of-the-art systems.

5.1 Dataset

The SportSett dataset [19] of NBA matches is used to generate an evaluation case base [3] in which a match becomes a case. Each match 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 dev and test sets contain matches from the 2017 and 2018 seasons (1230 matches each) respectively.

The training set is used to create the case-base following the method described in Section 4.2. There are total 4775 cases in the case-base for concept selection. For the entity ranking method, we again use the instances from the train set for preparing the PSO train data. We collect 66,738 instances for the players' feature weighting task while 7,380 instances are available for the teams' feature weighting task.

5.2 Benchmarks and Baselines

CBR-Plan is compared with four existing models, as follows:

- **Template-Based (Template)**: the baseline model proposed in [23] which contains a few handcrafted templates to verbalise the data of a few entities from an event. In this work, an updated version of this model [22] is used which adds a few more templates for generating extra information (next-game information of a team).

[3] We have used the GEM version of the dataset from https://huggingface.co/datasets/GEM/sportsett_basketball

– **Case-Based Model (CBR)**: a case-based approach to Data-to-Text generation [22] which breaks down the summaries into several components. Then a case-base for each component is built which consists of the entity's feature values as problem representation while templates verbalising that entity's features as the solution. To generate an event summary, a standard case-based approach is used to retrieve the best template for different entities in each component.
– **Entity Model (Ent)**: an entity-focused approach [11] uses a sequence-to-sequence model consisting of an MLP encoder and LSTM decoder with copy mechanism. An added module updates the input record's representation during the generation process. At each decoding step, a GRU is used to decide the record that needs to be updated and then update its value.
– **Macro-Plan Model (MP)**: a neural pipeline model for data-to-text generation proposed in [12]. It consists of two separate modules: first, a micro-planning module, which takes all the entities as input and selects and orders the important entities (or combination of entities) using a Pointer network to build a micro-plan. The second module is a text generator which makes use of a standard LSTM based sequence-to-sequence model with a copy mechanism to generate a summary from the micro-plan.

Both the neural models are trained using the same hyper-parameters as described by the authors in their works.

5.3 Evaluation Metrics

Performance is measured on two important dimensions of data-to-text generation: content selection and diversity. Content selection is evaluated by comparing each method's output with the human reference summaries. Two measures are calculated, first for the proposed concepts and then for the entities selected.

Each method outputs a list of concepts for each new problem in the test set using the approach described in Section 4.2. They also produce a list of entities with the same length as the concept list, where the planned text to be generated for each concept would include the entity of the same index from the entity list, using the approach described in Section 4.3. The concept and entity list is extracted from human reference and generated summaries by first splitting the summaries into sentences and then extracting the entities from each sentence using the approach employed for calculating extractive evaluation metrics in [23]. This gives an entity list for the current problem. Finally, the sentence can be classified into its content-type, which when combined with entity-type in the sentence will give the concept list for the case's text summary. For each method, concept and entity lists are compared with the human reference (**Gold**) lists by calculating **F2**, **precision** and **recall** scores to evaluate the **content selection** capability of the systems. Since the system generations are expected to be of similar length as gold summaries, a system achieving higher precision with smaller generations is not good. So F2 becomes a better measure to evaluate these systems which give more weight to recall than precision. For **diversity**, we

Table 1: Content Selection scores and average concept length

	Concepts			**Entities**			**#Concept**
System	**F2**	**Prec**	**Rec**	**F2**	**Prec**	**Rec**	**Avg**
Gold	-	-	-	-	-	-	12.76
Template	25.52	37.81	23.6	48.24	89.39	43.26	7.97
CBR	32.15	47.3	29.76	60.09	90.91	55.39	8.03
Ent	25.38	24.05	25.73	54.05	60.27	52.7	13.66
MP	26.47	33.13	25.2	49.9	78.98	45.7	9.71
CBR – Plan	36.75	33.32	37.72	61.97	61.12	62.19	14.45

measure the **proportion of different concepts** used by each method compared to the gold summaries.

6 Results and Discussion

The results are discussed in two parts: first, for content selection; and second, for diversity compared to the human reference summaries.

