AUEB-Archimedes at RIRAG-2025: Is obligation concatenation really all you need?

Ioannis Chasandras [1], **Odysseas S. Chlapanis**[1,2] **and Ion Androutsopoulos**[1,2]
[1]Department of Informatics, Athens University of Economics and Business, Greece
[2]Archimedes/Athena RC, Greece

Abstract

This paper presents the systems we developed for RIRAG-2025, a shared task that requires answering regulatory questions by retrieving relevant passages. The generated answers are evaluated using RePASs, a reference-free and model-based metric. Our systems use a combination of three retrieval models and a reranker. We show that by exploiting a neural component of RePASs that extracts important sentences ('obligations') from the retrieved passages, we achieve a dubiously high score (0.947), even though the answers are directly extracted from the retrieved passages and are not actually generated answers. We then show that by selecting the answer with the best RePASs among a few generated alternatives and then iteratively refining this answer by reducing contradictions and covering more obligations, we can generate readable, coherent answers that achieve a more plausible and relatively high score (0.639).

1 Introduction

The Regulatory Information Retrieval and Answer Generation (RIRAG)[1] shared task focuses on the development of systems that can effectively retrieve relevant information from regulatory texts to generate accurate answers for obligation-related queries. It is divided into two subtasks: *passage retrieval*, where systems identify the ten most relevant passages from regulatory documents, and *answer generation*, which requires synthesizing comprehensive answers from the retrieved passages.

We participated with three systems and released our code publicly.[2] Each one of them uses a Rank Fusion (Wang et al., 2021) combination of three retrieval models: BM25 (Robertson et al., 1994), and two neural domain-specific retrievers, based on a law- and a finance-specific embedding model, respectively. We also apply a neural reranker to the top-N retrieved passages.

For answer generation, our first system adversarially exploits the evaluation metric of the task, called RePASs, by using one of its neural components. Specifically, we extract important sentences ('obligations') from the retrieved passages and then concatenate these sentences to get an 'answer'. Even though the produced answers may be incoherent and may not answer the question directly, this system achieves a perfect score, much higher than the score of human experts. The second system extends this approach with an LLM that generates an answer (for each question) by iteratively reformulating (as parts of an answer) the extracted obligations of the previous system. This results in more readable answers, but performance deteriorates to RePASs scores below those of the challenge's baseline (Gokhan et al., 2024).

Our third system works by a) generating multiple candidate answers and using RePASs to select the best answer, and b) iteratively refining the selected answer by removing contradictions and adding 'obligation' sentences that increase RePASs. This system performs worse than the adversarial (first) system, but much better than the baseline, and the answers are coherent and readable.

2 Task setup

Dataset: The dataset of the task consists of train, development, and test sets (22k, 2.8k, 2.7k questions respectively). Passages are retrieved from a corpus of 40 regulatory documents from the Abu Dhabi Global Markets (ADGM) collection. The task organizers used a separate hidden test set, with 446 questions, to evaluate the participants.

Evaluation: Passage retrieval is evaluated using recall@10 and MAP@10. Answer generation is evaluated using RePASs, a reference-free metric (Gokhan et al., 2024). To calculate RePASs, *en-*

[1] https://regnlp.github.io/
[2] https://github.com/nlpaueb/verify-refine-repass

Proceedings of the 31st International Conference on Computational Linguistics, pages 52–58
January 19–24, 2025. ©COLING 2025

tailment and *contradiction* scores are obtained by comparing each sentence of the retrieved passages (used as premises) with each sentence of the generated answer (hypothesis) using an NLI model. For each generated sentence (of the answer), the highest probabilities for entailment and contradiction (comparing to retrieved sentences) are selected, and the scores are averaged over all the sentences of the answer. Additionally, *obligation*-sentences are extracted from the retrieved passages using a LegalBERT model (Chalkidis et al., 2020) fine-tuned on a synthetic dataset (Gokhan et al., 2024). For an obligation to be considered *covered* by the generated answer, a sentence of the answer must entail the obligation-sentence with a confidence above a certain threshold, according to another NLI model.

3 Passage retrieval

All three of our systems use the same passage retrieval, which improves upon the baseline retrieval system of the shared task (Gokhan et al., 2024) in three ways: a) we use domain-specific neural retrieval models, b) we extend the Rank Fusion approach (Wang et al., 2021) to include three models instead of two, and c) we use a reranker.

