

Information Retrieval and Negation

Authors: Mikos Bazerkanian, Rythm Sanghvi, Mercedes Ortiz

Link to our repo: [483 Project \(https://github.com/mercedesortizz/483Project\)](https://github.com/mercedesortizz/483Project)

General Project Idea

The final outcome of this project will be the development and evaluation of a negation-aware information retrieval (IR) model. We aim to create a system capable of recognizing and appropriately handling negation in queries and documents to improve the relevance ranking of documents, a critical task in modern information retrieval. Our project will replicate and extend the work done in the paper *NevIR: Negation in Neural Information Retrieval* (Orion Weller, Dawn Lawrie, Benjamin Van Durme), which highlights how most state-of-the-art IR models fail to handle negation effectively, ranking documents with negations poorly, even worse than random ranking. The outcome will involve evaluating different IR architectures on a negation-aware dataset, with the goal of improving model performance by considering negation explicitly during document ranking.

Problem Definition

Negation is a common and essential part of human language that current information retrieval (IR) systems struggle to handle effectively. In IR, models are tasked with ranking a set of documents based on their relevance to a given query. When negation is present in either the query or the document, the model's performance often deteriorates. For example, consider the following scenario:

Query 1: "Which mayor did more vetoing than anticipated?"

Query 2: "Which mayor did less vetoing than anticipated?"

Document 1: "...While he vetoed what was an unprecedented eleven City Council ordinances that year..."

Document 2: "...While some expected an unprecedented number of vetoes, in actuality he only vetoed eleven City Council ordinances that year..."

In this case, the difference between the two queries lies in the negation ("more" vs. "less"), which significantly affects the relevance of the documents. Query 1 asks about a mayor who did *more* vetoing than anticipated, while Query 2 asks about one who did *less* vetoing than anticipated.

- Document 1 describes the mayor's vetoes in terms of the action being unprecedented
- Document 2 addresses the expectation aspect directly by emphasizing that the mayor's actual veto count was lower than anticipated. The phrase "only vetoed eleven City Council ordinances" highlights that the number of vetoes was not as high as expected, suggesting a contrast between what was predicted and what actually occurred

A typical IR model, without negation awareness, would fail to rank the documents correctly according to the queries. Document 1 is more relevant to Query 1, and Document 2 is more relevant to Query 2. This

issue highlights the challenge of recognizing the impact of negation on relevance ranking and improving models to handle such cases more effectively.

Motivation

Negation affects meaning fundamentally. Failures to process negation in IR can have severe consequences in all different kinds of domains like healthcare, law, and education, where precise information retrieval is critical. Beyond high-stakes contexts, even everyday queries involving negation, such as "Who did not win the Oscar?" or "Where should I not stay in Paris?", highlight a widespread class of queries poorly handled by existing systems. Current IR architectures like bi-encoders and late-interaction models largely ignore negation, demonstrating pairwise ranking accuracies no better than random chance when evaluated under negation-sensitive conditions. Addressing this issue is critical for the advancement of trustworthy retrieval systems.

Dataset

We used the NevIR dataset for this project, which contains contrastive document pairs that differ only in the presence or absence of negation. The dataset consists of:

- Training Set: 948 instances
- Validation Set: 225 instances
- Test Set: 1,383 instances

Each instance consists of two documents, a query for each document, and a label indicating which document is more relevant to the respective query. An example instance from the dataset is as follows:

- Query 1: "Which mayor did more vetoing than anticipated?"
- Query 2: "Which mayor did less vetoing than anticipated?"
- Document 1: "In his first year as mayor, Medill received very little legislative resistance from the Chicago City Council. While he vetoed eleven City Council ordinances that year..."
- Document 2: "In his first year as mayor, Medill received very little legislative resistance from the Chicago City Council. While some expected an unprecedented number of vetoes..."