6.1 Content Selection

F2, precision and recall scores are shown in Table 1 for both entities and concepts selected by the five evaluated models. With concept selection, Template achieves the second-worst F2, second-highest precision, and worst recall score. CBR has the highest precision while second-best F2 and recall. For the two neural systems, Ent has the worst F2 and precision with second-worst recall; while MP has third-best F2, second-worst precision and recall. Our proposed algorithm, **CBR-Plan**, achieves the highest F2 and recall score with third best precision score. Similar patterns can be seen in the entity selection scores as well.

In both cases, (**CBR-Plan**) achieves the highest F2 and recall scores. This suggests that the proposed method can select more relevant concepts and entities, similar to human reference summaries when compared to the other systems. However, it is also selecting most concepts and entities that are not present in Gold summaries (note these selections may still be relevant). Looking at the average number of concepts selected (see the last col. of Table 1) gives a deeper insight into the precision and recall scores. We can observe that most systems that have higher precision scores are generating smaller summaries. By doing so, the systems can achieve higher precision but miss many relevant concepts that are interesting and should be included in the summary. Similar behaviours have been observed with neural models in other relevant works as well [18]. By generating a solution based on the number of concepts present in similar human-generated solutions, CBR-Plan is producing a more realistic solution aligned with the human summaries than most of the comparative algorithms.

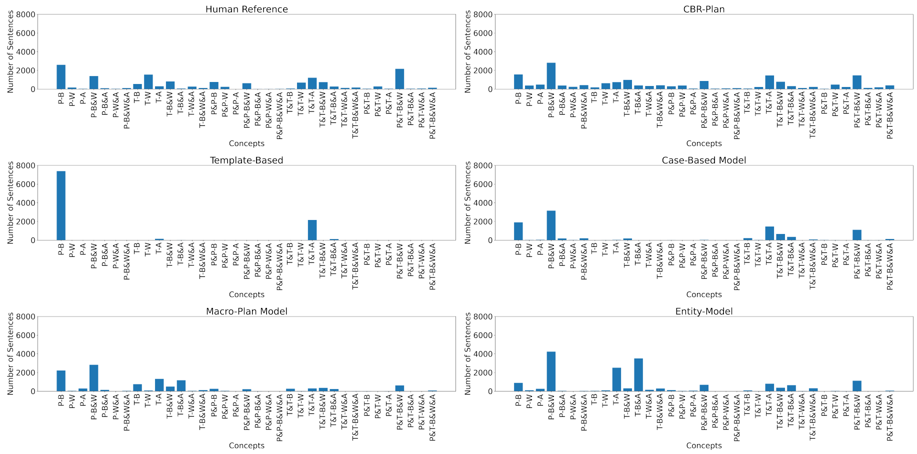


Fig. 5: Proportion of different concepts in different systems

Table 2: Correlation between concept frequency of different systems versus gold

System	**CBR-Plan**	**Template**	**CBR**	**MP**	**Ent**
Correlation	0.7571	0.6288	0.6973	0.5954	0.3105

6.2 Diversity

In this section, we investigate the ability of different systems to select different concepts in their summary generations. In Figure 5, the frequency of different concepts being selected is shown for the human reference summaries and each evaluated model. The human reference summary has a relatively even distribution over all concepts. Intuitively, we would expect a well-automated system to generate summaries with a similar distribution over the concepts as the human-generated solution.

If we look at the distribution of the different systems, we can observe that Template and CBR are only selecting a few popular concepts and completely ignoring the others. With the neural systems, while MP does select across many concepts there is still a popularity bias by heavily selecting the most popular concepts. For example, it is selecting mostly P-B, P-B&W, T-B, and T-B&W, which seem to be the most popular ones, but is missing many other important concepts. Similarly, Ent has a popularity bias in selecting high numbers of a few popular concepts.

CBR-Plan can select a broad range of concepts that is most similar to the distribution seen with human reference summaries. Along with selecting the most popular ones, CBR-Plan is also able to select non-popular but important concepts. We show Pearson's correlation coefficient of concept distribution in system generation versus human reference summaries in Table 2. CBR-Plan has the highest correlation with human reference summaries followed by CBR, MP,

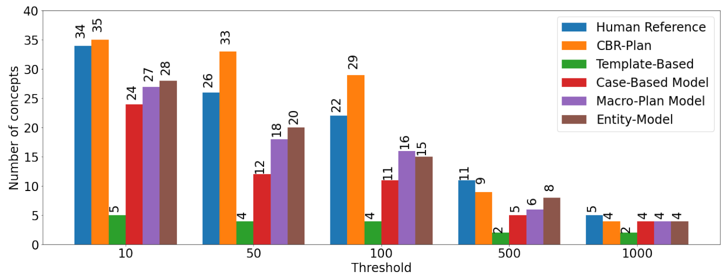


Fig. 6: Concept frequency of different systems after a threshold

Template and Ent. It is also surprising to see the Ent model performing so poorly on this measure, which appears to be due to the model selecting higher numbers of some rare concepts, e.g. T-B&A, and T-A.

