UPADHYAY, A., MASSIE, S., SINGH, R.K., GUPTA, G. and OJHA, M. 2021. A case-based approach to data-to-text generation. In *Sánchez-Ruiz, A.A. and Floyd, M.W. (eds.) Case-based reasoning research and development: proceedings of 29th International conference case-based reasoning 2021 (ICCBR 2021), 13-16 September 2021, Salamanca, Spain*. Lecture notes in computer science (LNCS), 12877. Cham: Springer [online], pages 232-247. Available from: https://doi.org/10.1007/978-3-030-86957-1_16

A case-based approach to data-to-text generation.

UPADHYAY, A., MASSIE, S., SINGH, R.K., GUPTA, G. and OJHA, M.

2021

The final authenticated version is available online at: https://doi.org/10.1007/978-3-030-86957-1_16. This accepted manuscript is subject to Springer Nature's AM terms of use.

OpenAIR @RGU

This document was downloaded from https://openair.rgu.ac.uk

SEE TERMS OF USE IN BOX ABOVE

DISTRIBUTED UNDER LICENCE

A Case-Based Approach to Data-to-Text Generation

Ashish Upadhyay[1], Stewart Massie[1], Ritwik Kumar Singh[2], Garima Gupta[2], and Muneendra Ojha[2]

[1] Robert Gordon University, Aberdeen, UK
{a.upadhyay,s.massie}@rgu.ac.uk
[2] International Institute of Information Technology, Naya Raipur, India
{ritwik17100,garima17100,muneendra.ojha}@iiitnr.edu.in

Abstract. Traditional Data-to-Text Generation (D2T) systems utilise carefully crafted domain specific rules and templates to generate high quality accurate texts. More recent approaches use neural systems to learn domain rules from the training data to produce very fluent and diverse texts. However, there is a trade-off with rule-based systems producing accurate text but that may lack variation, while learning-based systems produce more diverse texts but often with poorer accuracy. In this paper, we propose a Case-Based approach for D2T that mitigates the impact of this trade-off by dynamically selecting templates from the training corpora. In our approach we develop a novel case-alignment based, feature weighing method that is used to build an effective similarity measure. Extensive experimentation is performed on a sports domain dataset. Through Extractive Evaluation metrics, we demonstrate the benefit of the CBR system over a rule-based baseline and a neural benchmark.

Keywords: Data-to-Text • Textual CBR • Feature Weighting

1 Introduction

Data-to-Text Generation (D2T) is a process that automatically generates textual summary of insights extracted from structured data [25,9]. With business processes often generating 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 [14,10,27] to medical support or sports journalism [20,26,31,4]. D2T is expected to be one of 5 core technologies enabling an economic impact of $5 trillion annually by 2025.

D2T requires two separate problems to be addressed: **content selection**, deciding important content from the input data (implicit or explicit), as in *what to say?*; and **surface realisation**, conveying the selected content into textual summaries, as in *how to say?* Traditional methods use a modular approach to divide the generation task into several smaller modules. These modules are

based on carefully crafted domain-specific rules and templates [25,27]. Recently, neural based learning approaches have shown promising results by integrating all modules into a single end-to-end architecture and learning domain-specific as well as generation rules from parallel corpora of data and summaries [31,21].

Neural systems demonstrate greater fluency and diversity in generated textual summaries but often *hallucinate* by producing inaccurate information that is not supported by the input data. One of the main reasons for hallucination is that the systems have to learn multiple domain specific rules to make sense of input data as well as learn how to verbalise that data [21]. Rule-based systems however are able to produce high quality texts in terms of accuracy but at the expense of diversity in the generated texts. Although, in real-world data-to-text applications, accuracy is usually much more important than fluency and diversity, thus making rule-based systems state-of-the-art in real-world applications.

We propose a Case-Based Reasoning (CBR) approach that learns content selection and realisation separately to generate accurate and diverse texts. Our model learns to choose important entities from the data and then verbalises them via templates extracted from the training corpus. We run experiments to evaluate our proposed method on a sports domain dataset, SportSett [28] and demonstrate it produces better quality texts than neural systems while also maintaining diversity. The contributions are as follows [3]:

1. introduction of a CBR D2T model with separated planning and realisation;
2. development of a novel feature weighting technique using case-alignment that can be used for weighting features in problems with complex solutions;
3. demonstrating with experiments the benefits of our CBR approach over neural and rule-based systems, as well as our case-alignment feature weighting method over information gain feature weighting.

2 Related Works

Natural Language Generation can be divided into two sub-fields based on the input to the systems. The task of generating text from unstructured linguistic data is referred as Text-to-Text generation; while generating text from structured non-linguistic data is known as D2T generation [9]. D2T has been studied for decades. One of the very first systems proposed in 1980s generated textual summaries of financial data [14]. Later other systems were developed to generate weather forecasts [10,27] and medical support documents [26,20]. These were modular systems developed using carefully engineered rules and templates with the help of domain experts dividing the whole task into several sub-tasks. Later in 2000s, statistical learning methods tried combine different modules into a single architecture [13,3]. Some methods also performed just content selection based on statistical methods while utilising rules for surface realisation [4,12].