3.1 Retrieval models

We experiment with BM25 (Robertson et al., 1994) and three of the best[3] text embedding models: `text-embedding-3-large` (OL3) from `OpenAI` (Neelakantan et al., 2022), `voyage-law-2` (VL2), and `voyage-finance-2` (VF2) from `Voyage`.[4] The OL3 embedding model is only used for comparison; it is not included in our final systems, because domain-specific embedding models worked better. We also use the `voyage-rerank-2` reranker.

3.2 Rank Fusion

The task combines the financial and legal domains, which motivates using two domain-specific neural retrievers. Also, according to Wang et al. (2021), BM25 should be fused with neural retrievers, because it captures exact term matching better. Hence, we expand Rank Fusion to handle three retrievers instead of two, as follows.

$$f(p) = a\hat{s}_x(p) + b\hat{s}_y(p) + (1-(a+b))\hat{s}_z(p) \quad (1)$$

[3]MTEB-law: `https://huggingface.co/spaces/mteb/leaderboard?task=retrieval&language=law`

[4]`https://docs.voyageai.com/docs/embeddings`

Here p is a retrieved passage, a and b are fusion weights, and $\hat{s}_x(p)$, $\hat{s}_y(p)$, $\hat{s}_z(p)$ are the normalized relevance scores of the three fused retrievers.

3.3 Experimental results for retrieval

We conduct three experiments on the public test set. In Table 1, we compare the scores of the four single retrieval models. We see that the domain-specific voyage-law-2 (VL2) and voyage-finance-2 (VF2) perform better than BM25 and the generic OL3.

Model	Recall@10	MAP@10
BM25	0.6994	0.5584
OL3	0.7385	0.5736
VL2	0.7705	0.6275
VF2	**0.7895**	**0.6559**

Table 1: Comparison of single retrieval models.

In the second experiment (Table 2), we compare Rank Fusion configurations, again on the public test set. The newly introduced triple Rank Fusion, with BM25, VL2 and VF2, is the best. The values of a, b were selected by trying a few combinations.

Rank Fusion	a	b	R@10	M@10
BM25, OL3	0.30	-	78.9	65.0
VL2, VF2	0.40	-	79.4	66.0
BM25, VL2	0.25	-	79.9	66.5
BM25, VF2	0.30	-	80.4	67.6
BM25, VL2, VF2	0.25	0.2	**81.1**	**69.0**

Table 2: Comparison of Rank Fusion configurations.

In the third experiment (Fig. 1), we investigate the effect of reranking the top-N retrieved passages, for different N values, by computing Recall@10 on the public test set. The best value is $N = 50$.

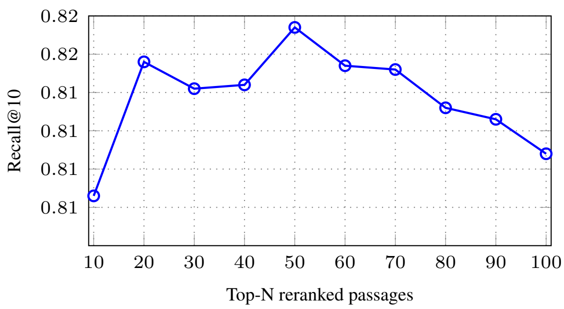

Figure 1: Recall@10 scores of our best retriever (Rank Fusion of BM25, VL2, VF2) when reranking the top-N retrieved passages, for different N values.

Our final retrieval model is a triple Rank Fusion model (BM25, VL2, VF2) with reranking (`voyage-rerank-2`, $N = 50$), which ranked 4th in the retrieval subtask, achieving 69.4 Recall@10, and 59.4 MAP@10 on the hidden test set.

4 Answer generation

The answer generators of this section use our best retriever (Section 3, BM25, VL2, VF2, reranker).

4.1 Preprocessing

Filtering: We follow Gokhan et al. (2024), i.e., we rank the retrieved passages by decreasing relevance scores; we then keep only passages that satisfy two conditions: (i) their score must be above a certain *threshold*, and (ii) their score must not fall below the previous passage's score more than *max drop*.

Extracting obligations: To obtain obligations from the retrieved passages, we use the same fine-tuned LegalBERT model used in RePASs (Section 2) for obligation extraction. If a passage does not contain any obligations, we use it as is.

