Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 6645–6650
Marseille, 11–16 May 2020

GameWikiSum: a Novel Large Multi-Document Summarization Dataset

Diego Antognini, Boi Faltings
Artificial Intelligence Laboratory
École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
firstname.lastname@epfl.ch

Abstract

Today's research progress in the field of multi-document summarization is obstructed by the small number of available datasets. Since the acquisition of reference summaries is costly, existing datasets contain only hundreds of samples at most, resulting in heavy reliance on hand-crafted features or necessitating additional, manually annotated data. The lack of large corpora therefore hinders the development of sophisticated models. Additionally, most publicly available multi-document summarization corpora are in the news domain, and no analogous dataset exists in the video game domain. In this paper, we propose GameWikiSum, a new domain-specific dataset for multi-document summarization, which is one hundred times larger than commonly used datasets, and in another domain than news. Input documents consist of long professional video game reviews as well as references of their gameplay sections in Wikipedia pages. We analyze the proposed dataset and show that both abstractive and extractive models can be trained on it. We release GameWikiSum for further research: `https://github.com/Diego999/GameWikiSum`.

Keywords: multi-document summarization, video games, professional reviews, wikipedia

1. Introduction

With the growth of the internet in the last decades, users are faced with an increasing amount of information and have to find ways to summarize it. However, producing summaries in a multi-document setting is a challenging task; the language used to display the same information in a sentence can vary significantly, making it difficult for summarization models to capture. Thus large corpora are needed to develop efficient models. There exist two types of summarization: extractive and abstractive. Extractive summarization outputs summaries in two steps, namely via sentence ranking, where an importance score is assigned to each sentence, and via the subsequent sentence selection, where the most appropriate sentence is chosen. In abstractive summarization, summaries are generated word by word auto-regressively, using sequence-to-sequence or language models. Given the complexity of multi-document summarization and the lack of datasets, most researchers use extractive summarization and rely on hand-crafted features or additional annotated data, both needing human expertise.

To our knowledge, Liu et al. (2018) is the only work that has proposed a large dataset for multi-document summarization. By considering Wikipedia entries as a collection of summaries on various topics given by their title (e.g., Machine Learning, Stephen King), they create a dataset of significant size, where the lead section of an article is defined as the reference summary and input documents are a mixture of pages obtained from the article's reference section and a search engine. While this approach benefits from the large number of Wikipedia articles, in many cases, articles contain only a few references that tend to be of the desired high quality, and most input documents end up being obtained via a search engine, which results in noisy data. Moreover, at testing time no references are provided, as they have to be provided by human contributors. Liu et al. (2018) showed that in this case, generated summaries based on search engine results alone are of poor quality and cannot be used.

In contrast, we propose a novel domain-specific dataset containing 14 652 samples, based on professional video game reviews obtained via Metacritic[1] and gameplay sections from Wikipedia. By using Metacritic reviews in addition to Wikipedia articles, we benefit from a number of factors. First, the set of aspects used to assess a game is limited and consequently, reviews share redundancy. Second, because they are written by professional journalists, reviews tend to be in-depth and of high-quality. Additionally, when a video game is released, journalists have an incentive to write a complete review and publish it online as soon as possible to draw the attention of potential customers and increase the revenue of their website (Zhou and Duan, 2010). Therefore, several reviews for the same product become quickly available and the first version of the corresponding Wikipedia page is usually made available shortly after. Lastly, reviews and Wikipedia pages are available in multiple languages, which opens up the possibility for multilingual multi-document summarization.

2. GameWikiSum

In this section, we introduce a new domain-specific corpus for the task of multi-document summarization, based on professional video game reviews and gameplay sections of Wikipedia.

2.1. Dataset Creation

Journalists are paid to write complete reviews for various types of entertainment products, describing different aspects thoroughly. Reviewed aspects in video games include the gameplay, richness, and diversity of dialogues, or the soundtrack. Compared to usual reviews written by users, these are assumed to be of higher-quality and longer.