Advancements in deep learning has boosted the interest in neural-D2T. Several datasets [7,8,31] and systems [31,22,24] have been proposed to utilise neural

[3] The code can be found at https://github.com/ashishu007/data2text-cbr

networks for solving D2T task. Most datasets, are not realistic of real-world challenges as they only verbalise few entities and do not require content selection on the input data. Some datasets, however, such as RotoWire [31] and MLB [22] do reflect real-world challenges and require extensive content selection from input data to verbalise many entities in the textual summaries. Initially, neural systems employed different versions of sequence-to-sequence models to generate text utilising an end-to-end architecture [31,22,24]. These end-to-end systems, despite being able to generate very fluent texts, fared poorly on the accuracy, coherence and structuring of the information in summaries. Pipeline-based systems that separate content-planning from realisation attempt to address some of these issues [6,21].

D2T has also been studied in CBR with systems ranging from weather forecasts to obituary generation [2,1,30], although these systems have been limited to generating smaller texts describing very few entities. As with most CBR systems, similarity is a key component of CBR-D2T systems as effective similarity measures ensure relevant previously solved problems are reused. Feature weighting is one approach to developing an effective similarity measure [11,32], however there are challenges in comparing feature weighting schemes where the problem has a textual solution. Case-alignment measures [15,17] have been employed for feature weighting in classification and regression environments [11]. Our proposed feature-weighting method uses all features to measure the case-alignment of a case-base and utilises this information to assign weights to the features.

3 Background

The data in most D2T task is organised on three dimensions. There are multiple **entities** (for example, players or teams in the case of sports domains), which are described by multiple **features** (points or goals scored in a match), all of which belong to an **event** (a match played between two teams). To simplify, a dataset will have multiple events consisting of multiple entities described by multiple features. Datasets such as RotoWire [31], SportSett [28] and MLB [22] have similar properties. Systems trained on these datasets require extensive content selection on the input data and coherent realisation of longer texts which reflects the challenges of real-world applications.

An example summary from a sports domain dataset, SportSett is shown in Fig. 1a with a subset of its corresponding box-score stat in Fig. 1b. The summaries in such problems have different level of complexities. For example, S12 in Fig. 1a identifies that *Reggie Jackson* scored a 'triple-double' in the game which was the continuation of top performance from the last game. There are at least two types of information that are explicitly not available in the input data:

– first, scoring triple-double is identified if the player scores double-digits in three categories - PTS, REB, and AST. To convey this information, either developers need to explicitly design a rule that identifies if a player scored such a thing (in case of a rule-based system) or the system needs to learn

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 .

(a)

Starters	MP	FG	FGA	FG%	3P	3PA	3P%	FT	FTA	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS	+/-
Kentavious Caldwell-Pope	35:41	7	16	.438	1	4	.250	5	6	.833	1	7	8	2	1	0	3	3	20	-2
Anthony Tolliver	33:50	3	5	.600	2	4	.500	1	2	.500	3	2	5	0	0	1	2	2	9	+5
Andre Drummond	32:17	3	8	.375	0	0		3	3	1.000	7	7	14	0	1	2	1	4	9	-3
Reggie Jackson	32:00	4	17	.235	1	3	.333	2	2	1.000	3	8	11	10	0	0	5	3	11	-4
Caron Butler	17:51	3	8	.375	1	6	.167	0	0		0	0	0	0	2	0	0	1	7	+1

(b)

Fig. 1: (a) An example summary from SportSett dataset. (b) Subset of box-score from the same game.

the rules to infer if a player scored such a thing (in case of a learning-based system). The player scoring triple-double is not explicitly given in the data;
– second, the claim that it was the continuation of top performance from the last game by the same player. To convey this information, the system needs access to the data from previous games as well which is not available for the system during run-time.

There are numerous examples of such situations, where several rules are needed to infer the information either from data of same event or of previous events. There can be a large number of different combinations of data that can be mentioned in a summary. Even in other domains, such as finance or weather reporting, an ample portion of summaries discuss the information aggregated over many entities, features or events [14,27]. This is one of the many reasons for the hallucination of neural systems, as they do not just have to learn the rules of language generation but also have to learn the rules to infer contextual information, as well as learn content selection. On the other hand, for a rule-based system, it can be extremely difficult to write rules for discussing important insights from all the possible combinations from the input data.

There is clear trade-off in current state-of-the-art rule-based and neural-based D2T systems. Based on the requirement of the application, the developed system will either be accurate with monotonous, non-fluent texts, if using rule-based

system; or capable of generating fancy and diverse texts but with some inaccurate information, if using neural system. Our proposed CBR system tries to reduce the impact of this trade-off between accuracy and fluency of texts by learning to reuse previous sentences based on the entities' feature values within an event. It will provide more accurate texts than neural systems without sacrificing diversity.