4.2 Experimental results for preprocessing

To select the values of the filtering *threshold* and *max drop* (Section 4.1), we conducted two experiments using `GPT-4o-mini`[5] for answer generation. The first experiment shows that the recommended values of $0.70, 0.20$ of Gokhan et al. (2024) are outperformed by $0.90, 0.10$, respectively (Table 3).

Threshold	Max Drop	RePASs
0.70	0.20	0.4708
0.75	0.05	0.5006
0.80	0.05	0.5050
0.85	0.15	0.5001
0.90	**0.10**	**0.5117**

Table 3: Performance of the baseline answer generator for different values of *threshold* and *max drop*, using our best retriever (BM25, VL2, VF2, reranker).

The second experiment compared the performance of the task's baseline when (a) the entire retrieved passages were given to the LLM, or (b) only the obligations were given, or (c) only the obligations were given, but with a tailored prompt. No significant difference was noticed between (a) and (b), but (c) was significantly better in RePASs (Table 4), due to the increase in *obligation coverage* and *entailment*, even though *contradiction* was worse. All prompts can be found in Appendix B.

[5] https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

Context	RePASs	Obl.	Ent.	Con.
Passages	0.411	0.147	0.177	**0.090**
Obligations	0.413	0.156	0.172	**0.090**
+ prompt	**0.512**	**0.278**	**0.366**	0.109

Table 4: Performance of the baseline system for different kinds of inputs (entire retrieved passages, obligations only, obligations with tailored prompt).

4.3 Naive Obligation Concatenation (NOC)

Our first answer generator (NOC) adversarially exploits the extracted obligations (Section 4.1). It simply concatenates and outputs them as the 'answer'. From the definition of RePASs (Section 2), this answer should get an almost perfect obligation score. Additionally, we expect a low contradiction score, as obligations should not conflict.

4.4 LLM Obligation Concatenation (LOC)

The answers of NOC (Section 4.3) do not answer the question directly; they are just excerpts from retrieved passages. To alleviate this, we create a variation of NOC, called LOC: for each extracted obligation, we prompt an LLM (`GPT-4o-mini`) to answer the given question using this obligation. If the generated answer does not *cover* (Section 2) the original obligation, then the LLM is prompted again, until a certain number of tries K has been reached (we use $K = 3$). Finally, the per-obligation answers are concatenated to form a complete answer.

4.5 Verify and Refine with RePASs (VRR)

Our third answer generator (VRR) first 'verifies' the correctness of the answers, then iteratively 'refines' them. The first stage (verification) is loosely inspired by self-consistency (Wang et al., 2023); it involves the generation of many alternative answers by the LLM and the selection of the one with the highest RePASs score. The selected answer is then iteratively refined by reducing *contradictions* and increasing *obligations*, as explained below.

4.5.1 Verification step

In the verification step, we obtain N alternative answers from the LLM (using all the extracted obligations and the question as input) and evaluate them using RePASs. We choose the alternative answer with the best RePASs score.

4.5.2 Refinement step

Contradiction removal: To remove contradictions: a) we compute the average contradiction

System / Group Name	RePASs	Obligation	Entailment	Contradiction
GPT-4o baseline*	0.583	0.220	0.769	0.238
Human experts*	0.859	**1.000**	0.837	0.260
Indic aiDias	**0.973**	0.993	**0.987**	0.062
Ocean's Eleven	0.971	0.991	0.986	0.065
AUEB NLP Group - NOC	0.947 (0.951)	0.951 (0.963)	0.986 (0.986)	0.096 (0.096)
AUEB NLP Group - VRR	0.639 (0.646)	0.502 (0.524)	0.446 (0.446)	**0.031** (0.031)
AICOE	0.601	0.230	0.827	0.254
AUEB NLP Group - LOC	0.562 (0.568)	0.423 (0.439)	0.375 (0.375)	0.110 (0.110)

Table 5: Leaderboard results for Subtask 2. Results computed by ourselves for our systems are shown in brackets. Differences are attributed to using different GPUs. *Scores taken from Gokhan et al. (2024).

score over all the answers (over all the best alternative answers for all questions) across the dataset using the same NLI model as in RePASs, and b) we remove the sentences of the answer that get a contradiction score higher than the average.

Obligation insertion: To locate missing obligations, we extract obligations from the retrieved passages and the current answer. Obligations from the retrieved passages that are not *covered* (Section 2) by the current answer are *missing* obligations. We prompt `GPT-4o` to insert the missing obligations by correcting a sentence or adding a new one to the current answer (complete prompt in Appendix B).