[1] `https://www.metacritic.com/game`

Metacritic[2] is a website aggregating music, game, TV series, and movie reviews. In our case, we only focus on the video game section and crawl different products with their associated links, pointing to professional reviews written by journalists. It is noteworthy that we consider reviews for the same game released on different platforms (e.g., Playstation, Xbox) separately. Indeed, the final product quality might differ due to hardware constraints and some websites are specialized toward a specific platform[3].
Given a collection of professional reviews, manually creating a summary containing all key information is too costly at large scale as reviews tend to be long and thorough. To this end, we analyzed Wikipedia pages for various video games and observed that most contain a gameplay section, that is an important feature in video game reviews. Consequently, we opt for summaries describing only gameplay mechanics. Wikipedia pages are written following the Wikipedia Manual of Style[4] and thus, guarantee summaries of a fairly uniform style. Additionally, we observed that the gameplay section often cites excerpts of professional reviews, which adds emphasis to the extractive nature of GameWikiSum.
In order to match games with their respective Wikipedia pages, we use the game title as the query in the Wikipedia search engine and employ a set of heuristic rules.

2.2. Heuristic matching

We crawl approximately 265 000 professional reviews for around 72 000 games and 26 000 Wikipedia gameplay sections. Since there is no automatic mapping between a game to its Wikipedia page, we design some heuristics. The heuristics are the followings and applied in this order:

1. Exact title match: titles must match exactly;
2. Removing tags: when a game has the same name than its franchise, its Wikipedia page has a title similar to *Game (year video game)* or *Game (video game)*;
3. Extension match: sometimes, a sequel or an extension is not listed in Wikipedia. In this case, we map it to the Wikipedia page of the original game.

We only keep games with at least one review and a matching Wikipedia page, containing a gameplay section.

2.3. Descriptive Statistics

We build GameWikiSum corpus by considering English reviews and Wikipedia pages. Table 1 describes its overall properties. Most samples contain several reviews, whose cumulative size is too large for extractive or abstractive models to be trained in an end-to-end manner. The total vocabulary is composed of 282 992 words. Our dataset also comes from a diverse set of sources: over 480 video game websites appear as source documents in at least 6 video games; they are responsible for 99.95% of the reviews.

Percentile	**20**	**40**	**50**	**60**	**80**	**100**
Num Documents	2	5	7	10	18	84
Summary Size	139	246	321	419	684	4639
Documents Size	2 536	5 604	7 815	10 634	20 498	249 062
ROUGE-1 recall	67.7	80.7	85.29	88.8	94.1	100.0
ROUGE-2 recall	14.3	23.0	27.4	31.9	41.9	100.0

Table 1: Percentiles for different aspects of GameWikiSum. Size is in number of words. ROUGE scores are computed with a summary given its reviews.

Following Liu et al. (2018), a subset of the input has to be therefore first coarsely selected, using extractive summarization, before training an extractive or abstractive model that generates the Wikipedia gameplay text while conditioning on this extraction. Additionally, half of the summaries contain more than three hundred words (see Table 1), which is larger than previous work.
To validate our hypothesis that professional game reviews focus heavily on gameplay mechanics, we compute the proportion of unigrams and bigrams of the output given the input. We observe a significant overlap (20% documents containing 67.7% of the words mentioned in the summary, and at least 27.4% bigrams in half of the documents), emphasizing the extractive nature of GameWikiSum. Several examples of summaries are shown in Section 3.4.
Table 2 shows a comparison between GameWikiSum and other single and multi-document summarization datasets. GameWikiSum has larger input and output size than single document summarization corpora (used in extractive and abstractive models) while sharing similar word overlap ratios. Compared to DUC and TAC (news domain), GameWikiSum is also domain-specific and has two orders of magnitude more examples, facilitating the use of more powerful models. Finally, WikiSum has more samples but is more suitable for general abstractive summarization, as its articles cover a wide range of areas and have a lower word overlap ratio.
We divide GameWikiSum into train, validation and testing sets with a rough ratio of 80/10/10, resulting in 11 744, 1 454 and 1 454 examples respectively. If a game has been released on several platforms (represented by different samples), we group them in the same subset to avoid review overlap between training, validation, and testing. The distribution of samples per platform is shown in Table 3. We compute in addition the mean number of input documents, ROUGE-1, and ROUGE-2 scores of the output given the input. We observe that most platforms have a mean ROUGE-1 score above 80 and 30 for ROUGE-2.

