NLP@IIMAS-CLTL at Multilingual Counterspeech Generation: Combating Hate Speech Using Contextualized Knowledge Graph Representations and LLMs

David Salvador Preciado Márquez
Faculty of Sciences, UNAM
Mexico City, Mexico
dpreciado3@ciencias.unam.mx

Helena Gómez Adorno
IIMAS, UNAM
Mexico City, Mexico
helena.gomez@iimas.unam.mx

Ilia Markov
CLTL, Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
i.markov@vu.nl

Selene Baez Santamaria
CLTL, Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
s.baezsantamaria@vu.nl

Abstract

We present our approach for the shared task on Multilingual Counterspeech Generation (MCG) to counteract hate speech (HS) in Spanish, English, Basque, and Italian. To accomplish this, we followed two different strategies: 1) a graph-based generative model that encodes graph representations of knowledge related to hate speech, and 2) leveraging prompts for a large language model (LLM), specifically GPT-4o. We find that our graph-based approach tends to perform better in terms of traditional evaluation metrics (i.e., RougeL, BLEU, BERTScore), while the JudgeLM evaluation employed in the shared task favors the counter-narratives generated by the LLM-based approach, which was ranked second for English and third for Spanish on the leaderboard.

1 Introduction

The prevalence of hate speech (HS) has become a problem in modern social networks (Nazmine et al., 2021), and its effects on people can range from causing fear of becoming the target of physical violence (Saresma et al., 2021) to an increase in suicide rates (Hinduja and Patchin, 2007). There are several strategies for HS mitigation, including content moderation and counterspeech intervention (Donzelli, 2021). The latter involves the use of counter-narratives (CNs), which can be defined as non-negative responses that focus on alternative perspectives and fact-based arguments (Benesch, 2014). Counterspeech is considered free from normative or censoring issues (Donzelli, 2021), which makes it an appealing strategy.

Automated CN generation is of particular interest, as the negative effects of HS on human content moderators are widely acknowledged (Spence et al., 2023) and manual CN generation is not feasible at large enough scales (Schieb and Preuss, 2016). This paper addresses the Multilingual Counterspeech Generation shared task at COLING 2025[1], which focuses on automated CN generation against HS leveraging additional background knowledge (KN). This KN provides additional informative content to fight HS (e.g., the sentence "Feminism means giving women equal opportunity and fair pay at work" serves as one of the KN context sentences for the HS "Women are weak and need men to be able to achieve something in their lives."). The shared task covers four different languages: English, Spanish, Basque, and Italian. Our solution is based on the approaches presented by Baez Santamaria et al. (2024) and Doğanç and Markov (2023). We hypothesize that the graph-based approach introduced in the former can be adapted from the dialogue-based domain to employ the background knowledge (KN) provided with the dataset, while the prompt personalization approach introduced in the latter would add relevant context to the LLM used in our experiments.

In this paper, we first provide an overview of the related work. Then, we describe the details of the dataset, as well as the details of our system and the experiments that were performed. Finally, we discuss the obtained results and derive conclusions.

2 Related Work

The task of automating CN generation was first proposed by Qian et al. (2019), and the CONAN

[1] https://sites.google.com/view/multilang-counterspeech-gen/shared-task

Proceedings of the First Workshop on Multilingual Counterspeech Generation (MCG), pages 29–36
January 19th, 2025.

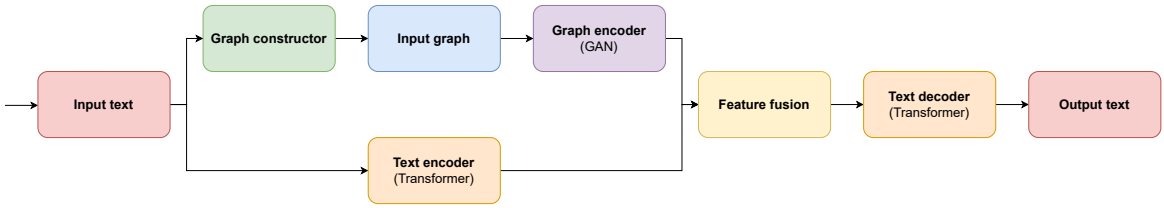

Figure 1: System architecture. From left to right: 1) The text representation is translated into a graph representation. 2) The graph and text inputs get encoded in parallel to generate vector representations. 3) The encoded representations are aligned using a feature fusion mechanism. 4) These aligned features are passed to a text decoder to generate output text.

dataset manually created by trained NGO operators was made available in (Chung et al., 2019), which enabled the training of automated generative systems. Since then, there have been approaches including both LLM-based (Vallecillo-Rodríguez et al., 2023) and graph-based (Baez Santamaria et al., 2024) strategies. We describe both the graph-based and LLM-based strategies in more detail in the following.

