A Comprehensive Taxonomy of Negation for NLP and Neural Retrievers

Roxana Petcu, Samarth Bhargav, Maarten de Rijke, Evangelos Kanoulas
University of Amsterdam, Amsterdam, The Netherlands
{r.m.petcu, s.bhargav, m.derijke, e.kanoulas}@uva.nl

Abstract

Understanding and solving complex reasoning tasks is vital for addressing the information needs of a user. Although dense neural models learn contextualised embeddings, they still underperform on queries containing negation. To understand this phenomenon, we study negation in both traditional neural information retrieval and LLM-based models. We (1) introduce a taxonomy of negation that derives from philosophical, linguistic, and logical definitions; (2) generate two benchmark datasets that can be used to evaluate the performance of neural information retrieval models and to fine-tune models for a more robust performance on negation; and (3) propose a logic-based classification mechanism that can be used to analyze the performance of retrieval models on existing datasets. Our taxonomy produces a balanced data distribution over negation types, providing a better training setup that leads to faster convergence on the NevIR dataset. Moreover, we propose a classification schema that reveals the coverage of negation types in existing datasets, offering insights into the factors that might affect the generalization of fine-tuned models on negation.

1 Introduction

A key factor contributing to accurate relevance in neural information retrieval (IR) systems, LLM-based re-rankers, and retrieval augmented generation (RAG) is acquiring language understanding capabilities through pre-training (Hosseini et al., 2021). Despite their extensive training setups, these models show persistent difficulty in handling negation (McKenzie et al., 2024), both in spoken and written language (Ortega et al., 2016). Negation is linguistically a complex phenomenon that, while guaranteed to be present in the training regime of any model, takes different forms depending on the task at hand. Human comprehension of negation comes as a result of understanding linguistic, morphological, and syntactic construction along with verbal cues (as defined in Appendix A.1) and facial expressions (Zuanazzi et al., 2023). However, this multifaceted linguistic phenomenon is often reduced to a binary description in language processing systems: Does negation exist or not in a specific data set (Weller et al., 2024; Zhang et al., 2024a), and is it encoded or not by a model (Ravichander et al., 2022). Addressing these discrepancies between human and system understanding of negation, we ask the following research questions:

(RQ1) Can we design a comprehensive taxonomy for negation?

(RQ2) How can this taxonomy be applied to generate a more complete and balanced dataset?

(RQ3) In what manner does model performance differ when fine-tuned on the taxonomy-driven dataset
versus prior existing datasets?

(RQ4) How can this taxonomy be used to understand why models underperform on existing negation datasets?

RQ1 aims to bring together research from the linguistic literature in a taxonomy on negation. We design our taxonomy to be exhaustive, with no overlap, relevant to IR tasks, and containing a balance of granularity that is appropriate for the AI community. To address RQ2, we propose two synthetically generated datasets that cover all proposed negation types. Figure 1 illustrates the task alongside the data type represented in our datasets. RQ3 analyzes the performance of neural IR models, providing insight into the gap between human understanding and LLM encoding of negation. RQ4 connects the taxonomy to formalizations that can be used as data classification mechanisms, which will allow us to study the distribution of negation in existing datasets and identify possible reasons why fine-tuning on such datasets does not guarantee a performance boost for neural IR models.

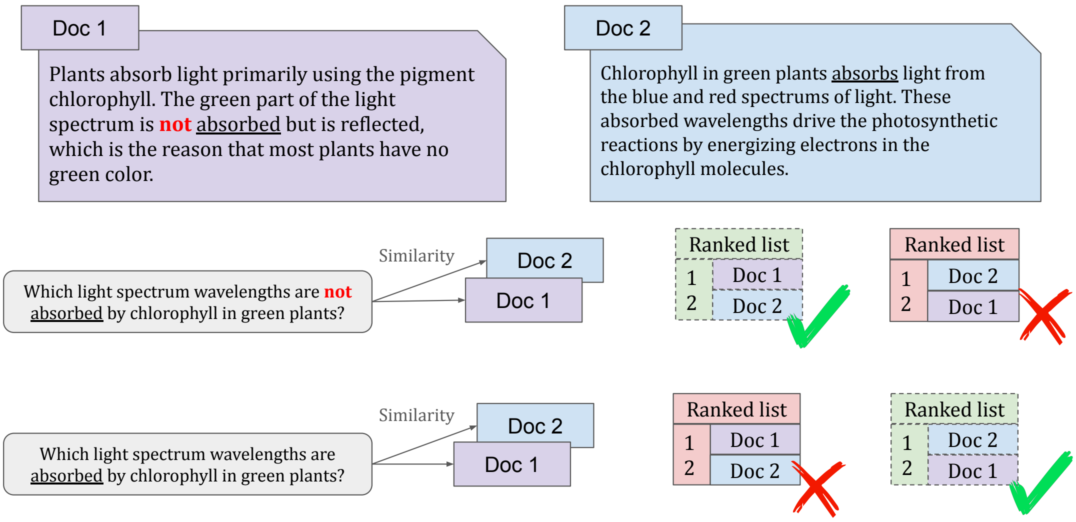

Figure 1: Example instance from our Free Generation dataset for sentential negation. Doc 1 is a passage retrieved from an existing Wikipedia article; Doc 2 is a minimally edited counterfactual whose truth value is flipped. The task is pairwise ranking. Given two queries that only differ in the presence of negation, the retrieval model must rank the corresponding document higher. The model succeeds if it ranks the correct document higher for both queries. There is a 25% random chance in pairwise accuracy.

2 Motivation

Negation has a long history in (computational) linguistics. The study of opposition and its expression in the form of negation is a phenomenon that has been debated by, and provoked interest from linguists, logicians, metaphysicians, and philosophers (Seiver, 1944; Horn, 1989; Kunen, 1987; Halpern and Pearl, 2005). It is a highly complex expression of thought given its apparent simple form (Horn, 1989). Other challenges are imposed by the ambiguity of the negation scope (Atlas, 1977), and pragmatic inferences in conversational settings (Schlöder and Fernández, 2015).

Proper treatment of negation is essential. Understanding negation is vital for retrieval models to provide the correct information to the user. Moreover, handling negation is vital to ensure that the retrieved generations are a correct response to the user query, since generated answers are particularly difficult to verify, as they cannot be grounded in established evidence (Wang et al., 2024). Equally important is ensuring that RAG systems respect user-specified negation and avoid retrieving information the user explicitly does not search for.

Fine-tuning on negation datasets. One could argue that this problem can be mitigated through fine-tuning (Dolci, 2022). However, catastrophic forgetting occurs when a model is fine-tuned on a new dataset (Hayes et al., 2019), even if its distribution is similar to the original training data. In certain cases, fine-tuning can lead to a degradation of performance in the original training set (Peters et al., 2019; Merchant et al., 2020). Model sensitivity to parameter adjustments is particularly noticeable in information retrieval settings. This has been observed in traditional BERT-based architectures (Gerritse et al., 2022) and LLMs (Soudani et al., 2024a). Although this behavior can be mitigated by freezing the model parameters and adding a language model head that is fine-tuned on a new dataset (Huang et al., 2022; Lin et al., 2022), this method restricts the capabilities the model can learn. Weller et al. (2024) shows that fine-tuning on their proposed dataset (NevIR) leads to a noticeable decline in MSMarco generalization performance. This could be explained by catastrophic forgetting and fine-tuning on an overly specialized dataset.

Representations of negation. Another explanation for models under-performing on negation is the lack of a specialized pre-training setup, which can arise from an under-representation of negation in popular pre-training datasets (Hossain et al., 2020), although LLMs have been trained on extensive crawled datasets and exhibit similar shortcomings. An improper training setup can also be caused by the training objective. While contrastive loss pushes representations that are different in content to be distant in the representation space, two negated statements are close in content while conveying opposite information. (Hosseini et al., 2021; Noji and Takamura, 2020) address the problem of having a misalignment between the training objective and the semantics of negation by proposing an 'unlikelihood' loss function used to further pretrain BERT on factually incorrect statements with negation cues.

3 Related Work

Negation in IR. Negation has been studied since early language models, e.g., Jumelet and Hupkes (2018) investigate the capabilities of LSTMs to locate the scope of negation, which they evaluate using a parse tree. Early work typically examines negation at the atomic sentence level. In contrast, negation in IR must be handled across pairs of queries and documents, as the presence of negation in a query can completely reverse the relevance of a document that otherwise is a semantic match. Therefore, IR systems must assess whether both the query and the document share the same polarity. i.e., positive or negative (McQuire and Eastman, 1998). Negation in IR often takes the form of exclusion, which involves filtering information, and rejection of suggestions, which involves dismissing information, as mentioned by Yaeger-Dror and Tottie (1993). Having distinct types of negation poses an added challenge to defining it in an IR context, which can therefore be difficult and ambiguous.

Negation in different modalities. Alhamoud et al. (2025) propose a benchmark for understanding negation across 18 tasks and modalities spanning image, video, and medical data. Their experiments reveal that even with large-scale training, modern vision language models (VLMs) struggle with negation, often performing at random. The authors show that fine-tuning on large-scale synthetic datasets can approach a 10% increase in performance. However, that forces the model to overfit on negation instead of making it reason on negation, as shown by achieving a good performance on one dataset but not generalizing on negation out of distribution (Zhang et al., 2020; Zhou and Srikumar, 2021).

Retrieval models and LLMs for retrieval. Information retrieval models evolved from lexical matching to dense retrieval, where the similarity between a query and documents is identified in a latent semantic space. These representations can be learned separately, i.e., with bi- and dual encoders, or together, i.e., with cross encoders. Dense models have been shown to outperform classical lexical matching in most scenarios (Karpukhin et al., 2020; Khattab and Zaharia, 2020). In addition, LLMs are being fine-tuned to serve as the backbone of retrieval and ranking tasks (Zhu et al., 2023), bringing a boost in performance through their rich representations. LLM-based models used for retrieval are constructed on small-scale models, such as BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020), or on larger-scale next token prediction models, such as Llama (Grattafiori et al., 2024), Mistral (Jiang, 2024) and Qwen (Yang et al., 2024).

Data generation using LLMs. Data generation using LLMs has gained significant attention (Abolghasemi et al., 2024; Askari et al., 2023; Tunstall et al., 2023; Abbasiantaeb et al., 2024; Liu et al., 2024), and has been shown to be a viable method to expand the training dataset, improving performance in several tasks such as dialog generation (Soudani et al., 2024b; Askari et al., 2025), reasoning (Yin et al., 2023), negation (Li et al., 2023) and exclusionary retrieval (Zhang et al., 2024a).

Existing negation datasets. One of the first forays into negation understanding was in the medical domain, where research focused on automatically indexing clinical reports and discharge summaries (Savova et al., 2010; Niu et al., 2005). For example, Bio-Scope (Zhu et al., 2019) is a corpus of biomedical text mining that focuses on extracting accurate information on biological relations. Today, in the IR literature, we have access to publicly available datasets such as NevIR (Weller et al., 2024), ExcluIR (Zhang et al., 2024a), BoolQuestions (Zhang et al., 2024b), Quest (Malaviya et al., 2023), and RomQA (Zhong et al., 2022). While these datasets contain logical operator annotations, the annotation system largely remains a single binary label for the presence of negation.