4 Methodology

The input to a D2T system is a set of records organised by entities and features. Based on these records, a output summary needs to be generated that describes the event to which given entities belong. We assume that a summary is the combination of multiple components organised on higher-level. Typically sports summaries (such as in SportSett [28] and MLB [22]) are organised in four components: **winning team**, a sentence about which team won, with scores; **teams' performance**, few sentences about teams' performance; **players' performance**, several sentences about different players performance from the game; and **next game**, next game fixtures of both teams;

Although, this assumption may be an over-generalisation of the problem, it does apply to a large portion of the SportSett dataset, where more than 90% of the summaries follow similar pattern (with some extra sentences in between). Also, similar components can be identified in other domains [27,12]. The texts in the second and third components (teams' and players' performance) appear to follow the principle of 'similar problems have similar solutions' and thus a CBR approach is used for their generation. The methods described later in the section are mostly centred around these two components. The methodology is briefly shown in the Fig. 2. The first and last components (winning team and next game) use manual rules to select an appropriate template from a bank of around 10 templates. The template bank is created by selecting a few standard sentences from the training set.

4.1 Case-Base Creation

First of all, we create separate case-bases for different components (for players' and teams' performance). The case-base creation is a semi-automated process where first textual summaries are broken into sentences and similar sentences are clustered into the same group. Clusters related to the different components are identified manually and then used to extract the templates [13].

Semantic Clustering We first extract all the sentences from the training set summaries and then abstract them based on their named-entities and pos-tags using the method described in [29]. The abstracting process uses open-source NLP libraries spaCy and neuralcoref combined with some domain-specific rules. Through this process, a sentence from the dataset '*The Atlanta Hawks (41-9) beat the Washington Wizards (31-19) 105-96 on Wednesday.*', is transformed to '*PROPN-ORG (X-Y) beat the PROPN-ORG (X-Y) X-Y on NOUN- DATE.*'.

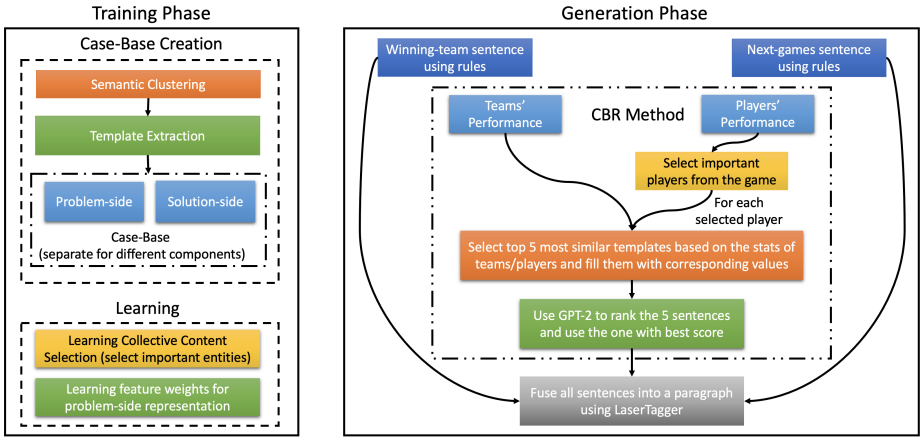

Fig. 2: Our methodology for CBR-D2T

These abstract sentences are then embedded into a 786 dimensional vector using DistilRoBERTa Language Model [4]. We plot the embedded sentences on a 3-dimensional space using the UMAP algorithm [18] and count around 50 clusters based on the plot's view. Then a K-Means clustering algorithm is used to cluster the embedded sentences into 50 similar groups. A manual process is then used to combine the similar clusters and assign them a label identifying the concept cluster items represent. This way we reduce from 50 clusters to 31 clusters, out of which four represent sentences from the teams' component and ten from the players' component (rest 17 contain sentences with more complex facts difficult to classify into just team or player component). Although, the number of clusters could be increased to a higher number to accommodate the possible diversity in the dataset, we leave that for future exploration as manually annotating large number of clusters is a time consuming task.

Template Extraction Template extraction for both the components is done separately but follows the same method. For each sentence in the cluster, the entity mentions are extracted. If a sentence only contains one entity mention then the entity's performance stats is taken from the corresponding game. Based on the stats, an entity matching is performed to replace any occurrence of a entity's feature value to its feature name. For example, from the sentence: "*Henry Sims was able to notch a double - double , contributing 11 points (4 - 12 FG , 3 - 4 FT) and 12 rebounds .*" where Henry Sims' performance stats are: $\{STARTER : no, PTS : 11, FGM : 4, FGA : 12, FG\_PCT : 33, FG3M : 0, FG3A : 0, FG3\_PCT : 0, FTM : 3, FTA : 4, FT\_PCT : 75, OREB : 5, DREB : 7, REB : 12, AST : 2, TO : 0, STL : 0, BLK : 0, PF : 2, MIN : 32, IS\_HOME : no, FIRST\_NAME : Nerlens, SECOND\_NAME : Noel\}$, the template ex-

[4] https://github.com/UKPLab/sentence-transformers

tracted is: "*FIRST_NAME SECOND_NAME was able to notch a double - double , contributing PTS points (FGM – FGA FG , FTM - FTA FT) and REB rebounds*".