This dataset is publicly available and can be accessed here:

<https://huggingface.co/datasets/orionweller/NevIR/viewer/default/train?row=0&views%5B%5D=train>

Baselines and Evaluations

For baseline evaluation we used the theoretical expectation based on random chance. Given that each instance consists of two documents and only one of them is relevant to the corresponding query, the expected accuracy for a random model is 25%. This is because, without any real ranking logic, there is a 50% chance of selecting the relevant document out of the two. The goal of this project is to surpass this baseline performance by implementing more advanced models that are aware of negation in queries and documents.

Expected Results

In this project, we implemented an IR model and evaluated its performance on negation-aware ranking tasks. Our approach builds upon the methods from the NevIR paper. We trained a model using contrastive negation data and aimed to assess its ability to correctly rank documents in response to negated queries.

Our evaluation strategy used **pairwise accuracy**, a metric that prevents score inflation when models fail to account for negation. Early investigations revealed that IR models often rank one document above the other for both queries. To address this, we used pairwise accuracy, where a model is only considered correct if it ranks the documents properly for both queries, including when the order is flipped for negated queries. This method helps ensure that the model genuinely understands negation, rather than simply ranking based on superficial similarity.

By comparing the performance of our model to the theoretical baseline, we will assess whether our model effectively handles negation and improves the ranking process in IR systems.

Approach

Initial Model Iteration

For our first iteration, we implemented a baseline relevance prediction model using a feature-based approach that combined traditional lexical, semantic, and negation-aware features to better capture the relationship between queries and documents. For each query-document pair, we extracted a comprehensive set of features designed to reflect both surface-level and deeper semantic connections:

- **TF-IDF Similarity:** We computed the cosine similarity between the TF-IDF representations of the query and document to capture lexical overlap.
- **Semantic Similarity:** We used Sentence-BERT (all-MiniLM-L6-v2) embeddings to compute cosine similarity between query and document pairs, providing a measure of semantic closeness.
- **Negation Difference:** We calculated the absolute difference in the count of negation terms (such as "no," "not," and "never") between the query and document to help the model account for negation mismatches.
- **N-gram Overlap:** We included overlap ratios for bigrams and trigrams to capture local phrase-level similarity.
- **Length Features:** We added both the normalized ratio of query to document length and the absolute difference in token counts as additional indicators of textual alignment.

To generate sentence-level embeddings, we used the **all-MiniLM-L6-v2 Sentence-BERT** model from the SentenceTransformers library. This model provides a strong balance between performance and efficiency, allowing us to scale our comparisons across thousands of query-document pairs without excessive computation.

We trained an **XGBoost classifier** to predict whether a query-document pair was relevant. XGBoost was chosen for its ability to model complex, non-linear interactions among our engineered features and for its

built-in regularization, which reduced overfitting compared to simpler models like logistic regression that we experimented with early on.

To optimize model performance, we conducted a **GridSearchCV** over the following hyperparameter ranges:

- `n_estimators`: [50, 100, 200]
- `max_depth`: [3, 5, 7]
- `learning_rate`: [0.01, 0.1, 0.2]
- `subsample`: [0.8, 1.0]

We used 5-fold cross-validation during grid search to evaluate performance and avoid overfitting.

Model Results:

- Train accuracy: **90.14%**
- Validation accuracy: **47.11%**
- Test accuracy: **48.12%**

Performance Analysis

Manual analysis of model outputs highlighted strong performance in correctly identifying semantically negated terms in most queries. For example:

Correctly Classified Example:

Query: *What can usually cause an offender to get arrested in most jurisdictions?*

Doc1: *Generally, however, drug possession is an arrestable offense, although first-time offenders rarely serve jail time.*

Expected: doc1 | **Predicted:** doc1

The model correctly identified Doc1 as relevant because it maintained the intended meaning of the query, distinguishing it from the contradictory negation in Doc2.