Research gap. How is a taxonomy different from linguistic formalisations of negation in logic? Aristotle transferred the study of negation from the domain of ontology to logic and language (Smith, 2022). The linguistic formalization of negation in logic defines how negation operates within formal systems (da Costa, 1974), such as in classical logic, where a proposition p is negated through $\neg p$ in which the truth value is flipped, or within modal and nonmonotonic logic (Ketsman and Koch, 2020), where it has more nuanced interpretations. In contrast, a taxonomy for negation would categorize different types and functions of negation in language and reasoning, such as lexical (Staliunaite and Iacobacci, 2020) vs. semantical (Urquhart, 1972) negation, metalinguistic (Horn, 1985) vs. descriptive (Miestamo, 2005; Lee, 2017), or negation as opposition (Mettinger, 1994) vs. absence (Faller, 2002). Although logic treats negation as a formal operation on truth values, a taxonomy explores its diverse roles in communication, cognition, and interpretation.

4 Methodology

We propose (1) a taxonomy for negation that is used to generate (2) two synthetic datasets that can be used for evaluating the performance of neural information retrieval models and for fine-tuning models to become more robust on negation, and (3) a classification mechanism that splits existing datasets into granular types of negation.

4.1 Taxonomy

We derive our negation taxonomy from definitions in logic, philosophy (Horn, 1989) and natural language processing literature (Yaeger-Dror and Tottie, 1993; McQuire and Eastman, 1998). Figure 2 presents the taxonomy as a hierarchical tree, where each node denotes a negation type and its child nodes correspond to finer-grained subtypes. Table 2 in Appendix A.2 includes query-document pairs exemplifying each negation type.

Our primary classification criterion is on the scope (see Appendix A.1), distinguishing explicit negation realized by a logical operator $\neg$ (Haegeman, 1995), from lexical negation that is present through the semantics of the word itself (Natayou, 2014). **Logical Operators** append to a word or clause, reversing its meaning. In **lexical** negation, a word or phrase inherently evokes negation, without the need for an appended operator.

We identify three types of logical operators. **Sentential** (Zeijlstra, 2004) negation is signalled by sentential operators such as *no*, *not* and *none*, which have a fixed syntactic role and occupy defined positions within a sentence. **Exclusion** (MacCartney and Manning, 2008) is signalled by exclusionary operators that are either **exceptors**, such as *besides* and *others* (exceptors represent a unique type of negation, see more in Appendix A.2), or **quantifiers**, such as the universal quantifier *for all* and the existential quantifier *exists*. In Aristotelian logic (Keenan and Westerståhl, 1997; Horn, 1989), these quantifiers define three fundamental relations: **Contradiction**, **Contraries**, and **Subcontradiction**. Finally, **Affixal** (Zimmer, 1966) negation is signalled by prefix and suffix operators that are preppended or appended to an existing word, such as: *un-*, *in-*, *im-*, *il-*, *ir-*, *dis-*, *non-*, *mis-*, *ill-*, *-less*, *-free* (Wahyuni, 2014).

We identify two types of lexical negation. **Implicit** (Madva, 2016) negation is composed of words that are inherently negative through their meaning, e.g.: *refuse*, *deny*, *exclude*, *reject*, *avoid*, *lack*, *fail*. **Contrasting** (Trillas, 2017) negation is composed of words that convey negation in pairs, but are not negative independently. These can be called contrasting pairs of antonyms. **Immediate** antonyms are opposite words with no degree of variation between them; **Polar** antonyms are opposite words with degrees of variation between them, and **Mid** antonyms represent samples from the interpolation of two polar antonyms. For more special cases of negation that we do not cover in this study, see Appendix A.4.

4.2 Data Generation

We generate two synthetic datasets designed to cover all negation types described in the taxonomy. We construct the datasets as follows: (1) we prompt the LLM to generate 100 *topics* of general knowledge to ensure familiarity and avoid long-tail knowledge (Askari et al., 2025); (2) for each topic, we ask the LLM to return one *Wikipedia page* that we check using the Wikipedia API, ensuring the generations are grounded in documented and factual information; (3) conditioned on a Wikipedia page, the LLM generates pairs (q_1, doc_1) and (q_2, doc_2) following the template of CondaQA (Ravichander et al., 2022) and NevIR (Weller et al., 2024). (3.1) Given detailed prompts constructed for the individual negation type, we ask the LLM to retrieve a paragraph that contains one specific negation as defined in the taxonomy. If the document does not contain explicit markers for the specified negation, the model will retrieve the closest match and rephrase it by injecting specific markers, i.e., keywords such as *impossible* instead of *not possible*. (3.2) Given the extracted paragraph, the LLM generates a query. This is the process of generating one pair (q_1, doc_1). (3.3) For generating the second pair, we employ two strategies to produce different degrees of lexical overlap between the negated datasets. (1) **Free Generation**: generate a positive query q_2 by removing the negation from q_1; generate a positive document doc_2 by answering q_2. (2) **Controlled Generation**: generate a positive query q_2 by removing the negation from q_1; generate a positive document doc_2 by removing the negation from doc_1. The two synthetically generated datasets have 1505 and 1479 instances, respectively, where a single instance has pairs (q_1, doc_1)

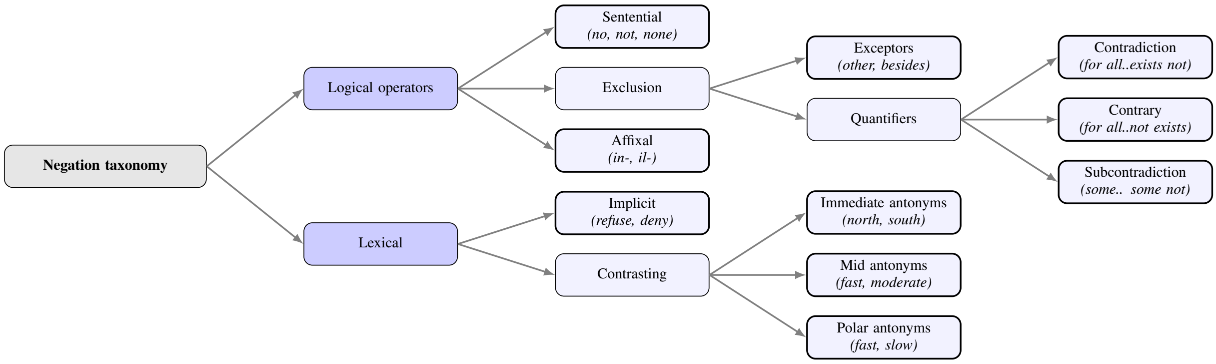

Figure 2: Negation taxonomy tree.

and (q_2, doc_2). Appendix A.3 provides the prompts used for generation, and an additional verification step for guaranteeing the relevance of documents; Table 3 and Figure 8 summarize the dataset statistics and distribution of generated labels.

4.3 LM Logic classification

Negation can be analysed at two granularities. **Sentence-level:** some negation types can be identified at the sentence level; if two sentences are either both negative or both positive, the pair agrees in polarity (Mahany et al., 2022), and if they do not, it conveys a negative polarity relationship (sentential, exclusionary, affixal, and implicit). **Pair-level:** the negation polarity can only be identified by comparison, i.e., whether both statements can be true at the same time (quantifiers and contrasting negation). We propose a classification mechanism that assigns each instance in an existing dataset a category outlined in our taxonomy by converting it to natural logic using typed lambda (λ) calculus formalisations (Barendregt, 1985) (see Appendix A.2). We generate formalisations for each instance by prompting a model with an instruction to generate the typed lambda calculus proof, and return the predicates, quantifiers and λ-typed formula, following the prompt illustrated in Appendix A.3. We categorize an existing dataset in four iterative steps:

Step 1: Predicate Classification We check the returned predicates. If any predicate defined in the deconstruction of the query is of sentential, exclusionary, affixal, or implicit nature (as classified by the LLM), we label the instance accordingly. Since they are sentence-level negations, we only study the queries.

Step 2: Quantifier Pattern Matching If no predicates are found, we analyse query and document pairs. We extract the logical quantifiers present in both the query and document (both pairs, see Appendix A.5), and check if any of the following logical patterns are identified as contradiction, contrary and subcontradition definitions (Horn, 1989): $(\forall \ldots \exists\neg)$, $(\forall \ldots \neg\exists)$, $(\exists \ldots \exists\neg)$. Instances matching any of these patterns are labelled accordingly.

Step 3: Semantic Antonyms Detection We will assume the only other potential negation is both at the semantic level and only detectable in paired interactions (in contrast, a predicate such as *refuse* inherently carries a negative polarity, whereas a predicate such as *slow* does not). We check such antonym pairs with the *nltk* library.

Step 4: Absence of Negation If none of the previous conditions are met, we conclude that the instance does not contain negation according to our taxonomy.

5 Experimental Setup

Throughout this study, we use the GPT-4o-mini model (OpenAI et al., 2024) to conduct experiments that aim to answer our research questions. More precisely, we evaluate retrieval models to reveal the necessity of our taxonomy-driven synthetic data, evaluate categorized existing datasets to show the usefulness of our logic-driven mechanism, and fine-tune to show that a coverage of negation types can help with generalisation.

Evaluation of the generation. We assess the quality of the generated datasets with human annotation on 5% of the generations, with two annotators evaluating each instance on: (1) relevance of documents to each query, (2) presence of negation, (3) naturalness, (4) coherence, and (5) consistency of information within the document. The annotation was conducted with LabelStudio.[1] We assess the annotations on quantitative and qualitative measures, together with the annotator agreement.

[1]https://labelstud.io/

Appendix A.6 illustrates the questions for the annotators alongside further details for the setup. Tables 4 and 5 report the annotation metrics. The main findings are as follows:

- Annotators reported 71-77 % accuracy for document relevance and 83% − 88% f1 score for negation presence.
- On a scale of 1-5, the annotators reported an approximate quality of 4 on naturalness, coherence, and consistency of language.
- The inner annotator agreement passed significance values for sentential and contrasting negation. For implicit and quantifiers, the test shows borderline agreement in language quality.
- The biggest disagreement was noticed in the exceptors.

Evaluation of the classification mechanism. We evaluate the quality of our classification mechanism by assessing it against the generated datasets, for which we have access to golden labels by design of construction: we generate data for each type of negation conditioned on a taxonomy-dependent prompt. We run the classification mechanism on the free generation dataset, and obtain a balanced accuracy score of 86.84% and an F1 score of 86.95%. We notice that around 54% of missclassifications are contrary negations missclassified as contradictions. In our experiments, all models perform similarly between these two types of negation, as they are logically and lexically very similar. Therefore, we assume it does not affect our study.