Figure 6 shows the number of total concepts that have been selected by each model above a threshold figure. With the increase in the threshold, the number of concepts will decrease. It can be seen that the steep drop in CBR, MP and Ent is higher than for the human reference which is indicative of popularity bias, whereas CBR-Plan is more similar to the human reference and sometimes selects even higher numbers.

6.3 Qualitative Analysis

In Figure 7, we show an example of concepts and entities selected in the human reference summary as well as in the different systems' generation. For example, the human summary starts with a 'T-W' concept associated with entity Chicago Bulls. That means, the first sentence talks about Chicago Bulls with Intra-Complex (Within) type information. We can observe that Template is mostly selecting concepts that includes a player with their Intra-Basic type information. CBR is selecting concepts of different types, some with combinations of different entities as well, but is still smaller in size. Both neural models are selecting different types of concepts but are either smaller (MP) or selecting easier (Ent, which rarely selects Inter-Event content). In contrast CBR-Plan is able to select different concept types of reasonable length with different content-types as well.

7 Conclusion and Future Works

In this work, a Case-Based planning approach is introduced for content planning in D2T problems. The proposed method first identifies important components for an event's summary known as concepts and then uses a CBR approach to select a subset of those concepts important and relevant for the event. In the final step, a ranking method is used to rank the entities of an event and align

```json
{
  "Gold": [
    "Chicago Bulls|T-W", "Philadelphia 76ers & Chicago Bulls|T&T-W", "Philadelphia 76ers & Chicago Bulls|T&T-W",
    "Philadelphia 76ers & Chicago Bulls|T&T-W&A", "Joel Embiid & Philadelphia 76ers|P&T-B&W", "Ben Simmons|P-B&W", "Robert Covington|P-B",
    "Philadelphia 76ers|T-B", "Zach LaVine & Chicago Bulls|P&T-B&W","Bobby Portis & Antonio Blakeney|P&P-B", "Chicago Bulls|T-B", "Chicago Bulls|T-A",
    "Philadelphia 76ers|T-A"
  ],
  "CBR-Plan": [
    "Philadelphia 76ers & Chicago Bulls|T&T-B&W", "Ben Simmons & Philadelphia 76ers|P&T-B&W&A", "Philadelphia 76ers|T-W", "Chicago Bulls|T-W",
    "Philadelphia 76ers|T-B&W", "Ben Simmons & Joel Embiid|P&P-B&W", "Dario \u0160ari\u0107 & Philadelphia 76ers|P&T-B&A", "Philadelphia 76ers|T-W",
    "Philadelphia 76ers & Chicago Bulls|T&T-B&W&A", "Joel Embiid & Philadelphia 76ers|P&T-B&W", "Ben Simmons & Landry Shamet|P&P-B",
    "Joel Embiid|P-B&W&A"
  ],
  "Template": [
    "Joel Embiid|P-B", "Zach LaVine|P-B", "Robert Covington|P-B", "Bobby Portis|P-B", "Antonio Blakeney|P-B", "Jabari Parker|P-B",
    "Philadelphia 76ers|T-A", "Philadelphia 76ers|T-A"
  ],
  "CBR": [
    "Philadelphia 76ers & Chicago Bulls|T&T-B&W", "Ben Simmons|P-B&W", "Joel Embiid & Philadelphia 76ers|P&T-B&W",
    "Robert Covington & Philadelphia 76ers|P&T-B&W", "Bobby Portis|P-B&W", "Zach LaVine|P-B&W", "Philadelphia 76ers & Chicago Bulls|T&T-A"
  ],
  "Macro-Plan": [
    "Philadelphia 76ers & Chicago Bulls|T&T-W", "Philadelphia 76ers & Chicago Bulls|T&T-B", "Ben Simmons|P-B&W", "Philadelphia 76ers|T-A",
    "Philadelphia 76ers & Chicago Bulls|T&T-W&A", "Zach LaVine|P-B&W", "Jabari Parker|P-B", "Bobby Portis|P-B", "Jabari Parker|P-B",
    "Chicago Bulls|T-A", "Philadelphia 76ers|T-A"
  ],
  "Entity-Model": [
    "Philadelphia 76ers|T-B&A", "Philadelphia 76ers & Chicago Bulls|T&T-B&W", "Joel Embiid & Philadelphia 76ers|P&T-B&W", "Zach LaVine|P-B&W",
    "Dario \u0160ari\u0107|P-B&W", "Amir Johnson & Bobby Portis|P&P-B&W", "Ben Simmons & T.J. McConnell|P&P-B", "Justin Holiday|P-B&W",
    "Joel Embiid & Justin Holiday|P&P-B&W", "Zach LaVine|P-B&W", "Dario \u0160ari\u0107|P-B&W", "Ben Simmons & T.J. McConnell|P&P-B&W",
    "Antonio Blakeney|P-A"
  ]
}
```


