A Automated Metric Results

We include results on common metrics of BLEU, Relation Generation (RG), Content Selection (CS), and Content Ordering (CO) for this task in Table 1. Automatic metrics are often expected in NLP papers, although their usefulness in this domain is limited at best. We include them in the appendix for this reason.

The V-SIMPLE and MP-SIMPLE systems, based on simple schema, had the highest RG scores, and hierarchical systems the lowest. Interestingly, CO scores are highest when models follow extracted schema from gold texts.

BLEU scores are within a narrow range, with Mathur et al. (2020) having shown that larger differences are required in order to make judgments. The information extraction based metrics prove more useful, with Wiseman et al. (2017) stating that their results were generally inline with their human evaluation. However, Thomson and Reiter (2021) observed that state-of-the-art metrics can detect simple errors, but struggle with more complex semantic and contextual errors. It is also worth noting that running BLEU on a deranged copy of the test set (comparing each game with a random game other than itself) can yield BLEU scores in the region of 8.0 to 10.0, simply due to common terminology and syntax.

System	RG		CS			CO	BLEU
	P%	#	P%	R%	F1		
REF	0.84	26.84	-	-	-	-	-
V-SIMPLE	0.87	26.21	0.60	0.57	0.58	0.21	19.68
V-GUIDED	0.81	17.56	**0.71**	0.48	0.57	**0.30**	17.29
V-EXTENDED	0.84	27.06	0.57	0.58	0.57	0.21	21.90
MP-SIMPLE	**0.88**	43.27	0.48	**0.73**	0.58	0.22	21.52
MP-GUIDED	0.82	30.02	0.60	0.67	**0.63**	**0.30**	**22.27**
H-FULL	0.76	27.76	0.42	0.47	0.44	0.16	17.73
H-NEXT	0.77	23.09	0.51	0.47	0.49	0.18	21.22

Table 1: Automatic metric results for all systems.

B Content Ordering Experiment

This experiment aims to determine whether sentences in generated summaries are in the correct order. In designing this experiment we had two main concerns. Firstly, inter-annotator agreement should at least be moderate, ideally high. This precludes designs where participants are free to rearrange all sentences; the large number of permutations increases the likelihood of disagreement. Secondly, it should be possible to perform meaningful error analysis in order to better understand both the systems, and the protocol itself. This rules out Likert-based approaches because, with paragraph-sized generations, it is impossible to tell which part of the summary caused a participant to score the text in the way they did. Likert ratings have been shown to have poor agreement in this domain (Puduppully and Lapata, 2021).

B.1 Design

We presented generated summaries to participants with the first two sentences highlighted as 'the beginning', the final two sentences highlighted as 'the end', and everything in between highlighted as 'the middle'. We then asked participants, for each of the four sentences in the beginning and end, whether it should:

- **Remain** where it is.
- be **Transposed** with its partner, i.e., the other sentence from the beginning or end.
- be moved to the middle, a **Short** distance.
- be moved to the opposing end of the summary, a **Long** distance.

When asked if sentences should be moved to another section participants did not specify exactly where, simply which other section. We also asked the middle was in an acceptable order (Yes/No).

Participants were placed into 35 non-exclusive groups (the number of combinations of size three for 7 participants). Each group evaluated a summary from each of the 7 systems, such that 245 unique summaries were evaluated by 3 annotators.

B.2 Results

For content ordering, we first consider whether participants believed a sentence should be moved to a different section. Inter-annotator agreement by Fleiss Kappa (Fleiss, 1971) was 0.591, indicating a moderate agreement. However, this falls to 0.469 when we consider the *Short*/*Long* move distances, and to 0.350 if we also consider transposition of beginning/end sentence pairs (p-value was less than 0.001 in all cases). This confirms our design assumption that allowing participants to freely rearrange texts of this length would result in low or no agreement. We did run an experiment where different participants (MTurk masters with US high-school diplomas) were asked to rate how readable and understandable generations were. Agreement for this was even lower, below 0.2, and results are not included for that reason.

System	Long	Short	Transpose	Remain
V-SIMPLE	1	3	55	361
V-GUIDED	1	33	10	372
V-EXTENDED	0	10	59	351
MP-SIMPLE	3	15	4	398
MP-GUIDED	3	53	7	353
H-FULL	2	65	1	352
H-NEXT	1	88	4	327

Table 2: Number of sentences that annotators would move, by destination.