Retrieval Models. We study the performance of lexical, bi-encoder, crossencoder, late interaction and tranformer models trained for first-stage retrieval, ranking, sentence similarity, natural language inference (NLI) and next token prediction (NTP). We follow the experimental setup introduced by Elsen et al. (2025). We show the specifications of all models in Table 6 (Appendix A.7).

Metrics. The metric used to evaluate the task presented in Figure 1 is **pairwise accuracy**: for each instance queries q_1, q_2 and documents d_1, d_2, the model independently ranks $\{d_1, d_2\}$. The prediction is correct only when the system places d_1 above d_2 for q_1 and inverts the order for q_2. Random performance for pairwise accuracy is 25%.

Fine-tuning. We fine-tune three models: ColBERTv1, multi-qa-mpnet-base-dot-v1, and Mistral-7B-Instruct for 20 epochs on the free generated dataset and evaluate on NevIR (Weller et al., 2024) and MSMarco (Bajaj et al., 2016) dev data.

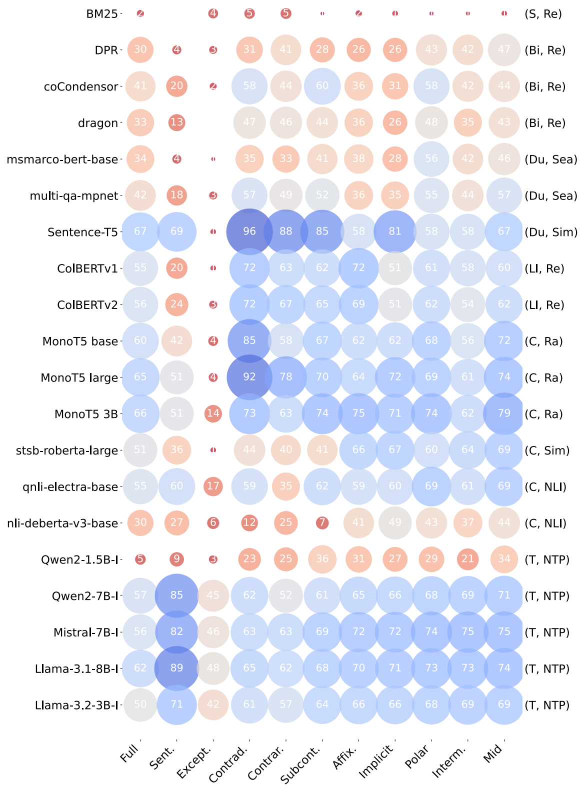

Figure 3: Pairwise Accuracy on the free generations dataset. The first result column contains the full dataset; later columns represent one negation type each. Models are represented by the rows, where **I** is a shortcut for **Instruct**. On the right, we assign labels expressing the architecture and training objective of each model: the first position shows the architecture, i.e., **S**parse, **Bi**-encoder, **Du**al encoder, **C**rossencoder, and **T**ransformer; the second position shows the training objective, i.e., **Re**trieval, **Sea**rch, **Sim**ilarity, **Ra**nking, **N**atural **L**anguage **I**nference, and **N**ext **T**oken **P**rediction. For a close-up, see Appendix A.8.

6 Results

Our experiments are designed to investigate the following hypotheses: (H1) some negation types are better encoded in the model internal representations than others, (H2) model specifics such as architecture, training objective, size and backbone significantly influence performance on negation, (H3) existing datasets have an uneven representation on negation, (H4) fine-tuning on our synthetically generated dataset will show systematic improvement in the downstream task presented in Figure 1.

6.1 Evaluation on Synthetic Data

Figure 3 illustrates 20 models evaluated on the free generation dataset. Sparse, dual, and biencoders exhibit poor performance on all types of negation, except Sentence-T5: a dual encoder trained for semantic similarity. Both late-interaction and all cross-encoder models, except nli-deberta-v3-base, show strong performance on all negation types. BERT and T5-based cross-encoders perform better than models with a RoBERTa, ELECTRA, and DeBERTa backbone. All transformer-based models, except for Qwen 1.5B (which has a disadvantage in size, and it has been trained for NTP) perform well on almost all negation types.

We perform a one-way ANOVA to test the significance of the results. ON model architecture, the ANOVA test reports a p-value of $1.0087e-11$, and the Tukey HSD shows a significant difference between sparse and dense models. When grouping on the training objective, ANOVA indicates $p = 1.5709e-04$, with significant differences between combinations of NTP, retrieval, and semantic search, and between sentence similarity vs. retrieval. The test shows a statistically significant difference between exceptors and all other types of negation. The experiments confirm hypothesis H1 and H2, that is, some negation types are better encoded than others, and that model specifics, such as architecture and training objective, influence performance. An analysis on the controlled generation dataset is illustrated in Figure 11 in Appendix A.8, where a similar behavior is seen; however, the patterns are even stronger, with a general trend toward higher performance. This can be inherent in the data generation process, i.e., document 2 is generated by changing the negation in document 1 (as compared to directly answering query 2).

6.2 Evaluation on Logic Filtered NevIR

When we apply the classification mechanism on the validation set of NevIR, we find that three main types of negation are present. Out of 225 pairs, $\{79, 54, 44\}$ correspond to $\{\text{Sentential}, \text{Affixal}, \text{Implicit}\}$, while 31 have been classified as not containing negation which we label as Others, while the remaining 17 pairs are spread across the other types of negation present in the taxonomy. These results are in line with hypothesis H3, which states that existing datasets have an uneven distribution of negation types.

Figure 4 shows that models perform worse on the NevIR dataset compared to our synthetically generated dataset. Sentence-T5 exhibits the best performance among bi- and dual-encoders. ColBERTv1 has a higher performance than ColBERTv2, and the MonoT5 models perform the best on all types of negation. Similarly to Figure 3, we notice that the performance in all models for sentential negation

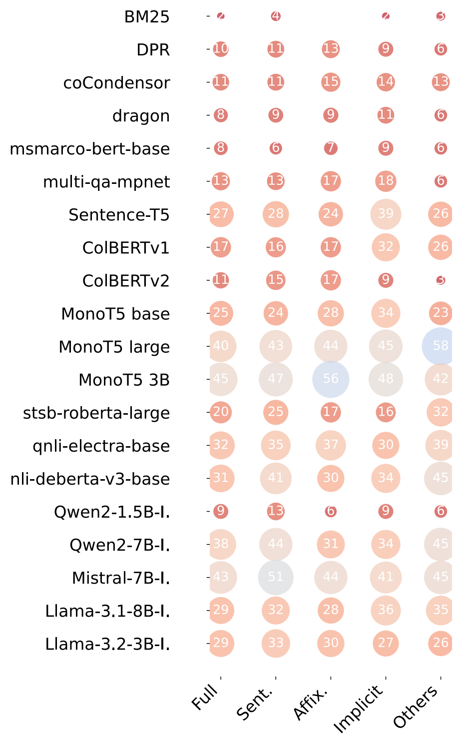

Figure 4: Pairwise Accuracy on NevIR as split with our classification mechanism.

is higher than affixal or implicit. Qwen2-1.5B performs the worst of all LLMs, similarly to synthetic experiments.

6.3 Evaluation on Logic Filtered ExcluIR

When applying the classification mechanism on the ExcluIR test set, we find three types of negation: $\{\text{Sentential}, \text{Exclusionary}, \text{Implicit}\}$ with $\{189, 2820, 113\}$ pairs out of 3452. Moreover, 297 have been classified as "Other" while 32 are distributed among the other classes. This means that more than 81% of the entire dataset has been classified as exclusionary. These results further support hypothesis H3.

As shown in Figure 12 (Appendix A.8), the performance of the model is approximately uniform between the three identified types of negation. This finding contradicts with out synthetic data experiments, where exclusionary negation was significantly more difficult to encode than the other types of negation. To further inspect the source of this discrepancy, we take a closer inspection of the ExcluIR instances identified as "Sentential" or "Implicit". This reveals that these instances only have a different rephrasing of a task that essentially is still exclusion. One example extracted from the dataset is *Can you tell me about Paul Ziert's involvement in founding the Bart Conner Gymnastics Academy in Norman, Oklahoma, while avoiding any mention* *of Bart Conner's role in the academy?*. Our categorization mechanism identifies this instance as "Implicit", while it has the form of a set subtraction, as per the definition of exceptors.

6.4 Fine-tuning

We fine-tune ColBERTv1, multiqa-mpnet-base-dot-v1, and Mistral-7B-Instruct on the free generation dataset, NevIR, and a mixed strategy with both datasets. We evaluate the finetuned models against NevIR dev set and MSMarco dev small.

Train partitions: The NevIR training set is composed of 1,896 triplets. The train partition of our synthetically generated dataset consists of 2,114 triplets. When fine-tuning mixed data, we have a total of 2,005 triplets.

Evaluation partitions: We evaluate against the dev partition of NevIR that has 450 triplets (2 triplets = 1 pair).

6.4.1 Evaluation on NevIR

As shown in Table 1 and in Figure 13 in Appendix A.8.1, fine-tuning **ColBERT** and **MultiQA** on our synthetic dataset yields an immediate performance gain on the NevIR development set, however peaking while fine-tuning on NevIR train reaches higher performance in the last epoch. This is to be expected as for the synthetic data we evaluate OOD. To assess in-distribution performance, we apply mixed fine-tuning by combining the two datasets and shuffling the data. The model achieves high performance significantly faster than when simply fine-tuned on NevIR, giving the overall best performance. **Mistral** shows the same behaviour with mixed fine-tuning. This supports hypothesis H4, that our synthetically generated dataset helps in capturing negation.

6.4.2 Evaluation on MSMarco

When evaluated against MSMarco (Table 1 and Figure 14 in Appendix A.8.1), we notice that the generalizability of **ColBERT** and **MultiQA** drops when fine-tuned on any dataset. Interestingly, **Mistral** displays a more stable fine-tuning process; however, adding synthetic data drops performance even further. Although MSMarco generalization is known to be negatively affected when models are fine-tuned out of distribution, our results show a trade-off: synthetic and mixed training helps generalisation in the negation domain, but it further harms generalisation on MSMarco.

		NevIR P.Acc. ↑			**MSMarco** MRR@10↑		
		E1	E6	E20	E1	E6	E20
ColBERT	NevIR	.17	.21	.43	**.37**	**.37**	**.34**
	Synth	.20	.34	.32	.36	.34	.31
	Mixed	**.21**	**.39**	**.44**	.37	.33	.31
MultiQA	NevIR	.16	.54	.48	**.35**	**.17**	**.06**
	Synth	.34	.45	.46	.33	.07	.03
	Mixed	**.38**	**.54**	**.55**	.26	.03	.01
Mistral	NevIR	.67	.71	.72	.53	.58	**.60**
	Synth	.57	.54	.50	**.59**	.55	.55
	Mixed	**.71**	**.71**	**.72**	.57	**.60**	.54

Table 1: Results for ColBERT, MultiQA and Mistral when trained on NevIR, Synth and Mixed data, and evaluated on NevIR and MSMarco. Columns E0, E1, E6, E20 represent epochs 0 (before backprop.), 1, 6 and 20; P. Acc. stands for pairwise accuracy, while MRR@10 for mean reciprocal rank at 10.