For teams' component templates we select sentences with more than two entity mentions (team names) for template extraction. This is done to take sentences that compare the performance of both teams, rather than just discussing one team's performance. Finally, we have two separate case-bases with their respective problem and solution representations. On the problem-side, the entities' performance (box-scores for player component and line-scores for team components) is used, while the extracted template is used for the solution-side representation.

4.2 Retrieval and Feature Weighting

After the case-base is created, the retrieval of similar cases for new problems is done by measuring euclidean distance. We also learn the feature weights for better similarity which is necessary because not all features have equal importance.

Content Selection Central to a D2T task is selection of important contents from the input data. Most of the entities in the input data are not mentioned in the output summary. Even for the entities mentioned in the summary, not all of their features are mentioned. In the SportSett dataset, each game features around 25 players from both teams, but game summaries only discuss 5 to 6 players. Thus, similar to [4] we train a classifier to select important entities from the input data based on their feature values. We use this classifier to select important players from the game. In most cases, importance of an entity is not independent and is related to the feature values of other entities as well. Thus, to represent an entity, we concatenate it's feature values, with the feature values of other entities in the data. So, for an event with e entities with f features, an entity E_1 is represented as: $\{(E_{11}, E_{12}, \cdots, E_{1f}), (E_{21}, E_{22}, \cdots, E_{2f}), \cdots, (E_{e1}, E_{e2}, \cdots, E_{ef})\}$, where E_{11} is E_1's first feature value and E_{ef} is E_e's f^{th} feature value.

An entity is given class 1 (important) if it was mentioned in the summary of that game, or class 0 (not important) if it wasn't mentioned. A classifier is then trained to learn if the entity should be selected for discussing in the summary or not. We train a logistic regression classifier which achieves 87% accuracy and 85% f1 score on the validation set of the SporSett dataset. Finally, the content selection is extended using templates, where after selecting important entities from an event by the classifier, important features are selected using the template of the most similar problem. For example, if a player has scored 'double-double' in the game, it is identified by a feature in the player's stats and a similar template such as '*FIRST_NAME SECOND_NAME lead the way for the PLAYER-TEAM-NAME, recording PTS points on FGM-of-FGA shooting, REB rebounds and AST assists in MIN minutes.*' is extracted which discusses some features in the sentence. This is often synonymous to how summaries are written (at-least in sports domains), as we first tend to select the important player from the game and then decide what features to discuss.

Algorithm 1 Calculate the loss value for each candidate generated in PSO algorithm

Input: PSO candidate $\mathbf{W}$, Problem-side $\mathbf{P}_{CB}$ and Solution-side $\mathbf{S}_{CB}$ of case-base
Output: The loss value for $\mathbf{W}$

$\mathbf{WP}_{CB} = \mathbf{P}_{CB} * \mathbf{W}$
$AlignScore_{CB} = 0$
for each **idx** $\in range(\mid \mathbf{WP}_{CB} \mid)$ **do**
  $\mathbf{P}_T$, $\mathbf{S}_T = \mathbf{WP}_{CB}[idx]$, $\mathbf{S}_{CB}[idx]$
  $\hat{\mathbf{P}_{CB}}$, $\hat{\mathbf{S}_{CB}} = \mathbf{WP}_{CB}[\sim idx]$, $\mathbf{S}_{CB}[\sim idx]$
  Generate problem-side ranked list **PL** using $\mathbf{P}_T$ and $\hat{\mathbf{P}_{CB}}$
  Generate solution-side ranked list **SL** using $\mathbf{S}_T$ and $\hat{\mathbf{S}_{CB}}$
  $AlignScore_{idx} = nDCG(\mathbf{PL}, \mathbf{SL})$
  $AlignScore_{CB} += AlignScore_{idx}$
end for
$\mathcal{L}_W = 1 - (AlignScore_{CB} / \mid \mathbf{WP}_{CB} \mid)$
return $\mathcal{L}_W$

Feature Weighting For the players component data, a classifier is already trained to identify the important players from a game. This classification setting can also be used to learn the feature-importance of players' component (information gain feature weighting). It is noted that this method cannot be applied in a non-classification setting, such as in teams' component. Thus, a novel case-alignment based feature weighting method is proposed for non-classification settings. CBR systems are based on the principle of 'similar problems have similar solutions' and case-alignment can provide a measure of the extent to which this principle holds true for a specific design e.g. feature weighting scheme.