3. Experiments and Results

3.1. Evaluation Metric

We use the standard ROUGE (Lin, 2004) used in summarization and report the ROUGE-L F1 score. ROUGE-L F1 is more appropriate to measure the quality of generated

[2] https://www.metacritic.com

[3] For example www.playstationlifestyle.net has only reviews for *Playstation* games.

[4] https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style

[5] https://www-nlpir.nist.gov/projects/duc/guidelines.html

[6] http://www.nist.gov/tac/

Dataset	Input	Output	# Examples	ROUGE-1 R
*Gigaword (Graff and Cieri, 2003)	10^1	10^1	10^6	78.7
*CNN/DailyMail (Nallapati et al., 2016)	10^2-10^3	10^1	10^5	76.1
DUC 2001-2004[5]	10^3	10^2	10^2	94.4
TAC 2008-2011[6]	10^3	10^2	10^2	95.3
WikiSum (Liu et al., 2018)	10^2-10^6	10^1-10^3	10^6	59.2
GameWikiSum (ours)	10^3-10^5	10^2-10^3	10^4	80.1

Table 2: Sizes and unigram recall of single (marked with *) and multi-document summarization datasets. Recall is computed with reference summaries given the input documents.

Platform	# Games	# Documents	ROUGE-1 R	ROUGE-2 R
PC	3586	8 ± 8	81.18 ± 15.45	27.32 ± 14.52
Wii U	224	10 ± 13	86.47 ± 10.78	34.14 ± 16.03
Nintendo 64	66	8 ± 3	77.46 ± 13.10	21.11 ± 9.37
Dreamcast	83	6 ± 2	66.12 ± 13.73	13.01 ± 6.27
PlayStation	86	4 ± 2	60.95 ± 14.67	10.97 ± 6.47
PlayStation 2	954	13 ± 9	85.93 ± 11.74	30.47 ± 11.89
Game Boy Advance	368	5 ± 4	69.38 ± 17.78	17.23 ± 11.15
GameCube	341	10 ± 7	82.26 ± 12.16	24.95 ± 10.66
Xbox	486	15 ± 9	88.40 ± 9.95	32.31 ± 10.79
DS	679	10 ± 9	85.27 ± 11.77	30.99 ± 13.38
PSP	407	12 ± 9	85.08 ± 13.85	30.71 ± 13.27
Xbox 360	1358	19 ± 14	86.90 ± 14.54	34.93 ± 15.72
PlayStation 3	1128	13 ± 11	84.53 ± 16.27	32.28 ± 15.48
Wii	665	10 ± 10	84.70 ± 14.07	32.18 ± 14.77
iOS	1344	4 ± 3	77.86 ± 15.48	23.39 ± 13.26
Xbox One	817	8 ± 9	83.33 ± 14.53	30.66 ± 15.63
3DS	312	15 ± 14	88.62 ± 12.87	39.75 ± 19.01
PlayStation Vita	337	7 ± 9	80.97 ± 14.50	28.21 ± 16.63
PlayStation 4	1103	14 ± 14	87.42 ± 14.02	37.84 ± 18.00
Switch	308	11 ± 12	89.97 ± 9.64	38.61 ± 15.95
All	14652	11 ± 11	83.19 ± 15.04	29.99 ± 15.48

Table 3: Game distribution over platforms with their average and standard deviation number of input documents and ROUGE scores.

summaries in this context because summary lengths are longer than usual (see Table 2) and vary across the dataset (see Table 1). Another motivation to use ROUGE-L F1 is to compare abstractive models with extractive ones, as the output length is unknown a priori for the former, but not for the latter. We report in addition ROUGE-1 and ROUGE-2 recall scores.