Fig. 7: Concepts and Entities selected in different systems

them to the concepts selected in previous step. Extensive experimentation is conducted to demonstrate the effectiveness of proposed methodology by comparing it against several benchmark and baseline systems of different types, ranging from template-based to case-based and neural approaches. Experiments demonstrate that the proposed method is able to achieve best recall in terms of selecting relevant content and provides most diversity by selecting different concepts more aligned with human reference summaries than the other systems.

In future, the aim is to utilise the selected concepts from this work in surface realisation and generating the final event summary. The next process will be inspired by previous CBR D2T systems [22] where the most suitable templates will be extracted for transforming the concepts with their respective entities into text. We also plan to enrich the entities' representation by adding across-event information in order to improve the retrieval process. A richer representation will help in the next iteration of surface realisation, ultimately improving the quality of the generated summaries.

References

1. Adeyanju, I., Wiratunga, N., Lothian, R.: Learning to author text with textual cbr. In: European Conf. on Artificial Intelligence (2010)
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. Gervás, P., Díaz-Agudo, B., Peinado, F., Hervás, R.: Story plot generation based on cbr. Knowledge Based Systems 18, 235–242 (2005)
5. Goldberg, E., Driedger, N., Kittredge, R.: Using natural-language processing to produce weather forecasts. IEEE Expert 9(2), 45–53 (1994)
6. Kennedy, J.: Particle Swarm Optimization, pp. 760–766. Springer US (2010)
7. Kukich, K.: Design of a knowledge-based report generator. In: 21st Annual Meeting of the Association for Computational Linguistics. pp. 145–150. Association for Computational Linguistics (1983)
8. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
9. Portet, F., Reiter, E., Gatt, A., Hunter, J., Sripada, S., Freer, Y., Sykes, C.: Automatic generation of textual summaries from neonatal intensive care data. Artificial Intelligence 173(7-8), 789–816 (2009)
10. 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)
11. 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)
12. Puduppully, R., Lapata, M.: Data-to-text generation with macro planning. Transactions of the Association for Computational Linguistics 9, 510–527 (2021)
13. 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)
14. Reiter, E.: An architecture for data-to-text systems. In: Proc. of the 11th European Workshop on Natural Language Generation. p. 97–104 (2007)
15. Reiter, E., Dale, R.: Building Natural Language Generation Systems. Cambridge University Press (2000)
16. Sripada, S., Reiter, E., Davy, I.: Sumtime-mousam: Configurable marine weather forecast generator. Expert Update 6(3), 4–10 (2003)
17. Stewart, S.I., Vogt, C.A.: A case-based approach to understanding vacation planning. Leisure Sciences 21(2), 79–95 (1999)
18. Thomson, C., Reiter, E.: Generation challenges: Results of the accuracy evaluation shared task. In: Proc. of the 14th Int. Conf. on Natural Language Generation. pp. 240–248. Aberdeen, Scotland, UK (2021)
19. 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)
20. Thomson, R., Massie, S., Craw, S., Ahriz, H., Mills, I.: Plan recommendation for well engineering. In: Int Conf. on Industrial, Engineering and Other Applications of Applied Intelligent Systems. pp. 436–445 (2011)
21. Upadhyay, A., Massie, S., Clogher, S.: Case-based approach to automated natural language generation for obituaries. In: Int. Conf. on Case-Based Reasoning. pp. 279–294. Springer (2020)
22. 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)
23. 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)