7 Conclusion

In this study, we propose a philosophy, logic and linguistic-grounded taxonomy for negation along two synthetic datasets that can be used for evaluating existing neural retrieval, ranking and LLM reranker models, and for fine-tuning models to increase their capabilities on negation. Through our study, we found that (1) cross-encoders and LLM rerankers are better at encoding negation, (2) NevIR and ExcluIR have a limited coverage of negation types, and (3) fine-tuning on our synthetic datasets helps performance in a negation domain.

These insights confirm that negation is a complex phenomenon and that a thorough taxonomy brings advantages as a starting point for generating fine-tuning data. The taxonomy-based classification of current datasets, together with model evaluation, shows that having a broad coverage of negation types is vital. Our fine-tuning experiments confirm that the synthetic datasets bring a performance boost; however, it also indicates that fine-tuning data might not be the sole factor behind model difficulty with negation. The training objective and architectural backbone play a big role in model performance performance. However, different training objectives are a promising direction for future work. Moreover, we propose investigating negation in a retrieval setting with a large corpora. Moreover, while generalization drops with fine-tuning, we propose investigating the training objective by applying reinforcement learning on negation with a small subset, similar to R1-Search (Jin et al., 2025).

Limitations

Our work proposes a new dataset for investigating negation and improving performance in a negation setting, and a filtering mechanism for studying existing datasets. However, there are certain limitations to our study. Our dataset is limited to a binary classification redefined as a pairwise ranking task, and therefore is not directly applicable to a ranking setting with a large corpus. Moreover, the data is generated using GPT-4o mini. While the faithfulness of information is not the direct scope of this paper, having a more controlled generation process would be beneficial. Lastly, a broader study on datasets such as BoolQuestions, RomQA and Quest would offer a more extensive study.

References

Zahra Abbasiantaeb, Yifei Yuan, Evangelos Kanoulas, and Mohammad Aliannejadi. 2024. Let the LLMs talk: Simulating human-to-human conversational QA via zero-shot LLM-to-LLM interactions. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 8–17.

Amin Abolghasemi, Zhaochun Ren, Arian Askari, Mohammad Aliannejadi, Maarten de Rijke, and Suzan Verberne. 2024. Cause: Counterfactual assessment of user satisfaction estimation in task-oriented dialogue systems. *arXiv preprint arXiv:2403.19056*.

Kumail Alhamoud, Shaden Alshammari, Yonglong Tian, Guohao Li, Philip Torr, Yoon Kim, and Marzyeh Ghassemi. 2025. Vision-language models do not understand negation. *arXiv preprint arXiv:2501.09425*.

Arian Askari, Mohammad Aliannejadi, Chuan Meng, Evangelos Kanoulas, and Suzan Verberne. 2023. Expand, highlight, generate: RL-driven document generation for passage reranking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10087–10099.

Arian Askari, Roxana Petcu, Chuan Meng, Mohammad Aliannejadi, Amin Abolghasemi, Evangelos Kanoulas, and Suzan Verberne. 2025. Self-seeding and multi-intent self-instructing LLMs for generating intent-aware information-seeking dialogs. *arXiv preprint arXiv:2402.11633*.

Jay David Atlas. 1977. Negation, ambiguity, and presupposition. *Linguistics and Philosophy*, 1(3):321–336.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Henk P. Barendregt. 1985. *The Lambda Calculus: Its Syntax and Semantics*. North-Holland.

Newton C. A. da Costa. 1974. On the theory of inconsistent formal systems. *Notre Dame J. Formal Log.*, 15:497–510.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Tommaso Dolci. 2022. Fine-tuning language models to mitigate gender bias in sentence encoders. *2022 IEEE Eighth International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 175–176.

Coen van den Elsen, Francien Barkhof, Thijmen Nijdam, Simon Lupart, and Mohammad Alliannejadi. 2025. Reproducing NevIR: Negation in neural information retrieval. *arXiv preprint arXiv:2502.13506*.

Martina Faller. 2002. *Semantics and Pragmatics of Evidentials in Cuzco Quechua*. Ph.D. thesis, Stanford University.

Luyu Gao and Jamie Callan. 2021. Unsupervised corpus aware language model pre-training for dense passage retrieval. *arXiv preprint arXiv:2108.05540*.

Emma J. Gerritse, Faegheh Hasibi, and Arjen P. de Vries. 2022. Entity-aware transformers for entity search. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 1455–1465. ACM.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Liliane Haegeman. 1995. *The Syntax of Negation*, volume 75. Cambridge University Press.

Joseph Y Halpern and Judea Pearl. 2005. Causes and explanations: A structural-model approach. Part I: Causes. *The British Journal for the Philosophy of Science*, 56:843–887.

Tyler L. Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. 2019. REMIND your neural network to prevent catastrophic forgetting. *arXiv preprint arXiv:1910.02509*.

Laurence R. Horn. 1985. Metalinguistic negation and pragmatic ambiguity. *Language*, 61:121–174.

Laurence R. Horn. 1989. *A Natural History of Negation*. University of Chicago Press.

Laurence R. Horn. 2010. Multiple negation in English and other languages. In *The Expression of Negation*, pages 111–148. De Gruyter Mouton Berlin, Boston.

Md Mosharaf Hossain, Venelin Kovatchev, Pranoy Dutta, Tiffany Kao, Elizabeth Wei, and Eduardo Blanco. 2020. An analysis of natural language inference benchmarks through the lens of negation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Arian Hosseini, Siva Reddy, Dzmitry Bahdanau, R Devon Hjelm, Alessandro Sordoni, and Aaron Courville. 2021. Understanding by understanding not: Modeling negation in language models. *arXiv preprint arXiv:2105.03519*.

Xiaoshui Huang, Sheng Li, Wentao Qu, Tong He, Yifan Zuo, and Wanli Ouyang. 2022. Frozen CLIP model is an efficient point cloud backbone. *arXiv preprint arXiv:2212.04098*.

Fengqing Jiang. 2024. Identifying and mitigating vulnerabilities in LLM-integrated applications. Master's thesis, University of Washington.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *Preprint*, arXiv:2503.09516.

Jaap Jumelet and Dieuwke Hupkes. 2018. In *Proceedings of the Workshop on Analyzing and Interpreting Neural Networks for NLP*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Edward L. Keenan and Dag Westerståhl. 1997. Generalized quantifiers in linguistics and logic. In *Handbook of Logic and Language*.

Bas Ketsman and Christoph E. Koch. 2020. Datalog with negation and monotonicity. In *International Conference on Database Theory*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–48.

Kenneth Kunen. 1987. Negation in logic programming. *The Journal of Logic Programming*, 4(4):289–308.

Chungmin Lee. 2017. Metalinguistic negation vs. descriptive negation: Among their kin and foes. In *The Pragmatics of Negation: Negative meanings, uses and discursive functions*. John Benjamins Publishing Company.

Judith Yue Li, Aren Jansen, Qingqing Huang, Joonseok Lee, Ravi Ganti, and Dima Kuzmin. 2023. MAQA: A multimodal QA benchmark for negation. *arXiv preprint arXiv:2301.03238*.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. *arXiv preprint arXiv:2302.07452*.

Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Y. Qiao, and Hongsheng Li. 2022. Frozen CLIP models are efficient video learners. In *European Conference on Computer Vision*.

Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. 2024. ChatQA: Building GPT-4 level conversational QA models. *arXiv preprint arXiv:arXiv:2401.10225*.

Bill MacCartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *International Conference on Computational Linguistics*.

Alex Madva. 2016. Why implicit attitudes are (probably) not beliefs. *Synthese*, 193:2659–2684.

Ahmed Mahany, Heba Khaled, Nouh Sabri Elmitwally, Naif Aljohani, and Said Ghoniemy. 2022. Negation and speculation in NLP: a survey, corpora, methods, and applications. *Applied Sciences*, 12(10):5209.

Chaitanya Malaviya, Peter Shaw, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2023. QUEST: A retrieval dataset of entity-seeking queries with implicit set operations. *arXiv preprint arXiv:2305.11694*.

Ian R. McKenzie, Alexander Lyzhov, Michael Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, Andrew Gritsevskiy, Daniel Wurgaft, Derik Kauffman, Gabriel Recchia, Jiacheng Liu, Joe Cavanagh, Max Weiss, Sicong Huang, The Floating Droid, and 8 others. 2024. Inverse scaling: When bigger isn't better. *arXiv preprint arXiv:2306.09479*.

April R. McQuire and Caroline M. Eastman. 1998. The ambiguity of negation in natural language queries to information retrieval systems. *J. Am. Soc. Inf. Sci.*, 49:686–692.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to BERT embeddings during fine-tuning? *arXiv preprint arXiv:2004.14448*.

Arthur Mettinger. 1994. *Aspects of Semantic Opposition in English*. Oxford University Press.

Matti Miestamo. 2005. *Standard Negation: The Negation of Declarative Verbal Main Clauses in a Typological Perspective*. De Gruyter Mouton.

Roser Morante and Walter Daelemans. 2012. ConanDoyle-Neg: Annotation of negation cues and their scope in Conan Doyle stories. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, Turkey, May 23–25, 2012*, pages 1563–1568. European Language Resources Association (ELRA).

Rosyane Florine Natayou. 2014. *Explicit and Implicit Means of Negation in the English Language*. Ph.D. thesis, Sumy State University.

Yun Niu, Xiao-Dan Zhu, Jianhua Li, and Graeme Hirst. 2005. Analysis of polarity information in medical text. In *AMIA Annual Symposium Proceedings*, pages 570–574.

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.

Hiroshi Noji and Hiroya Takamura. 2020. An analysis of the utility of explicit negative examples to improve the syntactic abilities of neural language models. *arXiv preprint arXiv:2004.02451*.

OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, and many others. 2024. GPT-4o system card. *arXiv preprint arXiv:2410.21276*.

Lourdes Ortega, Andrea Tyler, Hae In Park, and Mariko Uno. 2016. *The Usage-based Study of Language Learning and Multilingualism*. Georgetown University Press.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? Adapting pretrained representations to diverse tasks. *arXiv preprint arXiv:1903.05987*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Abhilasha Ravichander, Matt Gardner, and Ana Marasović. 2022. CONDAQA: A contrastive reading comprehension dataset for reasoning about negation. *arXiv preprint arXiv:2211.00295*.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2021. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. *arXiv preprint* arXiv:2112.01488.

Guergana K. Savova, James J. Masanz, Philip V. Ogren, Jiaping Zheng, Sunghwan Sohn, Karin Kipper Schuler, and Christopher G. Chute. 2010. Mayo clinical text analysis and knowledge extraction system (cTAKES): Architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513.

Julian J Schlöder and Raquel Fernández. 2015. Pragmatic rejection. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 250–260.

George O Seiver. 1944. Cicero's de oratore and rabelais. *PMLA*, 59(3):655–671.