2.1 LLM-based approaches

Vallecillo-Rodríguez et al. (2023) used LLMs, like GPT-3 (Brown et al., 2020), in a few-shot setup. The prompt contained a few examples of HS-CN pairs and directed the model to generate a CN against a given HS. An alternative LLM-based strategy was proposed in (Doğanç and Markov, 2023), which involved a multi-step pipeline that directed the model to create personalized CNs based on the demographic characteristics of the author of HS. Both approaches rely on annotator-based evaluations that are not directly comparable, but effectively demonstrate the potential of prompt-based techniques in generating CNs using pre-trained LLMs.

2.2 Graph-based approaches

Baez Santamaria et al. (2024) presented a graph-based approach for automated CN generation. The model was trained on the DIALOCONAN dataset (Bonaldi et al., 2022), which extended HS-CN pairs with dialogue history context. The architecture proposed by Baez Santamaria et al. (2024) encodes the dialogue history into graphs with the Graph-Of-Thought (GOT) strategy (Yao et al., 2023), making use of the OpenIE framework to extract semantic triples. The constructed graph is then fed to a Graph Attention Network (GAN) (Veličković et al., 2018), the output of which is processed in a fusion layer together with an embedding of the original text, and then passed through a decoder layer to obtain the final counter-narratives. Text encodings are extracted from the *Flan-Alpaca*[2] transformer model.

3 Dataset

The ML-MTCONAN-KN dataset[3] used in the shared task consists of 596 HS-CN pairs. It covers four different languages: English, Spanish, Italian, and Basque. Each entry in the dataset contains the HS and CN, along with a description of the demographic group targeted by the HS. The dataset also includes five background knowledge (KN) sentences. The dataset statistics are provided below:

- Train: 396 pairs;
- Development: 100 pairs;
- Test: 100 pairs.

4 System Overview

Our participation involves both LLM-based and graph-based approaches. Each of our approaches uses the KN sentences provided within the ML-KN-MTCONAN dataset. We do not use additional data and rely solely on the dataset provided by the organizers. We describe the implementation details in the following sub-sections.

4.1 LLM-based approach

For this approach, we used OpenAI's API [4] to obtain chat completions from the GPT-4o model (version `gpt-4o-2024-08-06`). No further fine-tuning was performed. Two different prompting strategies were explored:

[2] https://huggingface.co/declare-lab/flan-alpaca-base

[3] https://huggingface.co/datasets/LanD-FBK/ML_MTCONAN_KN

[4] https://platform.openai.com/

ID	Coreference resolution	Data setup	Maximum nodes	Language	Triplet extraction engine
GB (1)	True	Sequential	100	English	OpenIE
GB (2)	True	Interspersed	100	English	OpenIE
GB (3)	False	Sequential	150	English	OpenIE
GB (4)	False	Interspersed	150	English	OpenIE
GB (5)	False	Interspersed	150	Multilingual	OpenIE
GB (6)	False	Interspersed	150	Multilingual	CLTL

Table 1: Configurations for our graph-based approach, GB stands for the graph-based approach.

1. The GPT-4o model was instructed to produce a CN personalized with respect to the author of the HS. In this strategy, the model was allowed a maximum output token window of 100.

2. Similarly to the previous strategy, the model was instructed to produce a CN personalized with respect to the author of the HS. In this case, a Chain-Of-Thought-inspired (Wei et al., 2023) prompt was implemented, with a 2,000 maximum output token window.

The prompts used for both strategies are provided in Appendix A. In all cases, the prompt and the HS instance were provided to the model, as well as the KN sentences present in the ML-KN-MTCONAN dataset.