4.6 Experimental results for generation

In the following experiments we use the hidden test set, `GPT-4o-mini` as the generator for LOC, and `GPT-4o`[6] as the generator for VRR.

Table 5 compares the task's baseline and human expert performance, as reported by Gokhan et al. (2024), to our three submissions (NOC, VRR, LOC) and to the best submissions of the top three competitors. NOC achieves an almost perfect RePASs score (0.947), surpassing human experts (+0.088). As expected, *obligation* and *contradiction* scores are excellent for the adversarial NOC, but surprisingly *entailment* scores are even better without directly optimizing towards them. Similar results are observed for the methods of the top scoring competitors. However, as already mentioned, NOC's answers are just verbatim sentences from the retrieved passages, which proves that RePASs can easily be deceived. LOC on the other hand, which rewrites the 'obligations' using `GPT-4o-mini`, performs even worse than the baseline model, which shows that RePASs is also very sensitive to the style of the answer. VRR, which actually generates answers from the retrieved passages, improves upon the task's baseline substantially (+0.056) and ranks first among systems that do not exceed human performance; we suspect that systems with super-human performance may trick the RePASs measure, like our NOC system.

VRR	RePASs	Improvement
Baseline (Ours)	0.506	-
+ Verification	0.611	**+ 0.105**
+ Refinement	**0.646**	+ 0.025

Table 6: Contribution of VRR stages, using GPT-4o.

The next experiment (Table 6) measures the contribution of the verification and refinement processes of VRR. Both processes are beneficial, but verification's improvement is more important.

5 Conclusion

We introduced three systems for the RIRAG shared task. The retrieval backbone of all systems combined BM25 with two domain specific neural retrievers and a reranker. We achieved a near-perfect score with an adversarial system that exploits the neural model for *obligation* extraction of RePASs, highlighting the difficulty of developing a robust reference-free metric for RAG evaluation. Our best non-adversarial system (VRR) first generates multiple alternative answers from the retrieved obligations, selects the alternative answer that maximizes RePASs, then iteratively improves it by maximizing obligation coverage and minimizing contradictions. This system produces coherent answers, and obtains the highest RePASs score among competitors that do not exceed human performance (which may be a sign of gaming RePASs).

[6] https://openai.com/index/hello-gpt-4o/

Limitations

We demonstrated that reference-free model-based metrics, such as RePASs, used for evaluating Retrieval-Augmented Generation (RAG) systems, can be susceptible to adversarial attacks. Specifically, we showed that it is possible to provide answers that receive a high score from the metric, but may not be useful to non-experts. The attack was tailored to RePASs and a specific domain, and it may not apply to other domains or metrics.

VRR requires an accurate verifier, such as RePASs, which is not always available. The *obligation extraction* component in RePASs is fine-tuned using a synthetic dataset (Gokhan et al., 2024), which in turn requires a powerful LLM teacher to solve the task with few-shot prompting alone. This is quite rare for hard domain-specific problems.

Acknowledgments

This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program. All experiments were done using AWS resources which were provided by the National Infrastructures for Research and Technology GRNET and funded by the EU Recovery and Resiliency Facility.

References

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Catalina Goanta, Nikolaos Aletras, Ilias Chalkidis, Sofia Ranchordás, and Gerasimos Spanakis. 2023. Regulation and NLP (RegNLP): Taming large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8712–8724, Singapore. Association for Computational Linguistics.

Tuba Gokhan, Kexin Wang, Iryna Gurevych, and Ted Briscoe. 2024. Regnlp in action: Facilitating compliance through automated information retrieval and answer generation. *Preprint*, arXiv:2409.05677.

Yichen Huang and Timothy Baldwin. 2023. Robustness tests for automatic machine translation metrics with adversarial attacks. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5126–5135, Singapore. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, pages 9459–9474, Red Hook, NY, USA. Curran Associates Inc.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2024. Self-refine: iterative refinement with self-feedback. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 46534–46594, Red Hook, NY, USA.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. 2022. Text and code embeddings by contrastive pre-training. *Preprint*, arXiv:2201.10005.

Xin Quan, Marco Valentino, Louise A. Dennis, and Andre Freitas. 2024. Verification and refinement of natural language explanations through LLM-symbolic theorem proving. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2933–2958, Miami, Florida, USA. Association for Computational Linguistics.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at trec-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994*, pages 109–126, Gaithersburg, Maryland, USA.