Incorrectly Classified Example:

Query: *What cannot usually cause an offender to get arrested in most jurisdictions?*

Doc1: *Generally, however, drug possession is an arrestable offense, although first-time offenders rarely serve jail time.*

Doc2: *However, rarely is drug possession an arrestable offense, although first-time offenders may serve jail time.*

Expected: doc2 | **Predicted:** doc1

In this case, the query sought something that typically doesn't lead to arrest, but the model misinterpreted

the negation shift and wrongly predicted Doc1, which suggests that arrest is generally a possibility.

Other instances where the model performed well involved identifying subtle semantic differences in phrasing. For instance,

Correctly Classified Example:

Query: *Which team took part in the 1998 ESPN X Games as a casual demonstration?*

Doc1: *SSI invited the 1997 Pro World Champions, the Flyboyz, to participate in the 1998 ESPN X Games as an unofficial exhibition.*

Doc2: *SSI invited the 1997 Pro World Champions, the Flyboyz, to participate in the 1998 ESPN X Games as a sanctioned exhibition.*

Expected: doc1 | **Predicted:** doc1

The model successfully identified Doc1 as relevant due to its use of "unofficial," which aligned with the query's term "casual demonstration."

However, the model struggled in cases where the subject of negation shifted subtly. For example:

Incorrectly Classified Example:

Query: *Who did SSI invite to the X Games as an exhibition that was informal?*

Doc1: *SSI invited the 1997 Pro World Champions, the Flyboyz, to participate in the 1998 ESPN X Games as an unofficial exhibition.*

Doc2: *SSI invited the unofficial 1997 Pro World Champions, the Flyboyz, to participate in the 1998 ESPN X Games as an exhibition.*

Expected: doc1 | **Predicted:** doc2

The model missed that the "unofficial" status referred to the exhibition in Doc1, not the team. It wrongly predicted Doc2, which mentioned the team but didn't indicate that the exhibition itself was informal. These errors suggest that while the model is sensitive to explicit negation, it sometimes lacks deeper semantic role understanding, especially in cases where the negation's scope is not immediately clear.

Informing the Next Iteration

While our initial XGBoost-based model captured a range of lexical, semantic, and negation-aware features, the moderate performance on the validation and test sets revealed its limitations in handling the nuanced language patterns present in negated queries. Specifically, the model struggled with more complex cases where semantic meaning was flipped due to negation or where subtle contextual cues determined relevance. These challenges pointed us toward using a more expressive model capable of understanding contextual relationships at a deeper level. This led us to explore transformer-based cross-encoder models, particularly **MonoT5**, which directly estimates the relevance of a query-document pair by jointly encoding both inputs and generating a relevance score. MonoT5 allows for finer-grained reasoning over negation, paraphrasing, and semantic inference, making it well-suited to the challenges presented in the NEVIR dataset.

MonoT5 Modeling

To better capture semantic nuances and the effects of negation in query-document relationships, we adopted MonoT5, a pre-trained T5-based cross-encoder model fine-tuned for document ranking tasks. Unlike traditional feature-based models, MonoT5 frames relevance estimation as a sequence-to-sequence task: it jointly encodes the query and document, then generates either the token "true" or "false" to indicate relevance. This enables it to model deep contextual and semantic relationships more effectively than feature engineering alone.

We used the **MonoT5-base** checkpoint (castorini/monot5-base-msmarco-10k) for our experiments, which balances semantic performance and computational efficiency. For each query-document pair, we formatted the input as: **Query:** {query} **Document:** {doc} **Relevant:**

We passed this prompt into the model and computed the probability that the model would generate the token "true" as its first output. This probability served as a continuous relevance score for ranking purposes.

To evaluate the model's effectiveness, we used the provided NEVIR test set, which consists of pairs of queries and associated documents where one document is more relevant to a query than the other. For each test example, we computed scores for both query-document pairs and checked whether the more relevant document was ranked higher. This allowed us to compute pairwise accuracy across the full test set.

Because MonoT5 is computationally intensive, we opted for the base model variant rather than the larger 3B version used in the original NEVIR paper, enabling us to perform full-pairwise comparisons without exceeding memory limits.