4.2 Graph-based approach

Our approach follows the same general architecture as presented by Baez Santamaria et al. (2024), which is shown in Figure 1. Our main adaptations for this shared task are in the graph constructor and the input text setup, both of which we explain below.

4.2.1 Architecture

For the construction of the contextualized graph representations, in the previous work Baez Santamaria et al. (2024) used the CoreNLP[5] tool to extract semantic triplets and perform the coreference resolution. However, this tool only has available models for English triplet extraction and coreference resolution.

In this work, we used the cltl-knowledgeExtraction[6] tool for the multilingual triple extraction. We did not use coreference resolution and increased the number of maximum nodes in the graph. The transformer model used for language encoding was not changed, as the underlying *Flan-T5* model already exhibits multilingual capabilities (Chung et al., 2022).

4.2.2 Input text setup

As the original architecture presented by Baez Santamaria et al. (2024) was proposed for a dialogue-based dataset, considerations were made to use the additional KN information in training. Two different strategies were explored:

- **Sequential:** KN sentences were fed in a sequential manner, concatenating them all together. Then, 1) the HS and 2) the CN were appended (e.g., [KN, ..., KN, HS, CN]).

- **Interspersed:** KN sentences were interspersed with repeated utterances of the targeted HS, and then the CN was appended (e.g., [KN, HS, KN, HS, ..., HS, CN]).

4.3 Experiments

Both the LLM-based and graph-based approaches were trained on the train subset of the dataset and evaluated on the validation subset of the dataset. We first evaluated several versions of the graph-based approach only on the English subset of the dataset, from which the best performers were chosen to be trained on all four languages (English, Spanish, Italian, Basque) available in the dataset. The training was performed on 2 NVIDIA RTX A5000 GPUs, with a batch size of 8 and over 50 epochs. All experiments performed with the graph-based approach are described in Table 1, which shows the System ID associated with each variation of the approach.

As for the LLM-based approach, different configurations were tested on all four of the available languages in the dataset. These configurations are shown in Table 2.

[5] https://stanfordnlp.github.io/CoreNLP/

[6] https://github.com/leolani/cltl-knowledgeextraction

ID	Maximum tokens	Prompt strategy
LLMB (7)	100	1
LLMB (8)	60	1
LLMB (9)	2000	2

Table 2: Configurations for our LLM-based approach, LLMB stands for the LLM-based approach. Prompting strategies are described in Section 4.1.

5 Results and Discussion

The evaluation was performed using the official evaluation scripts provided by the task organizers.[7] The evaluation includes both traditional metrics (RougeL, BLEU, BERTScore, generation length, novelty), and the JudgeLM score, an LLM-based evaluation (Zhu et al., 2023). We discuss the obtained results below.

5.1 Development phase

According to the results for the graph-based approach (shown in Table 3), systems GB (4) and GB (1) were the top performers, with GB (4) having better performance on the traditional metrics and GB (1) on the JudgeLM ranking. The decision was made to move forward with system GB (4).

ID	JudgeLM	RougeL	BLEU	BERT	gen	novelty
GB(1)	**282.5**	**0.4550**	0.3516	0.7891	30.187	0.7811
GB(2)	226	0.4173	0.3036	0.7651	**35.625**	0.7825
GB(3)	272	0.4328	**0.3623**	0.7851	31.737	**0.7857**
GB(4)	273	0.4547	0.3585	**0.7914**	30.725	0.7838

Table 3: Results for the different configurations of the graph-based approach on the English subset of the dataset. The best results are highlighted in bold.

Considering the results from Table 3, we trained the systems GB (5) and GB (6) without coreference resolution and with interspersed data, the only difference being the different triple extraction engines (OpenIE and CLTL respectively). We implemented the OpenIE tool as a fallback in system GB (6), as using the CLTL tool, we were able to extract triplets only for about 30% of the training dataset.

The LLM-based approach was tested with the two different strategies presented in Section 4.1, being systems LLMB (7) and LLMB (9), respectively, with an additional experiment (system LLMB (8)) with a reduced output token window size of 60 to test the importance of this parameter.