B.3 Conclusion

The results in Table 2 show that all models do a good job at avoiding *Long* errors, that is they do not confuse the beginning of the narrative with the end. The simple schema of both V-SIMPLE and MP-SIMPLE have fewer *Short* errors, especially compared with the hierarchical encoder systems. Our models in V-SIMPLE and V-EXTENDED mode *Transpose* sentences in the *beginning* or *end* with higher frequency. Looking into this further, our schema (for both models) was set to realize the upcoming game for the winning team in the *Penultimate* sentence, then the losing team in the *Final* sentence. This was deemed incorrect by some annotators (the losing teams players are usually discussed immediately before the end, therefore the context at that stage is the losing team). Our system is capable of adjusting for this, with a simple schema change reversing the order of these sentences. The MP-SIMPLE system does not have the fine-grained control to constrain generation to two separate sentences, therefore it frequently discusses both teams upcoming games in a single *Final* sentence and does not encounter this *Transpose* problem as often as our models. It is also unclear how the *Short* errors of such a system could be corrected.

This experiment is included in the appendix because whilst it was unsuccessful at demonstrating a difference between systems (agreement was low), it does provide some insight and with some refinement of experimental design could be a useful approach (agreement was not so low that there are no possible pathways to higher agreement).

C Post-hoc error analysis

In addition to the quantitative data, our accuracy evaluation yielded qualitative data in the form of free-text comments that annotators could leave when reporting each error. We therefore performed an error analysis, something that is crucial to to gain insight into where our systems are failing (van Miltenburg et al., 2021, 2023). With the MP-SIMPLE and MP-SIMPLE systems some annotators queried the protocol because some names were spelled incorrectly. This had not been a problem for word-based systems, but since the system of Puduppully and Lapata (2021) operates at the subword level, it would sometimes generate texts that contained out of vocabulary words once subwords were reconstructed. An example can be seen in the sentence: *"**Well ell ell ell ell ell ell Carter**^{N} , as he scored 25 points to go along with eight rebounds and five assists ."*, where *"ell"* is an out-of-vocabulary word. The annotator for this sentence marked it as an error, leaving the mildly derisive comment of *"more commonly referred to as just **Wendell Carter**^{N}"*. Upon further investigation, this problem is not uncommon in the generations of this system, yet it would be missed by the RG metric and at times our human evaluation as well[10]. In one of the worst cases (from the full test set, not an item from our human evaluation), the complete generation was: *"The Miami Heat (27 - 33) defeated the Golden State Warriors (43 - 18) 126 - 125 on Friday . Justise Winslow and Bam AAAAAAAAAAA"*, followed by the letter *'b'* repeated 808 times. Our view based systems also struggled at times to generate full sentences about players such as Bam Adebayo, who had not been seen during training. For example, one output was *"Bam Adebayo, it **wasn't enough to overcome the Heat**^{W}."*, where the model knew it should generate a sentence about Bam Adebayo, but did not include any statistics. It is possible the models are relying on the values of the player name field rather than generalizing.

To gain further insight, we performed some automated error analysis on outputs from the full test set (2018 season). Table 4 shows the average token counts and out of vocabulary[11] tokens for the generations of each system. Our view based systems each generated a small number of out-of-vocab tokens by erroneously copying boolean values from the input data (we would fix this by

[10]The factual accuracy annotation instructions of Thomson and Reiter (2020) ask annotators to ignore spelling, syntax and grammar, so some annotators did not mark these as errors (if they could make out which player was being referred to).

[11]A vocabulary was created using all test data values, training data texts and a range of numbers in word and digit form.

System	NAME	NUMBER	WORD	CONTEXT	OTHER	NOT CHECKABLE	TOTAL
V-SIMPLE	44	115	134	16	19	11	339
V-GUIDED	76	233	153	18	16	14	510
V-EXTENDED	60	218	206	18	30	17	549
MP-SIMPLE	195	79	91	22	6	5	398
MP-GUIDED	186	129	134	33	29	2	513
H-FULL	109	232	186	14	32	2	575
H-NEXT	113	232	243	24	38	2	652

Table 3: Errors for each system by type. Systems that were guided by simple schema (V-SIMPLE, MP-SIMPLE) produced the fewest factual mistakes whilst offering the most control.

only including lexical values as input data values). The references texts had out-of-vocab tokens because human authors are not constrained to the set of training words. The MP-SIMPLE and MP-SIMPLE systems both had more out of vocabulary words. Also shown is a count of singleton trigrams (where all three tokens in the trigram are identical), a measure of repetition, where gain the MP-SIMPLE and MP-SIMPLE systems had higher mean counts. In both cases, this is likely due to the incorrect recombination of subwords. It may be possible to adjust the training of models to aleviate this, but it is important to note that automatic metrics all miss this kind of error and it was only found because of our error analysis of human annotated errors.