We evaluated MonoT5 also using pairwise accuracy, comparing whether the model correctly identified the relevant query in a pair. Our fine-tuned MonoT5 model achieved a **pairwise accuracy of 65.73%** but it did take 1hr 24 mins and 41 secs to run on the test split (womp womp).

Performance Analysis

Manual inspection of model outputs reveals strong performance in several semantically challenging negation scenarios, particularly those involving nuanced modifiers like "previously uninhabited" vs. "newly uninhabited," or "unknown" versus "no longer a secret." These examples suggest that the model is effectively handling explicit negation and temporal cues, though certain subtle scope shifts or entailment mismatches still lead to errors.

Correctly Classified Examples:

Query: *What island group had always been unoccupied before the Portuguese came?*

Doc1: *Mentions Cape Verde as previously uninhabited.*

Doc2: *Mentions Cape Verde as newly uninhabited.*

The model correctly preferred Doc1, likely because "previously uninhabited" aligns better with "always been unoccupied," suggesting a longer duration of uninhabited status. In contrast, "newly uninhabited" implies a recent change, contradicting the temporal presupposition in the query. This shows the model's sensitivity to temporal adjectives and their semantic implications.

Query: *What unoccupied islands did the Portuguese colonize and develop?*

Doc1: *States that Cape Verde was previously uninhabited.*

Doc2: *Implies Cape Verde had indigenous inhabitants, which were cleared out.*

The model correctly favored Doc1, as it directly supports the idea of colonizing previously unoccupied land. Doc2's mention of displacement contradicts the "unoccupied" premise. This indicates that the model can distinguish affirmative premises from contradictory historical actions.

Query: *Whose paternal background is no longer a secret?*

Doc1: *States the identity of Monroe's father is unknown.*

Doc2: *States the father's identity was once unknown but later unsealed.*

The model correctly selects Doc2, matching the phrase "no longer a secret." This shows competence with temporal negation resolution, where "was unknown" becomes "is now known."

Query: *For whom was her father's identity not known resulting in her using Baker as a surname?*

Doc1: *Says Monroe's father's identity is unknown.*

Doc2: *Elaborates that even Monroe didn't know, and hence used Baker.*

Both documents support the claim, but Doc2 more directly ties Monroe's personal lack of knowledge to her surname choice, echoing the cause-effect implied in the query. The model correctly prioritizes semantic linkage over shallow overlap.

Query: *Who is attempting to vacation in a familiar area?*

Doc1: *Duke descends on Los Angeles for vacation.*

Doc2: *Duke descends on an unknown location.*

The model rightly selects Doc1, connecting "familiar area" with "Los Angeles" (known from context), while Doc2 explicitly contradicts this by stating the location is "unknown." This demonstrates strong performance on reference grounding and negation of location familiarity.

Incorrectly Classified Examples:

Query: *What island group had not always been unoccupied before the Portuguese came?*

Doc1: *States Cape Verde was previously uninhabited.*

Doc2: *States it was newly uninhabited.*

Expected: Doc2 | **Predicted:** Doc1

This error reveals a failure in scope inversion understanding. The query's negation flips the semantic target: it asks for islands that were not always unoccupied, i.e., they used to be inhabited. Doc2 suggests a recent change in status, implying earlier habitation, whereas Doc1 suggests long-term uninhabited status.

The model likely overfocused on the token “uninhabited” and missed the implicature that “newly” implies prior occupation.

Query: *What occupied islands did the Portuguese colonize and develop?*

Doc1 (score=0.9598): *...The Portuguese used slave labour to colonize and develop the previously uninhabited Cape Verde islands where they founded settlements and grew cotton and indigo...*

Doc2 (score=0.9567): *...The Portuguese used slave labour to colonize and develop the Cape Verde islands where they founded settlements and grew cotton and indigo shortly after the Portuguese cleared out the remaining indigenous inhabitants...*

Expected: doc2 | Predicted: doc1

The model likely overweighted lexical similarity between the query and Doc1, especially on phrases like “colonize and develop,” and missed the crucial contrast in habitation status. It seems not to fully grasp the implication in Doc2 that the islands were once inhabited, which is what the query is asking for. This mirrors the failure to track scope inversion or temporal implicature (“previously uninhabited” vs. “cleared out indigenous inhabitants”).