Ante Wang, Linfeng Song, Ye Tian, Baolin Peng, Lifeng Jin, Haitao Mi, Jinsong Su, and Dong Yu. 2024. Self-consistency boosts calibration for math reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6023–6029, Miami, Florida, USA. Association for Computational Linguistics.

Shuai Wang, Shengyao Zhuang, and Guido Zuccon. 2021. Bert-based dense retrievers require interpolation with bm25 for effective passage retrieval. In

Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR '21, page 317–324, New York, NY, USA. Association for Computing Machinery.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: deliberate problem solving with large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, pages 11809–11822, Red Hook, NY, USA. Curran Associates Inc.

A Related work

RAG: Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) systems can help tackle domain-specific problems that RegNLP (Goanta et al., 2023) presents, by incorporating information from large regulatory document collections.

Verify and Refine: VRR is loosely inspired by LLM methods that select the best answer from multiple candidates and iteratively refine these answers (Wang et al., 2024; Madaan et al., 2024; Yao et al., 2024; Quan et al., 2024), frameworks like Explanation-Refiner (Quan et al., 2024) that use theorem proving to validate and refine explanations, and WizardLM (Xu et al., 2024) that evolves instruction data to enhance model performance.

Adversarial attacks: Many works implement adversarial attacks that are similar to our NOC system. BERT-ATTACK (Li et al., 2020) leverages a pretrained BERT model to deceive other models. Huang and Baldwin (2023) show that popular model-based evaluation metrics for machine-translation are susceptible to inconsistencies when given adversarially-degraded translations.

B Prompts

For all our prompts we have used `GPT-4o` to improve them, and then kept those that performed the task better (according to our opinion) in a few (2-3) sample questions.

Baseline prompt (Gokhan et al., 2024)

You are a regulatory compliance assistant. Provide a detailed answer for the question that fully integrates all the obligations and best practices from the given passages. Ensure your response is cohesive and directly addresses the question. Synthesize the information from all passages into a single, unified answer.

Prompt for obligations in the context (VRR)

You are a regulatory compliance assistant. Your task is to provide a brief but concise and detailed answer to the Question, ensuring that all Obligations are fully addressed. Directly integrate each obligation into the response, ensuring no obligation is missed or implied. Avoid adding information beyond what is explicitly stated in the Obligations, and cite specific rules when necessary. Use the exact terminology and structure from the obligations where applicable, to ensure high alignment and logical consistency. Focus solely on the provided obligations to craft a response that is well-structured, concise, and free of contradictions.

Prompt for inserting obligations (VRR)

You are a regulatory compliance assistant. Your task is to integrate the following Obligations that are missing from the Answer. You may change sentences or add new ones to cover all Obligations. Avoid adding changes or sentences that contradict the Answer and/or the Obligations.

Prompt that rewrites an obligation (LOC)

You are a regulatory compliance assistant. Your task is to construct a brief but concise response that addresses the Question by focusing exclusively on the specified Obligation. Ensure your response clearly identifies and explains the obligation, including any relevant conditions or restrictions. Avoid addressing unrelated aspects of the Question, and limit your response strictly to what is explicitly stated in the provided passage.

C Detailed experiments for VRR

Table 7 shows the progression of RePASs throughout the execution of the VRR algorithm. The Verification step leads to an increase in all metrics. Obligation Refinement ('Ref. Obl.') alone does not lead to an increased score, Contradiction Refinement ('Ref. Contr.') is necessary. Even though

Obligation Coverage ('Obl.') increases at the expense of the Entailment ('Ent.') score, RePASs improves overall.

Step	**RePASs**	**Obl.**	**Ent.**	**Con.**
Preprocessing	0.506	0.246	0.408	0.136
Verify	0.611	0.389	0.527	0.083
Ref. Contr. 1	0.638	0.389	**0.554**	0.030
Ref. Obl. 1	0.634	0.465	0.490	0.053
Ref. Contr. 2	0.643	0.464	0.497	0.032
Ref. Obl. 2	0.637	0.496	0.464	0.049
Ref. Contr. 3	0.643	0.494	0.467	**0.030**
Ref. Obl. 3	0.642	0.527	0.446	0.046
Ref. Contr. 4	**0.647**	0.525	0.446	0.031
Ref. Obl. 4	0.641	0.538	0.430	0.045

Table 7: RePASs progress during VRR execution.