Robin Smith. 2022. Aristotle's Logic. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*, Winter 2022 edition. Metaphysics Research Lab, Stanford University.

Heydar Soudani, Evangelos Kanoulas, and Faegheh Hasibi. 2024a. Fine tuning vs. retrieval augmented generation for less popular knowledge. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, SIGIR-AP 2024, Tokyo, Japan, December 9-12, 2024*, pages 12–22. ACM.

Heydar Soudani, Roxana Petcu, Evangelos Kanoulas, and Faegheh Hasibi. 2024b. A survey on recent advances in conversational data generation. *arXiv preprint arXiv:2405.13003*.

Ieva Staliunaite and Ignacio Iacobacci. 2020. Compositional and lexical semantics in RoBERTa, BERT and DistilBERT: A case study on CoQA. *arXiv preprint arXiv:2009.08257*.

Enric Trillas. 2017. Antonyms. negation, and the fuzzy case. In *On the Logos: A Naïve View on Ordinary Reasoning and Fuzzy Logic*, pages 25–34.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, and 1 others. 2023. Zephyr: Direct distillation of LM alignment. *arXiv preprint arXiv:2310.16944*.

Alasdair Urquhart. 1972. Semantics for relevant logics. *Journal of Symbolic Logic*, 37:159–169.

Sri Wahyuni. 2014. An analysis on affixal negation in English. S1 Thesis. University of Mataram.

Yuxia Wang, Minghan Wang, Muhammad Arslan Manzoor, Fei Liu, Georgi Georgiev, Rocktim Jyoti Das, and Preslav Nakov. 2024. Factuality of large language models: A survey. *arXiv preprint arxiv:2402.02420*.

Orion Weller, Dawn Lawrie, and Benjamin Van Durme. 2024. NevIR: Negation in neural information retrieval. *arXiv preprint arXiv:2305.07614*.

Malcah Yaeger-Dror and Gunnel Tottie. 1993. Negation in english speech and writing: A study in variation. *Language*, 69:590.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Xunjian Yin, Baizhou Huang, and Xiaojun Wan. 2023. Alcuna: Large language models meet new knowledge. *Preprint*, arXiv:2310.14820.

Hedde Zeijlstra. 2004. *Sentential Negation and Negative Concord*. Ph.D. thesis, LOT.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2020. Revisiting few-sample bert fine-tuning. *arXiv preprint arXiv:2006.05987*.

Wenhao Zhang, Mengqi Zhang, Shiguang Wu, Jiahuan Pei, Zhaochun Ren, Maarten de Rijke, Zhumin Chen, and Pengjie Ren. 2024a. ExcluIR: Exclusionary neural information retrieval. *arXiv preprint arXiv:2404.17288*.

Zongmeng Zhang, Jinhua Zhu, Wen gang Zhou, Xiang Qi, Peng Zhang, and Houqiang Li. 2024b. BoolQuestions: Does dense retrieval understand boolean logic in language? In *Conference on Empirical Methods in Natural Language Processing*.

Victor Zhong, Weijia Shi, Wen tau Yih, and Luke Zettlemoyer. 2022. RoMQA: A benchmark for robust, multi-evidence, multi-answer question answering. *arXiv preprint arXiv:2210.14353*.

Yichu Zhou and Vivek Srikumar. 2021. A closer look at how fine-tuning changes BERT. *arXiv preprint arXiv:2106.14282*.

Yanjie Zhu, Yuanyuan Liu, Leslie Ying, Xin Liu, Hairong Zheng, and Dong Liang. 2019. Bio-scope: fast biexponential t1ρ mapping of the brain using signal-compensated low-rank plus sparse matrix decomposition. *Magnetic Resonance in Medicine*, 83:2092 – 2106.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*.

Karl E. Zimmer. 1966. Affixal negation in english and other languages: An investigation of restricted productivity. *Language*, 42:134.

Arianna Zuanazzi, Pablo Ripollés, Wy Ming Lin, Laura Gwilliams, Jean-Rémi King, and David Poeppel. 2023. Tracking the behavioral and neural dynamics of semantic representations through negation. *bioRxiv*.

A Appendix

This appendix offers further material that supports the study. It is organised as follows: Appendix A.1 defines the properties of negation that are briefly referenced in the study. Appendix A.2 gives an example in an information retrieval style for each type of negation present in the taxonomy, alongside further definitions of exceptors and typed lambda calculus. Appendix A.3 lists all the prompts used to generate the datasets. Appendix A.4 mentions use cases that we do not explicitly account for in this study, although they are interesting to study. A.5 lists details into applying the categorization mechanism on the ExcluIR dataset. Appendix A.6 includes the survey that the human annotators completed to perform a qualitative evaluation of the generated data. Appendix A.7 contains a table with specifications of all models used in this study: architecture, training objective, training dataset, size and tokenizer - these specifications influence the model's performance on encoding negation. Appendix A.8 contains the results of evaluating the models against the controlled generated dataset and the ExcluIR data. Finally, Appendix A.6.1 offers a statistical analysis of the annotator's answers.

A.1 Negation Properties

Drawing inspiration from Morante and Daelemans (2012), we define the following properties of negation:

- **Negation cues**: Negation cues can be single words, multiwords, prefixes, such as im-, or suffixes, such as -less. They introduce the negation in the sentence.

  Example: *She did **not** go to the movies, but went to the theater instead.*

- **Negated event**: The main event or property that is being negated. For example, if we define $\neg$ as a negation operation, i.e. $\neg A$, then A is the negated event.

  Example: *She did **not** <u>go</u> to the movies, but went to the theater instead.*

- **Negated scope**: Extension of the negated event; part of the sentence where the negation propagates and changes its semantics. The parts of the sentence that are not affected by negation should be left out the scope.

  Example: *She did **not** <u>go to the movies,</u> but went to the theater instead.*

A.2 Taxonomy

In this section, we give a definition of exceptors using set operations, supporting our claim that exceptors are inherently a different type of negation compared to the rest of the taxonomy. This difference might influence how models perform on this negation type. We also give a definition of typed lambda calculus. Moreover, we provide examples for each negation type present in the taxonomy in the movie domain to exemplify the negation types in a retrieval setting. The examples are illustrated in Table 2.

Exceptors represent a unique type of negation. While the other negation types take the form of opposition, i.e., two propositions p and $\neg p$ cannot be true at the same time, exceptions are a form of set subtraction. More precisely, if we denote a domain S = {all candidate answers}, an exception set $E \subseteq S = \{\text{items to exclude}\}$ and an exclusionary query $Q_{\text{ex}} = S \setminus E$, then any document D that satisfies the exclusionary query Q_{ex} will inherently satisfy the whole set S as a consequence of $S \setminus E \ \subseteq S$.

Typed lambda calculus is a formal system that decomposes any statement into a logic form, by defining abstract predicates and determiners, either assuming their truth value, or reaching unit clauses that can only be True or only False (reaching a contradiction). The primary goal of typed lambda calculus is to provide a framework for meaning composition with flexible functions (predicates and determiners).

A.3 Data Generation

In this section, we show the prompts used for generating the synthetic datasets for free and controlled generation. We illustrate the prompt for generating sentential negation in Figure 5. The prompts for generating exceptors, affixal and implicit negation are similar, where only steps 1 and 2 are different. We illustrate steps 1 and 2 for each of these negation types in Figure 7. The prompts for contrasting clauses and quantifiers are shown in Figure 6.

Scope	Negation category	Negation subcategory	Aristotelian logic	Examples	Level
Logical operators	Sentential *(no, not, none)*			Q: Movies that **do not** feature Tom Hanks. D: Forrest Gump features Tom Hanks.	Sentence
	Exclusion	Exceptors *(others, besides but, except)*		Q: Movies with Tom Hanks **besides** Forrest Gump. D: Forrest Gump is a widely acclaimed movie.	Sentence
		Quantifiers	Contradiction	Q: What are **all** movies with Tom Hanks? D: Here are **some** movies **without** Tom Hanks..	Pair
			Contrary	Q: What are **all** movies with Tom Hanks? D: There **exist no** movies with Tom Hanks.	Pair
			Subcontradiction	Q: What are **some** movies with Tom Hanks? D: Here are **some** movies **without** Tom Hanks.	Pair
	Affixal			Q: What are some movies with **un**happy endings? D: These movies have happy endings.	Sentence
Lexical	Implicit			Q: Are there any movies with Tom Hanks that **failed** people's expectations?. D: This movie succeeded in public's eye.	Sentence
	Contrasting	Immediate Antonyms		Q: A movie that is **professional**. D: This is a **casual** movie.	Pair
		Mid Antonyms		Q: Movie where Tom Hanks is running very **fast**. D: In this movie, Tom Hanks runs **moderately paced**.	Pair
		Polar Antonyms		Q: Movie where Tom Hanks is running very **fast**. D: In this movie, Tom Hanks runs very **slow**.	Pair

Table 2: The proposed taxonomy of negation categories and their formalization.

Prompt for Sentential Negation

You are a system that receives a document. I want you to follow the next four steps:

1. Generate a search query that contains exactly **one** negation word ('no', 'not', or 'none').
   It should **not** be accompanied by a quantifier.
   The query **must be well-defined and have a finite, verifiable answer** even outside the document.
   Avoid queries that could have an **infinite, unbounded or exhaustive** number of answers.
   Also, avoid queries that have the answer 'yes' or 'no'.
   The query must be specific, and sound like something someone would type into a search engine.
2. Extract a short **retrieval-style passage** that contains exactly **one** negation word ('no', 'not', or 'none').
   - If the passage **does not contain** a negation, add exactly **one** negation word ('no', 'not', or 'none').
3. Generate the positive version of the search query by removing the negation.
4. Generate the positive version of the passage by removing the negation. Keep the other words intact.
5. Respond in JSON format.

Figure 5: Prompts for Sentential Negation

Prompt for Contrasting Clauses You are a system that receives a document. I want you to follow the next four steps.
Given the following definitions of types of antonyms:

- **Polar antonyms**: Words with absolute, direct opposite meaning with no other words between them.
- **Mid antonyms**: Words differing slightly, not completely opposed.
- **Intermediate antonyms**: Words with absolute, direct opposite meanings, with mid antonyms between them.

Pick a pair of mid antonyms that match this document. Name them `word1` and `word2`. Avoid antonyms that have a prefix.

1. Generate a search query that contains `word1`. The query **must be well-defined and have a finite, verifiable answer** even outside the document. Avoid queries that could have an **infinite or unbounded** number of answers. The query must be specific and sound like something someone would type into a search engine.
2. Extract a short **retrieval-style passage** that answers the query and **must** contain `word1`.
3. Generate the positive version of the search query by switching `word1` with `word2`.
4. Generate the positive version of the passage by switching `word1` with `word2`.

Respond in JSON format.