We use the method proposed in [30] for measuring the case-alignment of the case-base. The alignment score is then used as a loss function for a Particle-Swarm Optimiser whose parameters are the features' weights for a case-base. The loss function is formally defined in Algorithm 1. The ranked list on the problem side is generated using the euclidean distance between the problem-side of the target problem and the cases in the case-base, while the solution-side ranked list is generated using the cosine distance between the solution-side of target problem and cases in the case-base.

4.3 Generation

Generation again is done separately for different components.For a target problem, k-nearest neighbours are retrieved from the case-base using problem-side representation. k solutions are generated by filling the tags with their corresponding values in the nearest neighbour solutions. A GPT-2 [23] language model is used to rank the five sentences based on perplexity score. The best among the five is chosen as the final solution for the given target problem. Since sentence ranking is a domain specific task, the GPT2 model is fine-tuned on the training set of the same dataset used in our experiments. Now several sentences are generated for different components and are fused into a paragraph using sentence

fusing algorithm LaserTagger [16] trained on the same training data used in our experiments.

In the case of SportSett data, for the players' component: first, important players are selected using the classifier mentioned in an earlier section; then, for each important player selected, a sentence is generated using the process described above. Similarly a sentence is generated for the teams' component. A set of rules is used to generate the first component sentence describing which team won, and another set of rules to generate a sentence for both teams' next fixtures. Finally, all these sentences are fused into a paragraph using LaserTagger.

5 Experimental Setup

5.1 Dataset

The SportSett dataset [28] is used to evaluate our proposed and benchmark algorithm. It contains textual summaries combined with the box- and line-scores of NBA matches. The training set contains the matches from 2014, 2015 and 2016 seasons (4745 instances) while the dev and test sets contain matches from 2017 and 2018 seasons (1228 and 1229 instances) respectively.

We use the train set of SportSett for the creation of our case-bases. For training the important players classifier, we used the train set of SportSett with the dev set for testing. For fine-tuning the GPT2 and LaserTagger, we also used just the texts from the train set of SportSett. Similarly, only train set data is used for creating the case-bases for teams' and players' components. With all seasons used from train set, the teams' component case-base consists of 1200 cases, while players' component has 14985. With just 2014 season used for training, 360 and 4405 cases are available in teams' and players' case-bases

5.2 Baseline and Benchmark

We compare our system with a rule based baseline and a neural benchmark:

- **Rule-Based System** is the templatized generator used as baseline in [31]. The system has a standard template for winning team, another template for players stats which is filled with six highest scoring players' stats, and finally last template for teams next-game fixture. We extend the rule-based system to include day name and arena of the game, as well as next-opponent team names since this new information is now available in the SportSett data.
- **Neural System** is a sequence-to-sequence model proposed in [22]. It consists of an MLP encoder and LSTM decoder with copy mechanism. There's an added module to update the input record's representation during generation process. At each decoding step, a GRU is used to decide the record that needs to be updated and updates it's value

Although, there are other neural systems with comparable performance to our selected benchmark, we use the selected one because of its reduced training

time and ease in reusing the code. For example, authors in [24] proposed a hierarchical Transformer encoder model with standard LSTM decoder which achieves slightly better performance than our selected benchmark. But it takes 10 days to train with the hyper-parameters mentioned in the paper on a 16GB Nvidia-P100 GPU compared to our selected benchmark which takes just 1 day.

5.3 Evaluation Methods

We use the family of Extractive Evaluation (EE) metrics for [31] for evaluating the models. These metrics are trained to extract entity names and numerical values from the text and predict the relation (feature name) between them. For example, from S05 in Fig. 1a, the IE models (ensemble of three LSTMs and three CNNs) can extract entity name *Ish Smith* and numerical value *15*. Then the model can predict its relation name as *PTS* (points) and will return a tuple as $t = (EntityName|FeatureValue|FeatureName)$ as $(IshSmith|15|PTS)$. The models extract several tuples from both human-written gold (y) and system generated ($\hat{y}$) summaries. These tuples are then compared to calculate the following metric scores:

- **Relation Generation (RG)** is the precision of unique tuples t extracted from generated summary $\hat{y}$ that also appeared in the input data. This metric can be used to measure the system's capability of generating factually correct texts supported by input data, i.e., accuracy of the system.
- **Content Selection (CS)** is the precision and recall between unique tuples t extracted from gold summary y and generated summary $\hat{y}$. Here, the systems ability of selecting content is measured in comparison with the human written summaries.
- **Content Ordering (CO)** is measured as the normalized Damerau Levenshtein Distance [5] between the sequences of tuples extracted from generated summary $\hat{y}$ and gold summary y. This demonstrates the systems ability of ordering the content in generated summary.