To ensure consistent results across all comparative experiments, extractive models generate summaries of the same length as reference summaries. In realistic scenarios, summary lengths are not pre-defined and can be adjusted to produce different types of summaries (e.g., short, medium or long). We do not explicitly constrain the output length for abstractive models, as each summary is auto-regressively generated.

3.2. Baselines

For extractive models, we include LEAD-k which is a strong baseline for single document summarization tasks and takes the first k sentences in the document as summary (See et al., 2017). TextRank (Mihalcea and Tarau, 2004) and LexRank (Erkan and Radev, 2004) are two graph-based methods, where nodes are text units and edges are defined by a similarity measure. SumBasic (Nenkova and Vanderwende, 2005) is a frequency-based sentence selection method, which uses a component to re-weigh the word probabilities in order to minimize redundancy. The last extractive baselines are the near state-of-the-art models C_SKIP from Rossiello et al. (2017) and SemSenSum from Antognini and Faltings (2019). The former exploits the capability of word embeddings to leverage semantics, whereas the latter aggregates two types of sentence embeddings using a sentence semantic relation graph, followed by a graph convolution.

We use common abstractive sequence-to-sequence baselines such as Conv2Conv (Gehring et al., 2017), Transformer (Vaswani et al., 2017) and its language model variant, TransformerLM (Liu et al., 2018). We use implementations from *fairseq*[7] and *tensor2tensor*[8]. As the corpus size is too large to train extractive and abstractive models in an

[7] github.com/pytorch/fairseq

[8] github.com/tensorflow/tensor2tensor

end-to-end manner due to hardware constraints, we use Tf-Idf to coarsely select sentences before training similarly to Liu et al. (2018). We limit the input size to 2K tokens so that all models can be trained on a Titan Xp GPU (12GB GPU RAM). We run all models with their best reported parameters.

3.3. Results

Model	R-L	R-1	R-2
LEAD-3	11.45	12.77	2.45
LEAD-5	18.78	19.82	3.42
TextRank	29.30	31.07	4.96
LexRank	29.74	31.26	4.96
SumBasic	30.36	31.82	4.79
C_SKIP	31.66	32.90	5.25
SemSenSum	31.72	35.11	5.56
Conv2Conv*	20.10	19.30	5.20
Transformer*	14.60	16.00	2.80
TransformerLM*	9.52	7.03	1.17

Table 4: Comparison extractive and abstractive (marked with *) models. Reported scores correspond to ROUGE-L F1 score, ROUGE-1 and ROUGE-2 recall respectively.

Table 4 contains the results. LEAD-5 achieves less than 20 for ROUGE-L as well as ROUGE-1 and less than 3.5 for ROUGE-2. Taking only 3 sentences leads to even worse results: below 13 and 3 respectively. Unlike in other datasets, these results are significantly outperformed by all other extractive models but surprisingly, abstractive models perform worse on average. This demonstrates the difficulty of the task in GameWikiSum compared to Nallapati et al. (2016) and Graff and Cieri (2003).

For extractive models, TextRank and LexRank perform worse than other models. The frequency-based model SumBasic performs slightly better but does not achieve comparable results with embedding-based models. Best results are obtained with C_SKIP and SemSentSum, showing that more sophisticated models can be trained on GameWikiSum and improve results significantly. Interestingly, taking into account the context of a sentence and hence better capturing the semantics, SemSentSum achieves only slightly better scores than C_SKIP, which relies solely on word embedding. We show in Section 3.4. several examples with their original summaries and generated ones with the best model.

Overall, the abstractive performance of sequence-to-sequence and language models are significantly lower than C_SKIP and SemSentSum in terms of ROUGE-L and ROUGE-1. However, Conv2Conv obtains only 0.05 less ROUGE-2 score compared to C_SKIP and 0.36 to SemSentSum. We suspect ROUGE-2 to be easier for abstractive sequence-to-sequence models, as half of the samples only have a ROUGE-2 around 27.00 without any limitation of the input size (see Table 1). Consequently, copying sentences from a small subset of the whole input documents for extractive models leads to worse ROUGE-2 recall. A normal transformer underperforms compared to Conv2Conv, and its language model variant achieves significantly worse results than other models due to a lack of data. We highlight that GameWikiSum has two orders of magnitude fewer samples (see Table 2) compared to Liu et al. (2018). Therefore, it is necessary to have either additional annotated data or pre-train TransformerLM on another corpus.