Prompt for Quantifiers
You are a system that receives a document. I want you to follow the next four steps. Generate one query. Then, re-write it in the following styles. Make sure all queries have exactly the same content:

1. The first search query must use exactly one **universal** quantifier ($\forall$).
2. The second search query must use exactly one **existential** quantifier ($\exists$), **followed by** a negation inside its scope ($\exists x \neg P(x)$). Do not use the word 'false'.
3. The third search query must use exactly one negation, **followed by** an **existential** quantifier ($\exists$) ($\neg \exists x P(x)$). Do not use the word 'false'
4. The fourth search query must use exactly one **existential** quantifier ($\exists$), such as "some". All queries **must be well-defined and have a finite, verifiable answer**. Avoid queries that could have an **infinite or unbounded** number of answers. The queries must be specific, and sound like something someone would type into a search engine. Do not use any symbols.Extract a short **retrieval-style passage** that answers the first query. Then, re-write it in the following styles:
5. The first passage must contain exactly one **universal** quantifier ($\forall$).
6. The second passage must contain exactly one **existential** quantifier ($\exists$), **followed by** a negation inside its scope ($\exists x \neg P(x)$). Do not use the word 'false'.
7. The third passage must contain exactly one negation, **followed by** an **existential** quantifier ($\exists$) ($\neg \exists x P(x)$).
8. The fourth passage must contain exactly one **existential quantifier** ($\exists$), such as 'some'.
9. "Respond in JSON format."

Figure 6: Prompts for Contrasting Clauses and Quantifiers

Extra Verification for the generated instances. After generation, we filter the instances by prompting the LLM to check the relevance of the documents for the queries. We only keep the instances for which both pairs pass the relevance self-check. This verification step is needed as sometimes the generated queries are too general, making the retrieved document not highly relevant.

Label Distribution. Figure 8 illustrates the distribution of negation types per synthetic dataset after the extra verification step. We notice that out of the generations, the sentential negations have been filtered the most.

Statistics of the generated datasets. Table 3 illustrates a summary of the two generated datasets, i.e., the free and controlled generation datasets. Length is calculated wrt. the number of words, while Data Size refers to the number of instances, where one instance is composed of pairs $<q_1, doc_1>$ and $<q_2, doc_2>$.

Statistics	Free Gen.	Contr. Gen.
Data Size	1505	1479
Query1 length	10.25	10.20
Query2 length	10.82	10.60
Doc1 length	36.65	36.48
Doc2 length	33.35	33.26

Table 3: Statistics of the two generated datasets. Free Gen. stands for free generation dataset, while Controlled Gen. stands for controlled generation dataset.

Variant	Differences in Step 1 and Step 2
Sentential	Step 1: Generate a query that contains exactly **one** negation word ('no', 'not', or 'none'). It should **not** be accompanied by a quantifier. The query **must be well-defined and have a finite, verifiable answer** even outside the document. Avoid queries that could have an **infinite, unbounded or exhaustive** number of answers. Also, avoid queries that have the answer 'yes' or 'no'. The query must be specific, and sound like something someone would type into a search engine. Step 2: Extract a short **retrieval-style passage** that contains exactly **one** negation word ('no', 'not', or 'none'). - If the passage **does not contain** a negation, add exactly **one** negation word ('no', 'not', or 'none').
Exceptor	Step 1: Generate a search query that contains exactly **one** exclusionary word such as ('others', 'besides', 'but', or 'except'). The query **must be well-defined and have a finite, verifiable answer** even outside the document. Avoid queries that could have an **infinite or unbounded** number of answers. The query must be specific, and sound like something someone would type into a search engine. Step 2: Extract a short **retrieval-style passage** that answers the query. Make sure the passage does **not** contain an exclusionary word such as ('others', 'besides', 'but', or 'except'). Make sure the passage also contains the excluded part from the query.
Affixal	Step 1: Generate a search query that contains exactly **one** affixal negation such as ('un-', 'in-', 'im-', 'il-', 'ir-', 'dis-', 'non-', 'mis-', 'ill-'). An affixal negation adds a prefix or suffix to reverse the meaning of a word. The query should not contain any other negation. The query **must be well-defined and have a finite, verifiable answer** even outside the document. Avoid queries that could have an **infinite or unbounded** number of answers. The query must be specific, and sound like something someone would type into a search engine. Step 2: Extract a short **retrieval-style passage** that answers the query. - In answering the query, the passage must contain exactly **the same** affixal negation as in the query. - If the passage **does not contain** an affixal word, add exactly **the same** one as in the query. The passage should not contain any other negation.
Implicit	Step 1: Generate a search query that contains exactly **one** implicit negation. An implicit negation is one that does not contain a negation operator. The word itself has negative semantics. Examples are ('avoid', 'refuse', 'deny', 'ignore'). It does **not** include affixal negations. The query should not contain any other negation. The query **must be well-defined and have a finite, verifiable answer** even outside the document. Avoid queries that could have an **infinite or unbounded** number of answers. The query must be specific, and sound like something someone would type into a search engine. Step 2: Extract a short **retrieval-style passage** that answers the query. - In answering the query, the passage must contain exactly **the same** implicit negation as in the query. - If the passage **does not contain** the implicit negation, add it yourself. The passage should not contain any other negation.

Figure 7: Summary of differences in prompt variants for different types of negation.

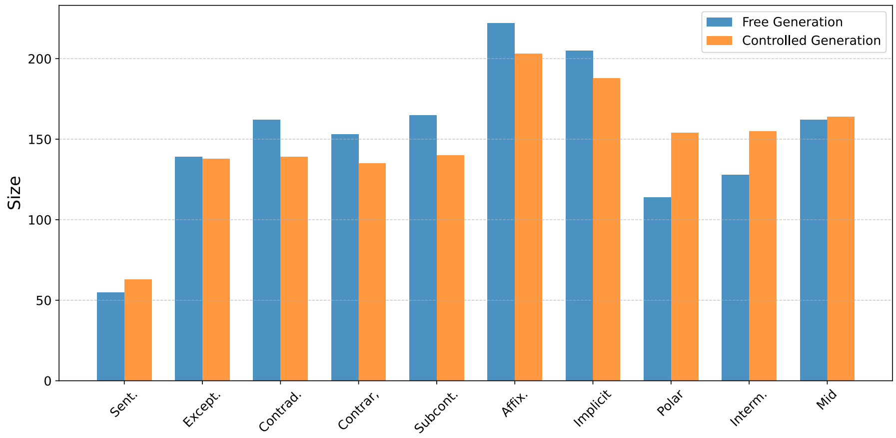

Figure 8: Distribution of negation types.

A.4 What we do not cover

This section contains negation phenomena and properties that, while interesting, we do not account for in this study.

In scope non-negated events. These are examples of events that are not negated, despite being within the scope of a negation Morante and Daelemans (2012). Examples are shown below. We exclude these cases from our study.

- I should be glad to be able to say afterwards that I had solved it without [your help].
- I call it luck, but [it would] not [have come my way had I not been looking out for it].
- I call it luck, but it would not have come my way [had I] not [been looking out for it].

Scope analysis. We also exclude analysis on the scope of the negation. In a sense, a query can be "Restaurants that do not serve food" and the returned document is "Restaurants that do not wash laundry". To maintain our study's focus, we do not delve into scope considerations. Moreover, the scope of negation can often shift according to context. For example, negation can have outer-read and inner-reading, for example "It is not likely that the Yankees will win.":

- outer-reading: (Likely...) as in, it is not probable that it will happen that the Yankees will win. $\neg\exists$
- inner-reading: Likely ... as in, it is likely the Yankees will not win. $\exists\neg x$

Litotes. Double negation does not always reduce to x, i.e., not not x does not necessarily mean x (Horn, 2010). Such figure of speech is called a litotes, where an understatement is made by adding a negative. Example can be:

- *I don't dislike cars.* ($\neg\forall\neg x = \exists\neg\neg x = \exists x$) can be seen as an understatement of *I like cars.* ($\forall x$)
- *Not bad!* is an understatement of *Good!*

Existential quantifiers with different scopes. Quantifiers such as "every" and "some" apply different scopes: Every man didn't win. Some man didn't win. $\forall x(\text{Man}(x) \rightarrow \neg W(x))$ and $\exists x(\text{Man}(x) \wedge \neg W(x))$.

A.5 LM Logic classification

When applying the typed lambda calculus formalization categorization, we check both pairs (q_1, doc_2) and (q_2, doc_1) for the presence of negation, as a result of not knowing necessarily where negation is present. For example, NevIR is constructed such that negation is always present in the first pair, while ExcluIR is constructed such that negation is always present in the second pair. Our classification mechanism is robust to such variations.

System Prompt

1. You are a Montagovian semanticist working in a typed λ-calculus framework.

2. For each **input query**, follow the next four steps:
   1. **LEXICON**: List every predicate and quantifier as a λ-term with an explicit Church type annotation.
   2. **SEMANTIC INVENTORY**: Output two comma-separated lists:
      - Predicates: `[]`
      - Quantifiers: [$\exists, \forall$]
   3. **NEGATION ANALYSIS**: For each predicate, indicate whether it matches one of the following categories:
      - Sentential (e.g. *no*, *not*, *none*, *never*, *cannot*)
      - Exclusionary (e.g. *besides*, *except*, *but*)
      - Affixal (e.g. bound morphemes *im-*, *in-*, *un-*, *-less*, etc.)
      - Implicit (e.g. verbs such as *deny*, *refuse*, *avoid*, *fail*)
   4. **FINAL FORMULA**: Present the fully reduced λ-term for S, or an equivalent first- or higher-order logic formula, enclosed in a fenced code block.

3. Respond in JSON format.

4. **Example:**

   Query:
   What organisms besides cyanobacteria perform anoxygenic photosynthesis?
   LEXICON:
   organism: $\lambda x : e. \, \mathrm{Organism}(x)$,
   cyanobacteria: $\lambda x. \, \mathrm{Cyanobacteria}(x)$,
   perform_anoxygenic_photosynthesis: $\lambda x. \, \mathrm{PerformAnoxygenicPhotosynthesis}(x)$,
   besides: $\lambda P \, Q \, x. \, Q(x) \wedge \neg P(x)$
   SEMANTIC INVENTORY:
   Predicates: [Organism, Cyanobacteria, PerformAnoxygenicPhotosynthesis], Quantifiers: [$\exists$]
   NEGATION ANALYSIS:
   Sentential: [], Exclusionary: [besides], Affixal: [], Implicit: []
   FINAL FORMULA:
   $\lambda x \colon e. \, \mathrm{Organism}(x) \;\wedge\; \mathrm{PerformAnoxygenicPhotosynthesis}(x) \;\wedge\; \neg \, \mathrm{Cyanobacteria}(x)$

Figure 9: Prompt for generating typed lambda calculus proofs.

A.6 Annotators Template

The queries and documents have been shuffled within the instance, and the sample used for annotations has a perfectly balanced distribution of labels. Given an instance (q_1, doc_1) and (q_2, doc_2), we ask the following questions to the annotators:

Q1: Which document is more relevant for q1?

- doc1
- doc2
- none
- both

Q2: Which document is more relevant for q2?

- doc1
- doc2
- none
- both

Q3: Which instances contain negation? Multiple choices are possible.
NOTE: If the individual instances do not contain negation, but the pair (q1, q2) contains antonyms, check both q1 and q2. Same goes for (doc1, doc2).