CS primarily targets the challenge of *what to say?*, while CO targets the *how to say it?* aspect. Apart from these metrics, we also use BLEU [19] score to compare the generations. BLEU score compares the n-gram overlap between the gold summary and generated summary and primarily rewards fluent texts rather than generations capturing more information from the input [31]. From all the metrics discussed here, RG can be used to measure the factual correctness of generations. While we acknowledge the fact that human judgement is the best evaluation practice for text generation systems, these autonomous metrics are a widely used proxy methods as crude surrogate for human judgement.

Initial versions of the Extractive Evaluation metrics [31,21] only evaluated the numerical claims (such as points, rebounds, steals made by a player or team) mentioned in the text summaries. Authors in [29], proposed an extended version capable of evaluating day names, dates and game arenas as well. We further extend these metrics to evaluate the few more claims made in texts, for example, if a player was the leading scorer or if a player scored a double-double.

Table 1: Comparison of our CBR system with baseline and benchmark

System	RG	CS-Precision	CS-Recall	CO	BLEU
Gold	89.46	–	–	–	–
Rule-Based	95.35	55.20	23.65	10.61	6.50
Neural$_{oneS}$	61.34	34.84	25.68	9.84	9.93
Neural$_{allS}$	71.07	45.66	40.81	19.56	17.68
CBR$_{oneS}$	73.22	50.18	24.78	10.91	10.40
CBR$_{allS}$	77.22	46.46	33.53	11.92	11.93

6 Results and Discussion

6.1 Comparison with Benchmark and Baseline

We train our CBR system and neural system on two different sizes of training data. First, we use only 2014 season data for training the models, denoted by **(Neural/CBR)**$_{oneS}$. Then, we use all season data from train set for training which is denoted by **(Neural/CBR)**$_{allS}$. We compare these four systems and a rule-based baseline discussed in the previous section. The results are shown in Table 1. Results for the neural system are given as the average over 10 training runs with different random seeds.

First of all, we note that our **CBR**$_{oneS}$ outperforms **Neural**$_{oneS}$ on all metrics, except CS-recall. This demonstrates that our CBR system is much better at producing quality texts compared to neural system, even with fewer training samples. However, with the increase in training data, there is a huge gain in **Neural**$_{allS}$ across all metrics while **CBR**$_{allS}$ system improves very little. Still with any amount of data, our CBR system achieves better performance on RG (73% & 77% for CBR vs 61% & 71% for neural) and CS-precision (50% & 46% for CBR vs 34% & 45% for neural) metrics as compared to the neural system, indicating CBR system's benefit over neural on accuracy. We also note that rule-based system achieves best score on RG (95%) and on CS-precision (55%). This is not surprising as the system is hard-coded with domain knowledge thus has very low accuracy errors [5]. But system fares poorly on mimicking the gold summaries, as it only recalls 23% of the contents from gold summaries. It also performs poor on the BLEU score which is conveys that rule-based system is not very fluent.

On the terms of fluency, we observe that BLEU score achieved by both CBR systems is better than the rule-based baseline. As compared to neural system, **CBR**$_{oneS}$ is slighlty better than **Neural**$_{oneS}$, while for **Neural**$_{allS}$, BLEU score is quite higher than **CBR**$_{allS}$. This reflects that our system is much fluent compared to rule-based system, as well as above-par to neural system when training data is scarce. To measure the diversity, we first calculate the vocabulary of texts generated from different systems. We identify that the gold summaries from test

[5] please note that the scores are not 100% because the metrics are based on trained models, which themselves achieve around 90% accuracy and f1 score while training

Table 2: Ablation study results

	RG	CS-Precision	CS-Recall	CO	BLEU
Info Gain FtrW	75.22	49.53	28.73	11.78	10.84
without GPT-2 scoring	76.09	44.64	34.88	12.33	9.26
without LaserTagger	76.36	44.50	33.04	11.30	11.04
CBR$_{allS}$	77.22	46.46	33.53	11.92	11.93

set have a vocabulary of more than 5000 words, while both the CBR and neural systems have the vocabulary of 2000 words but rule-based system has vocabulary of only 900 unique words. In terms of content selected for output summary, the rule-based system is only able to discuss one-third of the unique record types discussed in gold summary, while our CBR system is able to discuss all of them. All this evidence suggests that the CBR system is able to decrease the trade-off between accuracy and diversity, especially in case of scarcity of training data.

6.2 Ablation Studies

We further perform three ablation studies on our **CBR**$_{allS}$ system. In the first study, we analyse the effect of our proposed case-alignment based feature weighting against the information gain based feature weighting (see Section 4.2). The results of this ablation are shown in the first row of Table 2. Here information gain from the important player classifier data is used to weight the features in players' component, however, no weighting is applied for teams' component. From the results, we can see that apart from CS-precision, there's at-least some drop in all metrics while a sizeable drop in CS-recall. The drop in CS-recall can mean that the system with case-alignment feature weighing is able to select templates that have contents closer to the human written summaries.