3.4. Examples of Original and Generated Summaries

Figure 1 shows two samples with their gameplay sections from Wikipedia and summaries generated by the best baseline SemSentSum. In the first example, we notice that the model has selected sentences from the reviews that are also in the original Wikipedia page. Additionally, we observe, for both examples, that several text fragments describe the same content with different sentences. Consequently, this supports our hypothesis that professional reviews can be used in a multi-document summarization setting to produce summaries reflecting the gameplay section of Wikipedia pages.

4. Related Work

To the best of our knowledge, DUC and TAC are the first multi-document summarization datasets. They contain documents about the same event and human-written summaries. Unsurprisingly, this approach does not scale and they could only collect hundreds of samples as shown in Table 2.

Zopf et al. (2016) applied a similar strategy using Wikipedia, where they asked annotators to first tag and extract information nuggets from the lead section of Wikipedia articles. In a further step, the same annotators searched for source documents using web search engines. As the whole process depends on humans, they could only collect around one thousand samples. Other attempts such as (Cao et al., 2016) have been made using Twitter, but the resulting dataset size was even smaller.

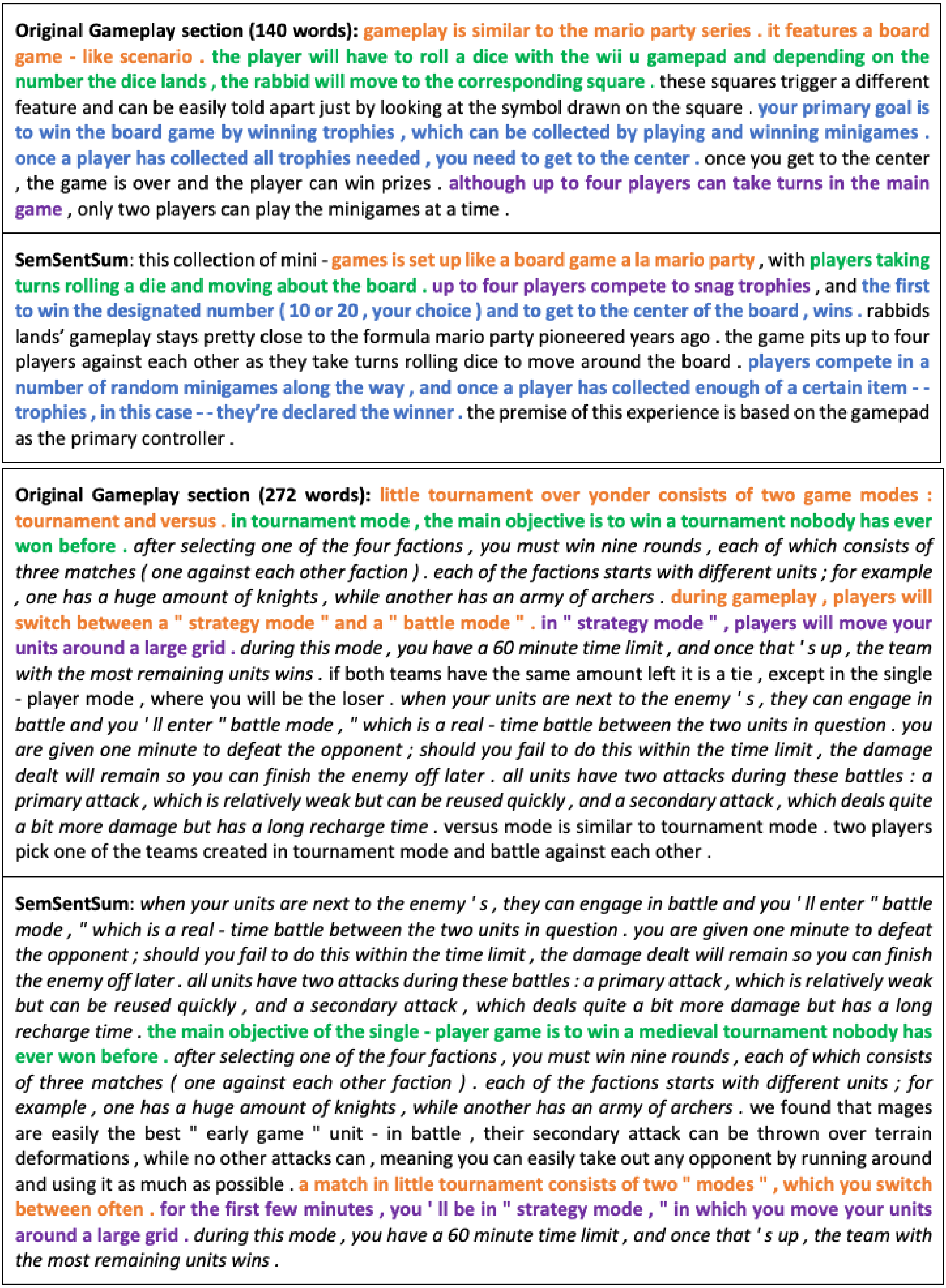
Only the recent work of Liu et al. (2018) addresses the automatic creation of a large-scale multi-document summarization corpus, WikiSum. Summaries are lead sections of Wikipedia pages and input documents a mixture of 1) its citations from the reference section 2) results from search engines using the title of the Wikipedia page as the query. However, references (provided by contributors) are needed for their model to generate lead sections which are not garbaged texts, as shown in the experiments (Liu et al., 2018). Consequently, this dataset is unusable for real use-cases. Similarly, Zopf (2018) propose a multilingual Multi-Document dataset of approximately 7 000 examples based on English and German Wikipedia articles. We, however, are focused on the video game domain and provide twice more samples.