Finally, the two GB (5, 6) and three LLMB (7, 8, 9) systems were evaluated using both traditional metrics and the JudgeLM score. We performed both global (all languages) evaluation and per-language evaluation, see Appendix B for details.

While the graph-based systems performed better in terms of the traditional metrics (i.e., RougeL, BLEU, BERTScore), the LLM-based approach performed better in terms of the JudgeLM score. We observed the same trend during the evaluation phase, which we describe in more detail below.

5.2 Evaluation phase

We submitted the three best-performing models per language based on the results obtained in the development phase:

- **Run 1:** Best performing systems in terms of traditional metrics.
- **Run 2:** Best performing systems in terms of their JudgeLM score.
- **Run 3:** A combination of the two.

The composition of the submission files is further explained in Table 4 (see Tables 1 and 2 for details about each system).

Language	Run 1	Run 2	Run 3
English	GB (6)	LLMB (7)	LLMB (7)
Spanish	GB (6)	LLMB (7)	GB (5)
Italian	GB (5)	LLMB (9)	GB (6)
Basque	GB (5)	LLMB (7)	GB (5)

Table 4: Composition of each submission run, with the ID of the system that was used to generate the CNs (see Tables 1 and 2).

Table 5 presents the official results obtained on the test set. We can observe that our run 2 yielded the best performance according to the JudgeLM score for all the languages. The final evaluation results were ranked primarily by the JudgeLM score. Our LLM-based run 2 was ranked 2nd for English and 3rd for Spanish, which highlights a promising performance and potential of prompt-based techniques in complex tasks like CN generation. This is in line with previous studies (e.g., Doğanç and Markov, 2023, Papaluca et al., 2024, Vatsal and Dubey, 2024, Gan et al., 2024). Our graph-based technique ranked lower in terms of the JudgeLM score but scored higher in terms of the traditional metrics than our LLM-based approach (full results on the test set available on the official shared task website[8]).

[7] https://github.com/hitz-zentroa/eval-MCG-COLING-2025

[8] https://sites.google.com/view/multilang-counterspeech-gen/shared-task

Lang.	Rank	Best Run	JudgeLM	Avg.
English	2/29	2	2498.5	459.4
Spanish	5/27	2	2086.0	358.85
Italian	8/24	2	1630.5	307.7
Basque	3/24	2	1919.0	380.8

Table 5: Final results on the test set.

5.3 Error analysis

In Appendix C, we provide examples of CNs generated by the systems submitted for the shared task. In the English examples, we can observe a pattern where the graph-based approach tends to only concatenate parts of the provided KN sentences, while the LLM-based approach generates more varied CNs that do not include the KN sentences directly. In this sense, the graph-based approach may be more viable in an application where it is necessary to keep KN information intact, while the LLM-based approach fits better in environments that allow for casual and creative language.

6 Conclusion

We have shown that prompting techniques as presented in Doğanç and Markov (2023) and Vallecillo-Rodríguez et al. (2023) can produce competitive results for the CN generation task. However, we believe that for less-resourced languages like Basque, alternative methods such as our graph-based approach are worth exploring. In this sense, future studies implementing graph-inspired architectures (like GoT) could bring the best of both worlds and introduce further improvements in the quality and effectiveness of automatically generated CNs.

Based on the fact that our graph-based approach performed better in terms of the traditional metrics than in terms of the JudgeLM score, it is worth exploring whether the JudgeLM system may exhibit biases towards LLM-generated text, as it has been shown previously for other evaluation systems (Dai et al., 2024).

References

Selene Baez Santamaria, Helena Gomez Adorno, and Ilia Markov. 2024. Contextualized graph representations for generating counter-narratives against hate speech. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7664–7674, Miami, Florida, USA. Association for Computational Linguistics.

Susan Benesch. 2014. Countering dangerous speech: New ideas for genocide prevention. *SSRN Electronic Journal*.

Helena Bonaldi, Sara Dellantonio, Serra Sinem Tekiroğlu, and Marco Guerini. 2022. Human-machine collaboration approaches to build a dialogue dataset for hate speech countering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8031–8049, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tom B. Brown et al. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Hyung Won Chung et al. 2022. Scaling instruction-finetuned language models. *Preprint*, arXiv:2210.11416.