In the second study, we compare the effect of selecting the 'nearest neighbour' against 'best out of top-k nearest neighbour' for generating the new solution, of which results are shown in second row of the Table 2. Here, the **CBR**$_{allS}$ system is used without GPT2 solution ranking module. Again, we can see there's not much difference in EE metrics but there's a sizeable difference in BLEU score. This is expected as GPT2 scores the sentences based on perplexity that rewards fluency. So with the addition of an extra scoring component for solution reuse, we can improve the fluency of our generated text summaries.

Third study analyses the effects of applying LaserTagger for sentence fusion. Results are shown in the third row of the Table 2. We can see that there's slight drop in most metrics when LaserTagger is not used for sentence fusion. This is because that the texts generated from CBR systems have some incohereny such as: '*Bradley Beal led the way for* ***the Wizards (32-48)*** *with 25 points, complementing* ***the Wizards (32-48)*** *with five assists and two rebounds.*'. This

The visiting Atlanta Hawks (13 - 29) defeated the host Philadelphia 76ers (27 - 16) 123 - 121 at Wells_Fargo_Center on Friday . The 76ers shot 52 percent from the field , including 33 percent from long range but were not able to hang on for the full 48 , as The Hawks surged back to get their revenge . Ben Simmons had a triple - double with 23 points (7 - 13 FG , 1 - 2 3Pt) , 10 rebounds , 15 assists , three steals and one block in 43 minutes . Meanwhile , Jimmy Butler was the high - point man for Philadelphia , with 30 points on 9 - of - 19 shooting , in 40 minutes . JJ Redick was next in line with 20 points , three rebounds , an assist and a steal , as the only other 76ers player who managed double - digit points . Rookie DeAndre' Bembry shot 6 - for - 11 from the field to score 14 points , while also chipping in five rebounds . Kevin Huerter was the high - point man for the Hawks as he tied a season - high with 29 points on 11 - of - 17 shooting , including 5 - of - 8 from long range . John Collins was the 3 prong of the Hawks attack , as he finished with 25 points (10 - 17 FG , 1 - 1 3PT , 1 - 3 FT) , along with five rebounds , two assists , two steals and one block , in 27 minutes . The 76ers now head to New_York for a Sunday night showdown versus the Knicks while the Hawks will return home to face the Bucks on Sunday .

Table 3: A summary generated from our $\mathbf{CBR}_{allS}$ system

incoherence is the result of using co-reference resolution while template extraction. With LaserTagger applied to the above sentence, it is modified into '*Bradley Beal led the way for* ***the Wizards*** *with 25 points, complementing* ***them*** *with five assists and two rebounds.*'. This is similar to the Referring Expression Generation phase of traditional D2T methods [25].

6.3 Qualitative Analysis

A summary of the NBA match between 76ers and Hawks on 11th Jan, 2019 generated from the proposed system is shown in Table 3. The first and last sentences are generated from a set carefully crafted rules, while other sentences are generated through the CBR methodology. The accurate facts are shown in green while the inaccurate ones in red. The red inaccuracies are due to the imperfection in template extraction process, where sometimes an entity is replaced with wrong feature name because of more than one features having the same value. By addressing such cases in template extraction can improve the accuracy of the system in terms on numerical facts being conveyed. Another type of inaccuracy is shown in grey sentence, which wrongly mentions JJ Redrick being the only player with double-digit points after Jimmy Butler. These facts are much more complex to calculate and are grounded in the text, as this information is calculated across multiple entities (you need to know other players' scores as well to decide only two scored in double-digits). To address such errors, new features are needed to explicitly identify the information from across entities and/or events. Those new features will help the system in deciding the similarity of a template with across-entity information in such cases. One interesting observation is that no discourse is more than a sentence long, that's because the template extraction is

done on a sentence level. Extraction on sub-paragraph level will make discourse longer and summaries much more human-like.

7 Conclusion and Future Work

In this work, we proposed a CBR system for Data-to-Text generation that aims to provide a good balance in the trade-off between accuracy and diversity of the text summaries produced by a D2T system. Our CBR system follows a modular approach to text generation in which content selection is performed first before surface realisation takes place via templates extracted from the training corpora. Experimentation results on a sports domain data-set show that our CBR system achieves better accuracy than a neural benchmark while better fluency and diversity than a rule-based baseline. We also introduce a novel case-alignment based feature weighting algorithm that is particularly effective in non-classification settings, such as text solutions. The benefit of our feature weighing algorithm over the information gain feature weighing baseline is also demonstrated.

In future, we would like to improve the template extraction process. In this paper, the extraction take place at a sentence level (micro-plan) which limits performance in relation to inter-sentence coherence in the generated texts. With an extraction process on sub-paragraph level (macro-plan) better coherence and discourse could be achieved for describing multiple entities in the summary. We would also like to investigate the possibility of having dynamic higher-level component organisation to increase the diversity in content structuring of the summaries. Finally, we plan to conduct more extensive evaluation of the system with alternative data-sets and using human judgement for evaluations.