5. Conclusion

In this work, we introduce a new multi-document summarization dataset, GameWikiSum, based on professional

[11] https://en.wikipedia.org/wiki/Rabbids_Land

[12] https://en.wikipedia.org/wiki/Little_Tournament_Over_Yonder



Original Gameplay section (140 words): gameplay is similar to the mario party series . it features a board game - like scenario . the player will have to roll a dice with the wii u gamepad and depending on the number the dice lands , the rabbid will move to the corresponding square . these squares trigger a different feature and can be easily told apart just by looking at the symbol drawn on the square . **your primary goal is to win the board game by winning trophies , which can be collected by playing and winning minigames . once a player has collected all trophies needed , you need to get to the center .** once you get to the center , the game is over and the player can win prizes . **although up to four players can take turns in the main game** , only two players can play the minigames at a time .

SemSentSum: this collection of mini - **games is set up like a board game a la mario party** , with **players taking turns rolling a die and moving about the board . up to four players compete to snag trophies** , and **the first to win the designated number (10 or 20 , your choice) and to get to the center of the board , wins .** rabbids lands' gameplay stays pretty close to the formula mario party pioneered years ago . the game pits up to four players against each other as they take turns rolling dice to move around the board . **players compete in a number of random minigames along the way , and once a player has collected enough of a certain item - - trophies , in this case - - they're declared the winner .** the premise of this experience is based on the gamepad as the primary controller .

Original Gameplay section (272 words): little tournament over yonder consists of two game modes : tournament and versus . in tournament mode , the main objective is to win a tournament nobody has ever won before . *after selecting one of the four factions , you must win nine rounds , each of which consists of three matches (one against each other faction) . each of the factions starts with different units ; for example , one has a huge amount of knights , while another has an army of archers .* **during gameplay , players will switch between a " strategy mode " and a " battle mode " . in " strategy mode " , players will move your units around a large grid .** *during this mode , you have a 60 minute time limit , and once that ' s up , the team with the most remaining units wins .* if both teams have the same amount left it is a tie , except in the single - player mode , where you will be the loser . *when your units are next to the enemy ' s , they can engage in battle and you ' ll enter " battle mode , " which is a real - time battle between the two units in question . you are given one minute to defeat the opponent ; should you fail to do this within the time limit , the damage dealt will remain so you can finish the enemy off later . all units have two attacks during these battles : a primary attack , which is relatively weak but can be reused quickly , and a secondary attack , which deals quite a bit more damage but has a long recharge time .* versus mode is similar to tournament mode . two players pick one of the teams created in tournament mode and battle against each other .