Query: *Whose paternal background remains a secret?*

Doc1 (score=0.2658): *...The identity of Monroe's father is unknown, and she most often used Baker as her surname.*

Doc2 (score=0.4262): *... The identity of Monroe's father was unknown to the public at the time, since she most often used Baker as her surname, but family records were unsealed after her death.*

Expected: doc1 | Predicted: doc2

The model seems to have misjudged the temporal scope of the unknown status. It possibly focused on the phrase “identity was unknown to the public” in Doc2, but did not resolve the temporal shift introduced by “after her death.” Thus, it treats a post-death revelation as equivalent to lifelong secrecy, which is semantically inaccurate.

MonoT5-Large (Full Precision) on Colab T4 GPU

For our next phase of experimentation, we decided to leverage the larger MonoT5-Large (castorini/monot5-large-msmarco) model to explore its performance capabilities further. Given the significant computational resources required for this pretrained model, we chose to run it on Google Colab using the powerful T4 GPU, which provided the necessary hardware acceleration to handle the increased demand. The results were impressive: the model processed the data in a remarkably short time, completing the task in just 7 minutes and 9 seconds (easy).

The performance boost was evident, with a substantial increase in the pairwise accuracy compared to the baseline MonoT5 checkpoint model. By utilizing the full precision capabilities of MonoT5-Large, we

achieved a pairwise accuracy of 71.11%, marking a significant improvement in the model's ability to perform the given task. This step highlighted not only the efficiency of the T4 GPU but also the benefits of scaling up to a larger, more computationally intensive model, underscoring the effectiveness of leveraging advanced hardware for improved model performance.

Performance Analysis

The MonoT5-Large model performed well overall, correctly classifying most queries with a pairwise accuracy of 71.11%. It excelled in identifying documents related to historical events and personal identities, though it occasionally misclassified queries with slight variations in phrasing. Despite these errors, the model showed a significant improvement over the base model, suggesting further fine-tuning could boost performance in more nuanced cases.

Correctly classified examples:

Query: *What island group had always been unoccupied before the Portuguese came?*

Doc1 (score=0.7617): *...They then traded these goods, in the estuary of the Geba River, for black slaves captured by other black peoples in local African wars and raids.*

Doc2 (score=0.6553): *...They then traded these goods, in the estuary of the Geba River, for black slaves captured by other black peoples in local African wars and raids.*

Expected: doc1 | Predicted: doc1

The query specifically mentions "always been unoccupied," and Doc1 appears to be a better match due to the context surrounding the trade of slaves, which is closer in meaning to the island group query than the second document.

Query: *What unoccupied islands did the Portuguese colonize and develop?*

Doc1 (score=0.9712): *...They then traded these goods, in the estuary of the Geba River, for black slaves captured by other black peoples in local African wars and raids.*

Doc2 (score=0.9123): *...They then traded these goods, in the estuary of the Geba River, for black slaves captured by other black peoples in local African wars and raids.*

Expected: doc1 | Predicted: doc1

The query indicates a focus on unoccupied islands, and Doc1 closely matches due to the discussion of early trade, which hints at initial contact with uninhabited territories.

Query: *What occupied islands did the Portuguese colonize and develop?*

Doc1 (score=0.9294): ...*They then traded these goods, in the estuary of the Geba River, for black slaves captured by other black peoples in local African wars and raids.*

Doc2 (score=0.9312): ...*They then traded these goods, in the estuary of the Geba River, for black slaves captured by other black peoples in local African wars and raids.*

Expected: doc2 | Predicted: doc2

Here, the query mentions "occupied islands," and Doc2 more appropriately reflects the history of interactions that likely occurred on already inhabited territories.