◊ q1
◊ q2
◊ doc1
◊ doc2

Q4: Rate the naturalness (fluency and readability) of the text.

1: Text is forced
2: Noticeably awkward
3: Minor issues
4: Language flows well
5: Perfectly polished

Q5: Rate the coherence (logical flow) of the text.

1: No logical flow [e]
2: Significant logical gaps
3: Basic logical structure
4: Generally logical and clear
5: Completely logical and clear

Q6: Rate the consistency of information in the text.

1: Contradictory
2: Unstable
3: Mixed
4: Aligned
5: Fully Aligned

A.6.1 Statistical analysis on annotation results

Table 4 shows the performance of annotators with respect to the ground truth labels of the generated datasets, i.e., averaged over both the free and controlled generation datasets. The rows q1-q6 indicate the six questions presented to the annotators, and the columns T1-T10 present the results of their answers split across the ten types of negation present in the sample shown to the annotators. For a brief description of the questions: q1-q2 ask about the relevance of the two documents for each query, and are assessed through accuracy; q3 asks about the presence of negation in the generation (binary question; therefore, it does not ask about the specific *type* of negation) and is assessed using the f1 score; q4-a6 are questions about the logic, naturalness, and consistency of information in the generated queries and documents, and are assessed by taking an average of the answers represented on an ordinal scale from 1-5.

Table 5 shows the inner agreement of the annotators when answering the questions wrt. the two generated datasets, i.e., averaged over both the free and controlled generation datasets. The rows q1-q6 indicate the six questions presented to the annotators, and the columns T1-T10 present the results of their answers split across the ten types of negation present in the sample shown to the annotators. For a brief description of the questions: q1-q2 ask about the relevance of the two documents for each query, and the agreement is measured using Cohen's Kappa; q3 asks about the presence of negation in the generation (binary question; therefore, it does not ask about the specific *type* of negation) and is assessed using recall of agreement; q4-a6 are questions about the logic, naturalness, and consistency of information in the generated queries and documents, and are assessed using a weighted Cohen's Kappa, given the answers represent an ordinal scale from 1-5. The scores are averaged across the two datasets.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
q1	0.79 ± 0.21	0.64 ± 0.21	0.79 ± 0.07	0.71 ± 0.14	0.86 ± 0.00	0.79 ± 0.07	0.79 ± 0.07	0.79 ± 0.07	0.79 ± 0.07	0.64 ± 0.21
q2	0.79 ± 0.07	0.21 ± 0.07	0.93 ± 0.07	0.71 ± 0.00	0.79 ± 0.07	0.79 ± 0.07	0.71 ± 0.00	0.79 ± 0.07	0.79 ± 0.07	0.57 ± 0.14
q3	0.91 ± 0.04	1.00 ± 0.00	0.90 ± 0.04	0.96 ± 0.03	0.94 ± 0.01	0.87 ± 0.03	0.90 ± 0.08	0.81 ± 0.00	0.77 ± 0.14	0.69 ± 0.07
q4	3.86 ± 0.00	3.71 ± 0.37	4.29 ± 0.57	3.79 ± 0.21	4.21 ± 0.21	4.29 ± 0.14	4.07 ± 0.18	4.36 ± 0.07	4.21 ± 0.07	4.29 ± 0.29
q5	3.86 ± 0.14	4.21 ± 0.24	4.07 ± 0.36	3.57 ± 0.14	4.14 ± 0.00	4.29 ± 0.14	4.14 ± 0.14	4.29 ± 0.00	4.21 ± 0.21	4.07 ± 0.21
q6	3.86 ± 0.29	4.21 ± 0.26	4.50 ± 0.50	4.57 ± 0.14	4.29 ± 0.00	3.71 ± 0.57	3.79 ± 0.36	4.50 ± 0.36	3.79 ± 0.79	3.93 ± 0.36

Table 4: Performance of annotators with respect to the ground truth labels on the generated query-document pairs of both synthetically generated documents. Each score represents a mean with an std. error over the two datasets.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
q1	0.60 ± 0.02	0.26 ± 0.17	0.89 ± 0.11	0.58 ± 0.18	0.52 ± 0.20	0.65 ± 0.35	0.52 ± 0.12	0.90 ± 0.11	0.53 ± 0.01	0.56 ± 0.03
q2	0.58 ± 0.02	0.30 ± 0.02	0.86 ± 0.14	0.53 ± 0.01	0.89 ± 0.11	0.57 ± 0.21	0.31 ± 0.20	0.90 ± 0.11	0.55 ± 0.02	0.58 ± 0.22
q3	0.78 ± 0.11	1.00 ± 0.00	0.93 ± 0.01	1.00 ± 0.00	0.92 ± 0.08	0.74 ± 0.16	0.67 ± 0.08	0.85 ± 0.05	0.87 ± 0.13	0.87 ± 0.02
q4	0.80 ± 0.01	0.30 ± 0.20	0.71 ± 0.29	0.52 ± 0.08	0.79 ± 0.21	0.79 ± 0.21	0.49 ± 0.14	0.76 ± 0.24	0.76 ± 0.04	0.89 ± 0.11
q5	0.75 ± 0.26	0.30 ± 0.20	0.68 ± 0.32	0.63 ± 0.37	0.89 ± 0.11	0.76 ± 0.02	0.69 ± 0.10	0.64 ± 0.09	0.71 ± 0.29	0.37 ± 0.01
q6	0.55 ± 0.02	0.36 ± 0.30	0.67 ± 0.05	0.36 ± 0.36	0.33 ± 0.40	0.44 ± 0.28	0.31 ± 0.13	0.78 ± 0.22	0.56 ± 0.20	0.56 ± 0.22

Table 5: Inner Agreement of annotators on their answers about the generated query-document pairs of both synthetically generated documents. Each score represents a mean with an std. error over the two datasets.

A.7 Models

Model	Architecture	Training objective	Training dataset	Size	Tokenizer
BM25	Sparse	Retrieval	N/A	N/A	N/A
DPR [29]	Bi-Encoder	Retrieval	NQ	219M	BERT
coCondenser [14]	Bi-Encoder	Retrieval	MSMarco	110M	BERT
Dragon [36]	Bi-Encoder	Retrieval	MS MARCO	N/A	BERT
msmarco-bert-base-dot-v5	Dual Encoder	Semantic Search	MSMarco	110M	BERT
multi-qa-mpnet-base-dot-v1	Dual Encoder	Semantic Search	QA	110M	MPNet
Sentence-T5	Dual Encoder	Sentence Similarity	NLI	220M	T5
ColBERTv1 [32]	Late Interaction	Retrieval	MSMarco	110M	BERT
ColBERTv2 [58]	Late Interaction	Retrieval	MSMarco	110M	BERT
MonoT5 Base [51]	Crossencoder	Ranking	MSMarco	223M	T5
MonoT5 Large [51]	Crossencoder	Ranking	MSMarco	737M	T5
MonoT5 3B [51]	Crossencoder	Ranking	MSMarco	2.85B	T5
stsb-roberta-large	Crossencoder	Sentence Similarity	STS-B	355M	RoBERTa
qnli-electra-base	Crossencoder	NLI	QNLI	110M	ELECTRA
nli-deberta-v3-base	Crossencoder	NLI	MultiNLI, SNLI	184M	DeBERTa
Qwen2-1.5B-Instruct [73]	Transformer	NTP	Crawled	1.5B	Qwen2Tokenizer
Qwen2-7B-Instruct [73]	Transformer	NTP	Crawled	7B	Qwen2Tokenizer
Mistral-7B-Instruct [26]	Transformer	NTP	Crawled	7B	BPE
Llama-3.1-3B-Instruct [16]	Transformer	NTP	Crawled	7B	Llama
Llama-3.2-8B-Instruct [16]	Transformer	NTP	Crawled	7B	Llama

Table 6: Model comparison for our experiments. NLI refers to natural language inference, and NTP refers to next token prediction. byte pair encoding with fallback. The crawled datasets represent undefined large training sets.

A.8 Results

In Figures 10, 11 and 12 we illustrate a close-up of the free generation synthetic experiments, the controlled generation experiments, and evaluation on ExcluIR as a result of our categorization mechanism.

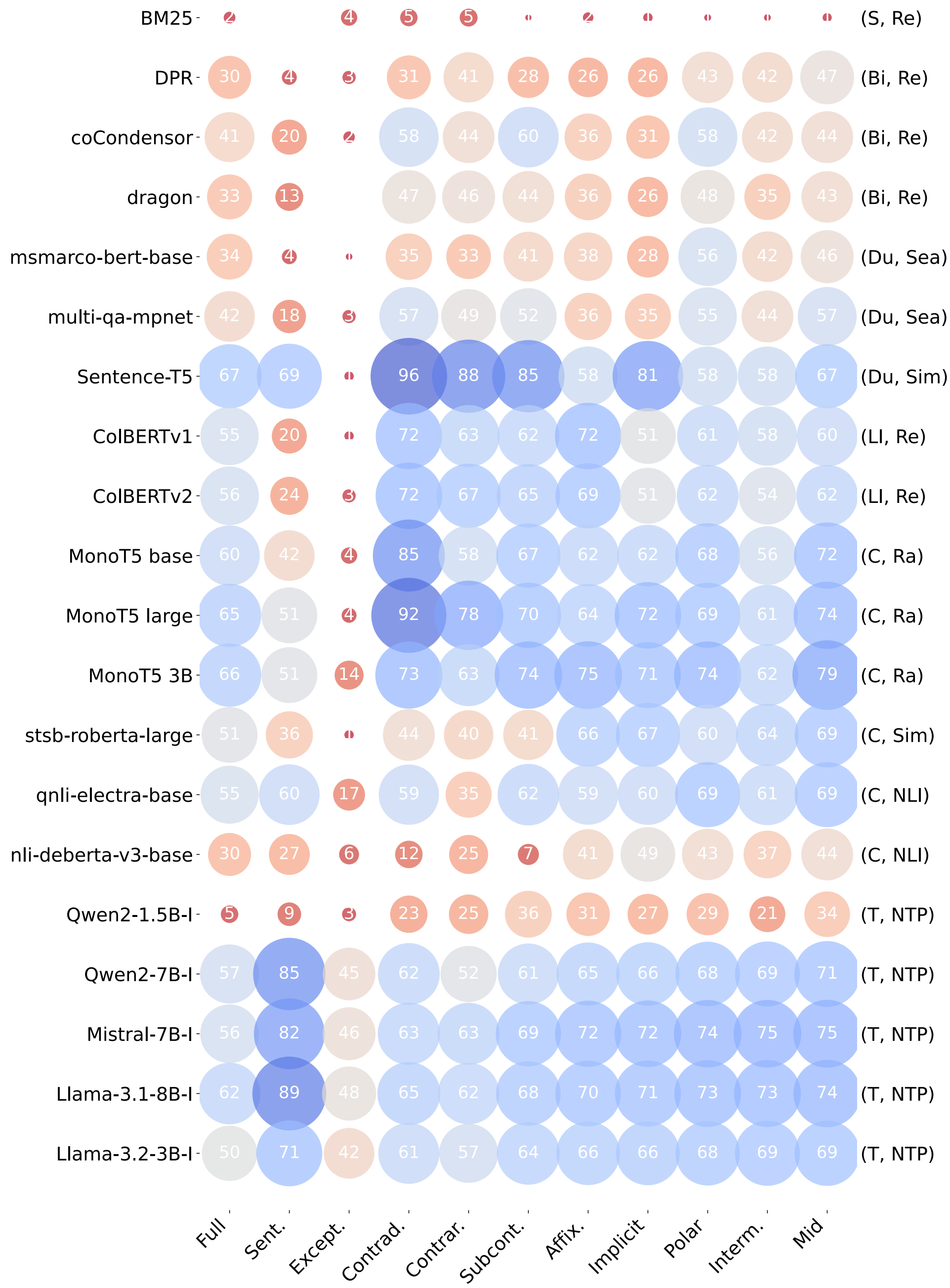

Figure 10: Close-up of results on the Free Generation.