Yi-Ling Chung, Elizaveta Kuzmenko, Serra Sinem Tekiroglu, and Marco Guerini. 2019. Conan - counter narratives through nichesourcing: a multilingual dataset of responses to fight online hate speech. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Sunhao Dai, Yuqi Zhou, Liang Pang, Weihao Liu, Xiaolin Hu, Yong Liu, Xiao Zhang, Gang Wang, and Jun Xu. 2024. Neural retrievers are biased towards llm-generated content. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 526–537. ACM.

Mekselina Doğanç and Ilia Markov. 2023. From generic to personalized: Investigating strategies for generating targeted counter narratives against hate speech. In *Proceedings of the 1st Workshop on CounterSpeech for Online Abuse (CS4OA)*, pages 1–12, Prague, Czechia. Association for Computational Linguistics.

Silvia Donzelli. 2021. Countering harmful speech online. (in)effective strategies and the duty to counterspeak. *Phenomenology & amp; Mind*, page 76.

Yujian Gan, Massimo Poesio, and Juntao Yu. 2024. Assessing the capabilities of large language models in coreference: An evaluation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1645–1665, Torino, Italia. ELRA and ICCL.

Sameer Hinduja and Justin W. Patchin. 2007. Offline consequences of online victimization. *Journal of School Violence*, 6(3):89–112.

Nazmine, Khan Manan, Hannan Khan Tareen, Sidra Noreen, and Muhammad Tariq. 2021. Hate speech and social media: A systematic review. *Turkish Online Journal of Qualitative Inquiry*, 12:5285–5294.

Andrea Papaluca, Daniel Krefl, Sergio Rodríguez Méndez, Artem Lensky, and Hanna Suominen. 2024. Zero- and few-shots knowledge graph triplet extraction with large language models. In *Proceedings of the 1st Workshop on Knowledge Graphs and*

Large Language Models (KaLLM 2024), pages 12–23, Bangkok, Thailand. Association for Computational Linguistics.

Jing Qian, Anna Bethke, Yinyin Liu, Elizabeth Belding, and William Yang Wang. 2019. A benchmark dataset for learning to intervene in online hate speech. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4755–4764, Hong Kong, China. Association for Computational Linguistics.

Tuija Saresma, Sanna Karkulehto, and Piia Varis. 2021. *Gendered Violence Online: Hate Speech as an Intersection of Misogyny and Racism*, pages 221–243. Springer International Publishing, Cham.

Carla Schieb and Mike Preuss. 2016. Governing hate speech by means of counterspeech on facebook.

Ruth Spence, Antonia Bifulco, Paula Bradbury, Elena Martellozzo, and Jeffrey DeMarco. 2023. The psychological impacts of content moderation on content moderators: A qualitative study. *Cyberpsychology: Journal of Psychosocial Research on Cyberspace*, 17(4):Article 8.

Maria Estrella Vallecillo-Rodríguez, Arturo Montejo-Raéz, and Maria Teresa Martín-Valdivia. 2023. Automatic counter-narrative generation for hate speech in spanish. *Procesamiento del Lenguaje Natural*, 71(0):227–245.

Shubham Vatsal and Harsh Dubey. 2024. A survey of prompt engineering methods in large language models for different nlp tasks. *Preprint*, arXiv:2407.12994.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint*. ArXiv:2201.11903.

Yao Yao, Zuchao Li, and Hai Zhao. 2023. Beyond chain-of-thought, effective graph-of-thought reasoning in language models. *arXiv preprint*.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. Judgelm: Fine-tuned large language models are scalable judges. *Preprint*, arXiv:2310.17631.

A GPT-4o Prompts

```
prompt = f'''You will be given a hate speech sentence, a target for the hate speech and several background knowledge sentences related to the hate speech. First, create a profile for the person who wrote the hate speech comment. The profile should include something about the time period they grew up in and the culture they are part of. Then, considering the profile, the hate speech, the target of the hate speech and the background knowledge sentences, write a personalized counternarrative. Only return the counternarrative, nothing more.

\nHate speech: {hate_speech}
\nBackground knowledge: {background_knowledge}
\nTarget: {target}'''

messages = [
    {"role": "system", "content": "You are an expert at generating counternarratives to hate speech based on background knowledge. "},
    {"role": "user", "content": prompt}
]
```