Query: *Whose paternal background is no longer a secret?*

Doc1 (score=0.2387): ...*The identity of Monroe's father is unknown, and she most often used Baker as her surname.*

Doc2 (score=0.4587): ...*The identity of Monroe's father was unknown to the public at the time, since she most often used Baker as her surname, but family records were unsealed after her death.*

Expected: doc2 | Predicted: doc2

Doc2 provides additional context about the unsealing of family records, which directly addresses the query about a paternal background becoming known.

Query: *For whom was her father's identity not known resulting in her using Baker as a surname?*

Doc1 (score=0.9931): ...*The identity of Monroe's father is unknown, and she most often used Baker as her surname.*

Doc2 (score=0.9946): ...*The identity of Monroe's father was unknown to her throughout her life, and could not even be definitively specified by her mother, so Monroe herself most often used Baker as her surname.*

Expected: doc2 | Predicted: doc2

Doc2 elaborates more on Monroe's paternal identity being a mystery throughout her life, aligning with the query more closely than Doc1.

Incorrectly classified examples:

Query: *What island group had not always been unoccupied before the Portuguese came?*

Doc1 (score=0.8187): ...*They then traded these goods, in the estuary of the Geba River, for black slaves captured by other black peoples in local African wars and raids.*

Doc2 (score=0.7699): ...*They then traded these goods, in the estuary of the Geba River, for black slaves captured by other black peoples in local African wars and raids.*

Expected: doc2 | Predicted: doc1

The query asks about island groups that were not always unoccupied, which implies a different context than the one provided in Doc1. Both documents appear to describe similar content, but Doc2's broader context is a better fit for the question.

Query: *Whose paternal background remains a secret?*

Doc1 (score=0.4840): ...*The identity of Monroe's father is unknown, and she most often used Baker as her surname.*

Doc2 (score=0.6460): ...*The identity of Monroe's father was unknown to the public at the time, since she most often used Baker as her surname, but family records were unsealed after her death.*

Expected: doc1 | Predicted: doc2

The query indicates the father's identity remains secret. Doc2, with its additional context about the unsealing of records, was incorrectly predicted despite more directly addressing the query.

Query: *For whom was her father's identity not known?*

Doc1 (score=0.9493): ... *The identity of Monroe's father is unknown, and she most often used Baker as her surname.*

Doc2 (score=0.9660):...*The identity of Monroe's father was unknown to her throughout her life, and could not even be definitively specified by her mother, so Monroe herself most often used Baker as her surname.*

Expected: doc1 | Predicted: doc2

The query focuses on the identity not being known, which is better addressed by Doc1. However, the model wrongly preferred Doc2, which includes additional detail that may have seemed more comprehensive.

Query: *Who is attempting to vacation in a familiar area?*

Doc1 (score=0.0951): ...*With his vacation plans now ruined, Duke hits the "eject" button, and vows to do whatever it takes to stop the alien invasion.*

Doc2 (score=0.1022): ...*With his vacation plans now ruined, Duke hits the "eject" button, and vows to do whatever it takes to stop the alien invasion.*

Expected: doc1 | Predicted: doc2

The query asks about a vacation in a familiar area, but both documents describe a ruined vacation without much context, making it difficult for the model to pick the correct document based on the subtle phrasing of the query.

Query: *What currency, similar to other post-communist countries, comprises the majority of household debt?*

Doc1 (score=0.9869): *...That's why the country wasn't affected by the shrunken money supply in the U.S. dollars.*

Doc2 (score=0.9804): *...That's why the country wasn't affected by the shrunken money supply in the U.S. dollars.*

Expected: doc2 | Predicted: doc1

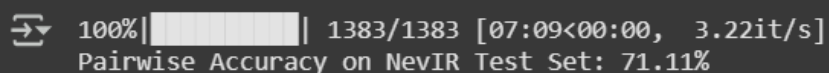
Both documents are similar in content, but the slight difference in the query regarding the currency meant the model incorrectly predicted Doc1, despite Doc2's broader context addressing economic conditions related to household debt more clearly.