SemSentSum: *when your units are next to the enemy ' s , they can engage in battle and you ' ll enter " battle mode , " which is a real - time battle between the two units in question . you are given one minute to defeat the opponent ; should you fail to do this within the time limit , the damage dealt will remain so you can finish the enemy off later . all units have two attacks during these battles : a primary attack , which is relatively weak but can be reused quickly , and a secondary attack , which deals quite a bit more damage but has a long recharge time .* **the main objective of the single - player game is to win a medieval tournament nobody has ever won before .** *after selecting one of the four factions , you must win nine rounds , each of which consists of three matches (one against each other faction) . each of the factions starts with different units ; for example , one has a huge amount of knights , while another has an army of archers .* we found that mages are easily the best " early game " unit - in battle , their secondary attack can be thrown over terrain deformations , while no other attacks can , meaning you can easily take out any opponent by running around and using it as much as possible . **a match in little tournament consists of two " modes " , which you switch between often . for the first few minutes , you ' ll be in " strategy mode , " in which you move your units around a large grid .** *during this mode , you have a 60 minute time limit , and once that ' s up , the team with the most remaining units wins .*

Figure 1: Reference summary and a summary produced by SemSentSum for the game *Rabbids Land*[11] and *Little Tournament Over Yonder*[12]. Sentences in italic correspond to identical text fragments and colored ones describe the same content.

video game reviews, which is one hundred times larger than commonly used datasets. We conclude that the size of GameWikiSum and its domain-specificity makes the training of abstractive and extractive models possible. In future work, we could increase the dataset with other languages and use it for multilingual multi-document summarization. We release GameWikiSum for further research: `https://github.com/Diego999/GameWikiSum`.

6. Bibliographical References

Antognini, D. and Faltings, B. (2019). Learning to create sentence semantic relation graphs for multi-document summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 32–41, Hong Kong, China, November. Association for Computational Linguistics.

Cao, Z., Chen, C., Li, W., Li, S., Wei, F., and Zhou, M. (2016). Tgsum: Build tweet guided multi-document summarization dataset. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Erkan, G. and Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org.

Graff, D. and Cieri, C. (2003). English gigaword 2003. In *Linguistic Data Consortium, Philadelphia*.

Lin, C. Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS)*, Barcelona, Spain, July 25-26.

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. (2018). Generating wikipedia by summarizing long sequences. *International Conference on Learning Representations*.

Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing order into texts. In *Proceedings of EMNLP-04and the 2004 Conference on Empirical Methods in Natural Language Processing*, July.

Nallapati, R., Zhou, B., dos Santos, C., glar Gulçehre, Ç., and Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. *CoNLL 2016*, page 280.

Nenkova, A. and Vanderwende, L. (2005). The impact of frequency on summarization. page 101, Microsoft Research, Redmond, Washington, Tech. Rep.

Rossiello, G., Basile, P., and Semeraro, G. (2017). Centroid-based text summarization through compositionality of word embeddings. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, pages 12–21.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1073–1083.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Zhou, W. and Duan, W. (2010). Online user reviews and professional reviews: A bayesian approach to model mediation and moderation effects. In *ICIS 2010*.

Zopf, M., Peyrard, M., and Eckle-Kohler, J. (2016). The next step for multi-document summarization: A heterogeneous multi-genre corpus built with a novel construction approach. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1535–1545.

Zopf, M. (2018). Auto-hMDS: Automatic construction of a large heterogeneous multilingual multi-document summarization corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).