Listing 1: Personalized CN prompt (Doğanç and Markov, 2023) for GPT-4o.

```
prompt = f'''Think carefully about the next steps you will take to generate a counternarrative. You will be given a hate speech sentence, a target for the hate speech and several background knowledge sentences related to the hate speech.

\nStep 1 - Create a profile for the person who wrote the hate speech comment. The profile should include something about the time period they grew up in and the culture they are part of. Do not return what you write in this step.

\nStep 2 - Considering the hate speech, the target of the hate speech and the background knowledge sentences, write a counternarrative. Do not return what you write in this step.

\nStep 3 - Now, take the counternarrative you wrote in Step 2 and personalize it based on the profile you created in Step 1.

\nStep 4 - Only return the results of step 3, do not return anything else. Limit your response to around 60 words, and use the same language as the hate speech.

\nHate speech: {hate_speech}
\nBackground knowledge: {background_knowledge}
\nTarget: {target}'''

messages = [
    {"role": "system", "content": "You are an expert at generating counternarratives to hate speech based on background knowledge. "},
    {"role": "user", "content": prompt}
]
```

Listing 2: Chain-of-thought prompt (Wei et al., 2023) for GPT-4o.

B Results on the Validation Set

ID	JudgeLM score	RougeL	BLEU	BERT	gen	novelty
GB (5)	1237	0.3730	0.2764	0.7710	30.330	0.8028
GB (6)	961.5	**0.3773**	**0.2881**	**0.7730**	30.075	0.8016
LLMB (7)	**1593.5**	0.1301	0.0212	0.6853	**55.547**	0.8289
LLMB (8)	833.5	0.1182	0.0169	0.6517	28.620	0.8308
LLMB (9)	1185.5	0.1290	0.0199	0.6823	47.677	**0.8331**

Table 6: Evaluation results during the development phase for all the languages in the ML-KN-MTCONAN dataset. The best results are highlighted in bold.

ID	JudgeLM score	RougeL	BLEU	BERT	gen	novelty
GB (5)	78	0.4727	0.3698	0.8010	29.44	0.7862
GB (6)	80.5	**0.4916**	**0.4176**	**0.8082**	30.44	0.7942
LLMB (7)	**373.5**	0.1310	0.0147	0.6822	**72.16**	0.8316
LLMB (9)	309.5	0.1285	0.0149	0.6783	49.39	**0.8366**

Table 7: English evaluation results during the development phase. The best results are highlighted in bold.

ID	JudgeLM score	RougeL	BLEU	BERT	gen	novelty
GB (5)	70	0.3908	0.2798	0.7722	35.67	0.7726
GB (6)	67	**0.4003**	**0.2946**	**0.7781**	34.66	0.7666
LLMB (7)	**358.5**	0.1680	0.0335	0.6965	**63.95**	0.7971
LLMB (9)	313.5	0.1595	0.0290	0.6924	49.87	**0.8009**

Table 8: Spanish evaluation results during the development phase. The best results are highlighted in bold.

ID	JudgeLM score	RougeL	BLEU	BERT	gen	novelty
GB (5)	69	**0.3438**	**0.2477**	0.7649	32.75	0.7914
GB (6)	79	0.3406	0.2303	**0.7651**	30.99	0.7881
LLMB (7)	318	0.1321	0.0208	0.6841	**51.39**	**0.8183**
LLMB (9)	**326**	0.1383	0.0257	0.6819	49.64	0.8171

Table 9: Italian evaluation results during the development phase. The best results are highlighted in bold.

ID	JudgeLM score	RougeL	BLEU	BERT	gen	novelty
GB (5)	259.5	**0.2848**	**0.1903**	**0.7461**	23.46	0.8612
GB (6)	189	0.2808	0.1902	0.7406	24.21	0.8574
LLMB (7)	**273.5**	0.0889	0.0109	0.6782	34.70	0.8688
LLMB (9)	174.5	0.0891	0.0113	0.6768	**41.81**	**0.8780**

Table 10: Basque evaluation results during the development phase. The best results are highlighted in bold.