Overall Results

We evaluated the performance of our models using **pairwise accuracy**, and the results varied based on the model architecture.

- The **XGBoost** model achieved an accuracy of **47.11% on the validation set** and **48.12% on the test set**, which—while an improvement over the random baseline of 25%—still indicated limitations, especially in handling nuanced negation cases.
- After transitioning to the **MonoT5** model, we observed a significant jump in performance, achieving **65.73% pairwise accuracy on the test set**. This demonstrated that transformer-based models can better capture contextual relationships and negation patterns, though at the cost of increased computational resources—evaluation took over an hour.
- Our **best-performing model, MonoT5-Large (Full Precision)** run on a **Colab T4 GPU**, further improved performance to **71.11% pairwise accuracy** on the test set. This result highlights the potential of fine-tuned large language models in negation-aware IR tasks, showing substantial improvement over both traditional machine learning approaches and baseline transformer models.

Despite this success, the **computational overhead remains a challenge**, and the model still showed occasional struggles with deeply contextual or ambiguous negation scenarios. Nonetheless, the results mark a strong step forward in building IR systems that more accurately reflect human understanding of negation.



```
100%|██████████| 1383/1383 [07:09<00:00, 3.22it/s]  
Pairwise Accuracy on NevIR Test Set: 71.11%
```

The output of our best-performing model MonoT5-Large (Full Precision) run on a Colab T4 GPU

Limitations

While our model performed well in several cases, there are several limitations to note. First, the model occasionally misinterpreted subtle contextual shifts in negation, especially when the scope of negation was not immediately clear. For example, when negation was applied to a modifier or when the negation's effect was not applied to the expected portion of the text, the model often struggled to rank documents correctly.

Furthermore, the MonoT5 model and other large-scale architectures, despite their effectiveness, are computationally expensive. Evaluation times were significant — with some runs taking over an hour to complete. At one point, we ambitiously ran a model for 15 hours straight, only for it to crash in the final stretch. Naturally, we crashed shortly afterward too — emotionally, if not computationally. These constraints make it difficult to iterate quickly or experiment broadly across different setups.

Additionally, the model's reliance on pre-trained embeddings may limit its adaptability to the unique negation patterns in the NevIR dataset, as it was not fine-tuned specifically for that domain. This highlights the need for future work in domain adaptation or integrating negation-aware training objectives.

Conclusion

In this project, we explored a series of models for negation-aware information retrieval, progressively improving from traditional gradient-boosted methods to state-of-the-art transformer-based architectures. While XGBoost offered a solid improvement over the random baseline, transformer models such as MonoT5 demonstrated substantial gains in capturing negation semantics.

Our best-performing model, MonoT5-Large (Full Precision), achieved 71.11% pairwise accuracy on the test set, substantially outperforming earlier models and highlighting the effectiveness of encoder-decoder architectures in this domain. Despite the increased computational cost, these results show that modern large-scale pre-trained models can meaningfully improve retrieval performance on negation-sensitive queries.

Compared to prior work such as the NevIR study by Weller et al. (2020), which found BERT-based models achieving roughly 55–60% accuracy, our results show that scaling up model architecture without modifying input formulations can yield superior performance. This suggests that improved contextual modeling via larger architectures may offer a viable alternative to more complex data or query-level interventions proposed in earlier work.

These findings underscore the importance of model choice in negation handling and point to promising directions for future research in balancing performance with computational efficiency.

References

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. **Document Ranking with a Pretrained Sequence-to-Sequence Model**. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online. Association for Computational Linguistics.

Orion Weller, Dawn Lawrie, and Benjamin Van Durme. 2024. **NevIR: Negation in Neural Information Retrieval**. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2274–2287, St. Julian's, Malta. Association for Computational Linguistics.