Shot breakdowns, which are a type of domain specific syntax breaking down the shooting of a player using between 2 and 6 numbers, e.g. *"(4-8 FG, 1-4 3Pt, 2-2 FT)"*, were also counted in Table 4. The number of shot breakdowns (extracted by regular expression) included by the MP-SIMPLE and MP-GUIDED systems could explain part of the increased RG# seen in Table 1. They densely transcribe either 2, 4, or 6 numeric facts yet are simple (once the decision has been made to include one, the structure is deterministic). Systems learn to generate so many shot breakdowns because that they are present in the training data, although they are seldom found in the test set reference texts from the 2018 season. This could be explained by drift due to a change in the specific authors writing the reference texts during that year (Upadhyay and Massie, 2022).

D Crowd-sourced worker recruitment

Participants were recruited on the Amazon Mechanical Turk platform. We used the recruitment policy of Thomson and Reiter (2020) participants were required to hold a US Bachelors degree, be US residents, and be Mechanical Turk Masters workers (a qualification issues by Amazon for high worker reliability). In addition, candidates had to complete a (paid) custom qualification exercise. Fair treatment of crowd-sourced workers is important (Silberman et al., 2018) both from an ethical standpoint and to ensure high quality work. We aimed to pay workers approximately US$20 per-hour for their time, which meant paying $8 for each of the 35 factual accuracy annotation tasks they completed, these take 20-25 minutes to complete. We paid $2 for each of the ordering tasks which take 5-6 minutes to complete. We also paid the same for the any practice work. The same 7 participants completed all work for both our factual accuracy and ordering experiments.

E View Grounding

Given a sentence, we consider all possible view sets as candidates for grounding. We propose to judge the alignment between one view set and the sentence as inversely proportional to the number of *alignment errors* it would entail. An alignment error simply refers to any token that could belong to one of the generated noun phrases but cannot be justified by the data contained in the view set.

To identify individual alignment errors, we first use a simple rule-based system to generate noun phrases based on the data within the view set. This includes phrases based on statistics like '14 points', or alternative forms such as '14-point'. We also include those derived from multiple statistics, e.g., 'double-double'. Named entities are also included, for example, 'Russel Westbrook'. This does introduce a requirement of manual definition, but generating noun phrases for data is a much simpler task than constructing grammar and narrative to connect them. We take the best of both rules and neural, defining that what which is simple and learning that which is complex or time-

	Token Count		Out-of-Vocab Count		Singleton Trigram Count		Shot Breakdown Count	
System	mean	stdev	mean	stdev	mean	stdev	mean	stdev
V-SIMPLE	276	30	0.027	0.19	0.04	0.237	0.151	0.663
V-GUIDED	241	48	0.022	0.147	0.013	0.145	0.178	0.636
V-EXTENDED	340	33	0.026	0.179	0.05	0.253	0.229	0.817
MP-SIMPLE	292	62	2.673	3.625	0.386	3.008	1.551	2.185
MP-GUIDED	309	95	2.108	5.449	0.63	6.157	0.83	1.948
H-FULL	366	71	0	0	0	0	0.191	0.774
H-NEXT	386	94	0	0	0	0	1.142	2.008
GOLD	339	39	0.618	0.958	0	0	0.008	0.134

Table 4: Mean count and standard deviation of tokens, out-of-vocabulary tokens, singleton trigrams (where the set of tokens within the trigram is a singleton), and shot breakdowns per-text.

consuming. Each sentence is parsed token-wise, and once a known noun-phrase (from a global list) is started, it must be able to continue within that view ('14' can continue as '14 points' or '14 - point'), or conclude ('14 - point' must conclude as there is no possible continuation), otherwise it is an error. There will be a small number of cases where the grounding cannot be narrowed down to 1 or 2 compatible views. However, all we require is enough correctly grounded views to introduce a training signal. When there is ambiguity, a model can be instructed to not update weights.

We conclude the view set selection procedure by selecting the smallest one, i.e. in this case the singleton of Westbrook's `<Whole-Game>` view (which had zero errors).