C Examples of Generated Counter-narratives

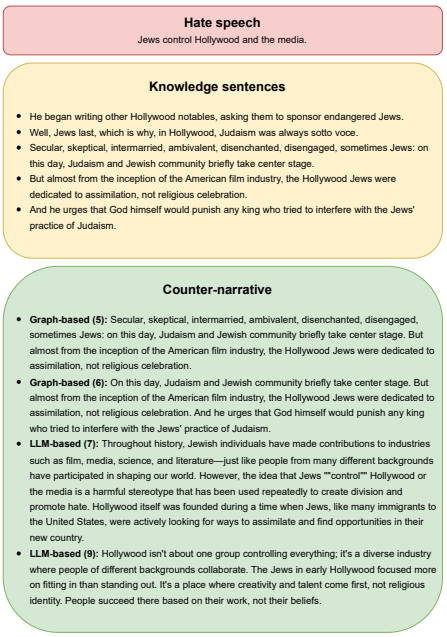

Figure 2: Examples of English counter-narratives generated by each of the systems submitted to the shared task (see Tables 1 and 2).

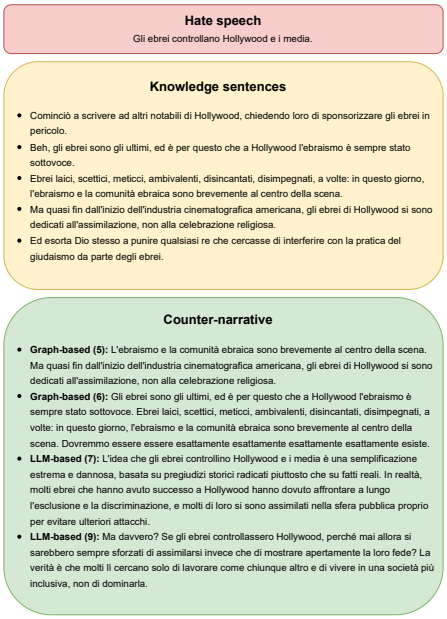

Figure 4: Examples of Italian counter-narratives generated by each of the systems submitted to the shared task (see Tables 1 and 2).

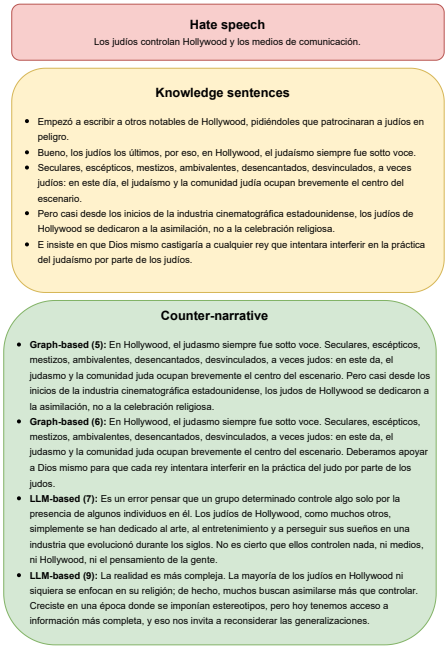

Figure 3: Examples of Spanish counter-narratives generated by each of the systems submitted to the shared task (see Tables 1 and 2).

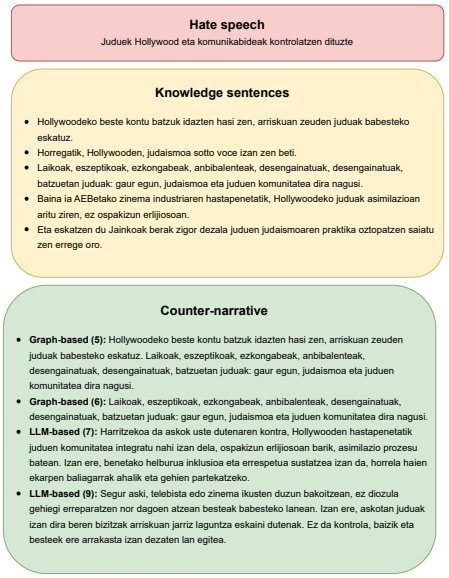

Figure 5: Examples of Basque counter-narratives generated by each of the systems submitted to the shared task (see Tables 1 and 2).