References

1. Adeyanju, I.: Generating weather forecast texts with case based reasoning. arXiv preprint arXiv:1509.01023 (2015)
2. Adeyanju, I., Wiratunga, N., Lothian, R.: Learning to author text with textual cbr. ECAI (2010)
3. Angeli, G., Liang, P., Klein, D.: A simple domain-independent probabilistic approach to generation. In: Proceedings of EMNLP. pp. 502–512 (2010)
4. Barzilay, R., Lapata, M.: Collective content selection for concept-to-text generation. In: Proceedings of HLT-EMNLP. pp. 331–338. (2005)
5. Brill, E., Moore, R.C.: An improved error model for noisy channel spelling correction. In: Proceedings of ACL. p. 286–293. (2000)
6. 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: Proceedings of EMNLP-IJCNLP. pp. 552–562 (2019)
7. Colin, E., Gardent,C., M'rabet,Y., Narayan,S., Perez-Beltrachini,L.: The webnlg challenge: Generating text from dbpedia data. In: Proc of INLG. pp.163–167 (2016)
8. Dušek, O., Novikova, J., Rieser, V.: Evaluating the State-of-the-Art of End-to-End Natural Language Generation: The E2E NLG Challenge. pp. 123–156 (2020)
9. Gatt, A., Krahmer, E.: Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. JAIR 61, 65–170 (2018)

10. Goldberg, E., Driedger, N., Kittredge, R.: Using natural-language processing to produce weather forecasts. IEEE Expert 9(2), 45–53 (1994)
11. Kar, D., Chakraborti, S., Ravindran, B.: Feature weighting and confidence based prediction for case based reasoning systems. In: ICCBR. pp. 211–225. (2012)
12. Kelly, C., Copestake, A., Karamanis, N.: Investigating content selection for language generation using machine learning. In: Proc. of ENLG. pp. 130–137 (2009)
13. Kondadadi, R., Howald, B., Schilder, F.: A statistical nlg framework for aggregated planning and realization. In: Proceedings of ACL. pp. 1406–1415 (2013)
14. Kukich, K.: Design of a knowledge-based report generator. In: ACL. pp. 145–150 (1983)
15. Lamontagne, L.: Textual cbr authoring using case cohesion. In: Proceedings of TCBR Workshop at the 8th ECCBR. pp. 33–43 (2006)
16. Malmi, E., Krause, S., Rothe, S., Mirylenka, D., Severyn, A.: Encode, tag, realize: High-precision text editing. In: Proc of EMNLP-IJCNLP. pp. 5054–5065 (2019)
17. Massie, S., Wiratunga, N., Craw, S., Donati, A., Vicari, E.: From anomaly reports to cases. In: ICCBR. pp. 359–373 (2007)
18. McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)
19. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of ACL. pp. 311–318 (2002)
20. 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)
21. Puduppully, R., Dong, L., Lapata, M.: Data-to-text generation with content selection and planning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 6908–6915 (2019)
22. Puduppully, R., Dong, L., Lapata, M.: Data-to-text generation with entity modeling. In: Proceedings of ACL. pp. 2023–2035 (2019)
23. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI blog 1(8), 9 (2019)
24. Rebuffel, C., Soulier, L., Scoutheeten, G., Gallinari, P.: A hierarchical model for data-to-text generation. In: ECIR. pp. 65–80. Springer (2020)
25. Reiter, E., Dale, R.: Building natural language generation systems (2000)
26. Reiter, E., Robertson, R., Lennox, A.S., Osman, L.: Using a randomised controlled clinical trial to evaluate an NLG system. In: Proc. of ACL. pp. 442–449 (2001)
27. Sripada, S., Reiter, E., Davy, I.: Sumtime-mousam: Configurable marine weather forecast generator. Expert Update 6(3), 4–10 (2003)
28. Thomson, C., Reiter, E., Sripada, S.: Sportsett: Basketball-a robust and maintainable dataset for natural language generation. In: IntelLanG (2020)
29. Thomson, C., Zhao, Z., Sripada, S.: Studying the impact of filling information gaps on the output quality of neural data-to-text. In: Proc. of INLG. pp. 35–40 (2020)
30. Upadhyay, A., Massie, S., Clogher, S.: Case-based approach to automated natural language generation for obituaries. In: ICCBR. pp. 279–294. Springer (2020)
31. Wiseman, S., Shieber, S.M., Rush, A.M.: Challenges in data-to-document generation. In: Proceedings of EMNLP. pp. 2253–2263 (2017)
32. Zhu, G.N., Hu, J., Qi, J., Ma, J., Peng, Y.H.: An integrated feature selection and cluster analysis techniques for case-based reasoning. Engineering Applications of Artificial Intelligence 39, 14–22 (2015)