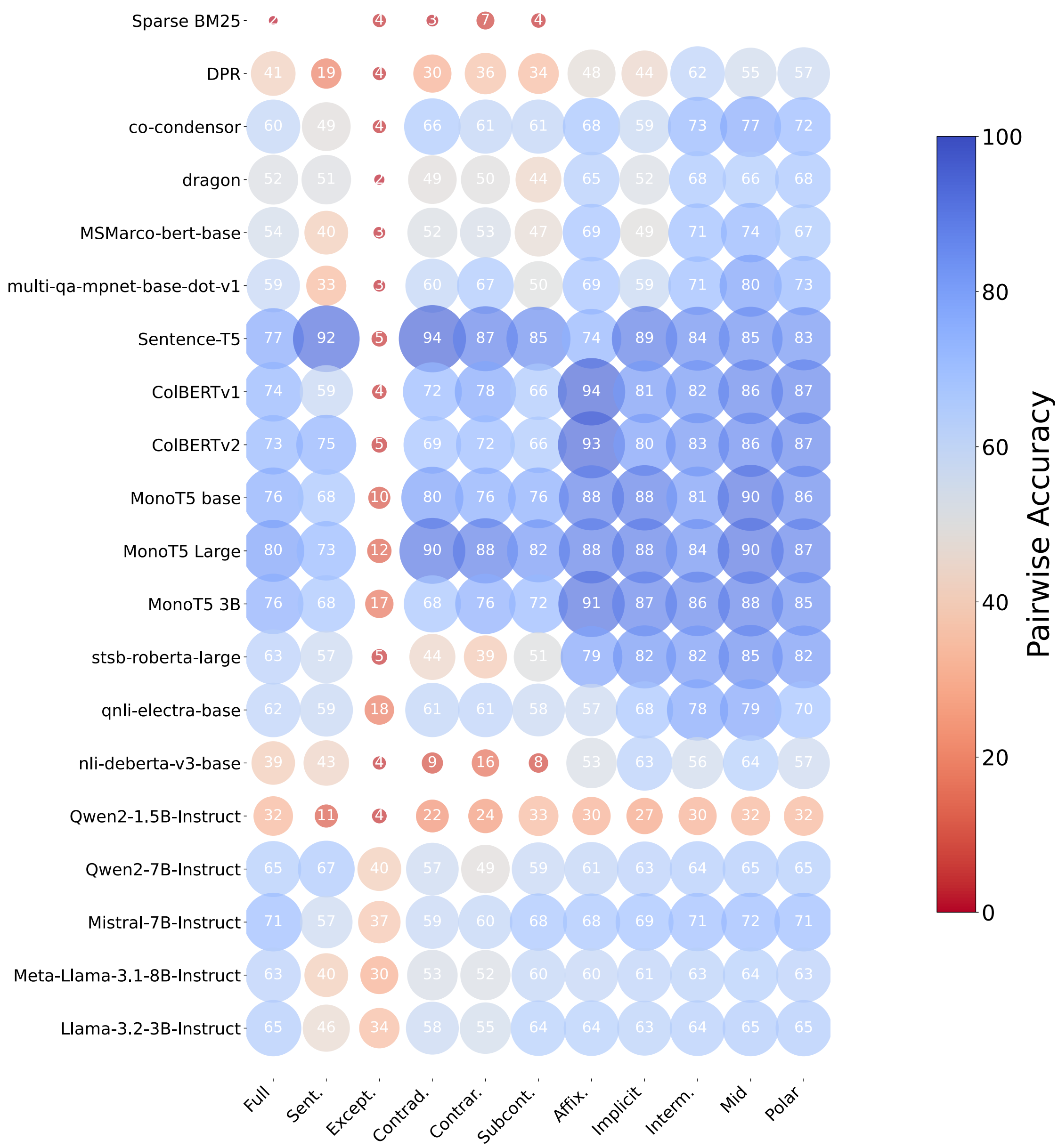

Figure 11: Pairwise Accuracy on the Controlled Generations dataset. Each column represents a negation type following our taxonomy, including the Full dataset in the first column. Each model is represented by one row.

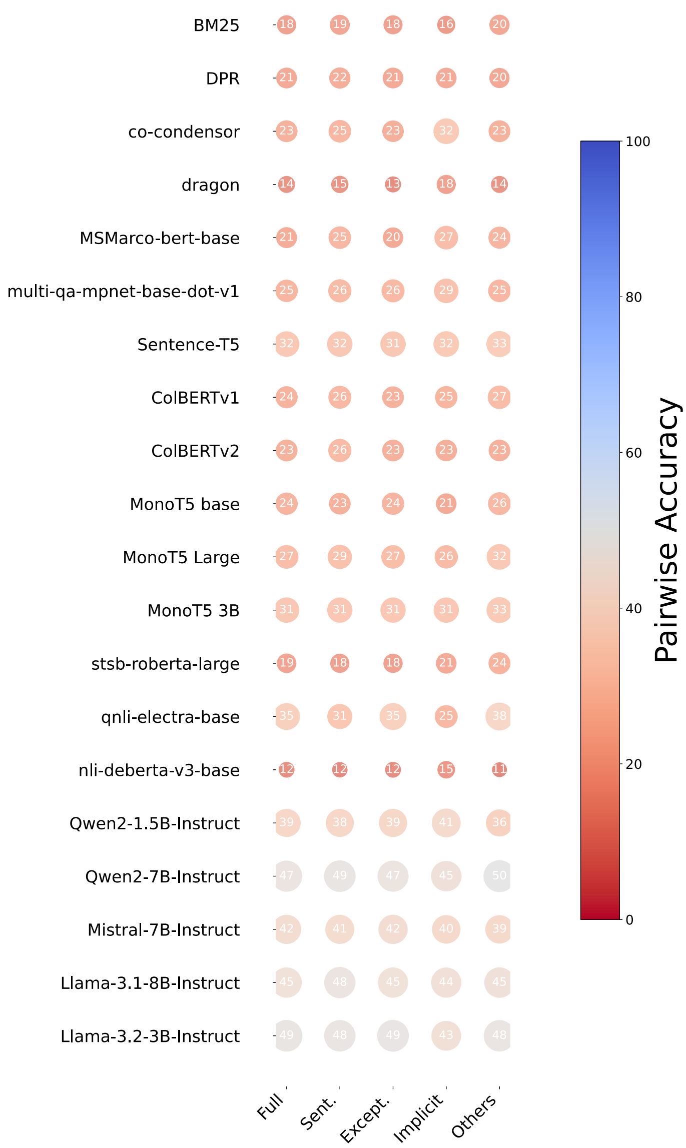

Figure 12: Pairwise Accuracy on ExcluIR. The dataset is split with out classification Mechanism.

A.8.1 Finetuning curves

Figures 13 and 14 illustrate the fine-tuning curves for ColBERT, MultiQA and Mistral when fine-tuned on synthetic, NevIR, and a mix of the two datasets. The evaluation is done on NevIR with pairwise accuracy, and on MSMarco with MRR@10.

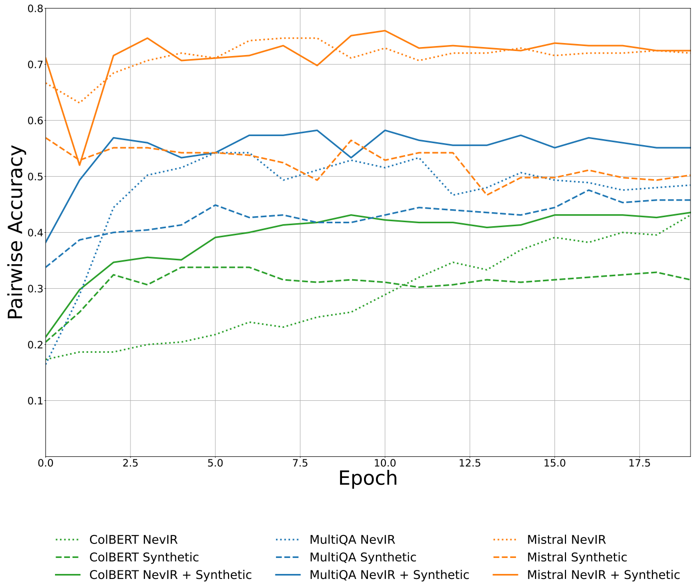

Figure 13: Fine-tuning results for ColBERT and MultiQA on 3 datasets: NevIR train, free generation train, and Mixed. Evaluated against NevIR dev.

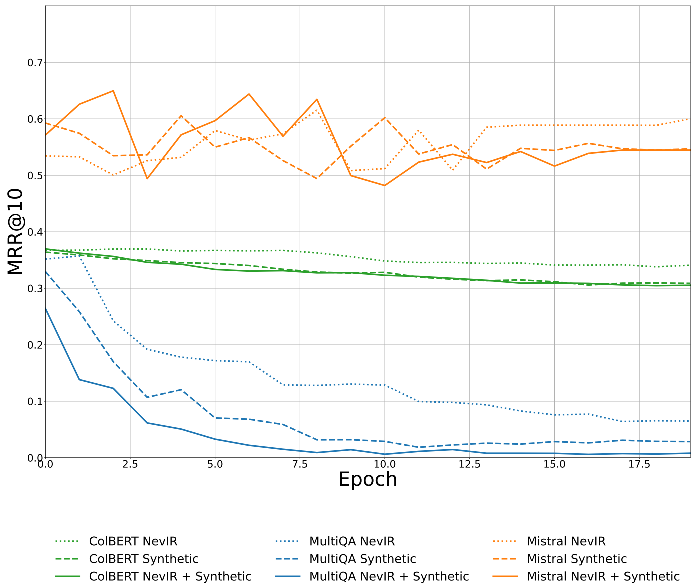

Figure 14: Fine-tuning results for ColBERT and MultiQA on 3 datasets: NevIR train, free generation train, and Mixed. Evaluated against MSMarco dev.